Ümit Demirbaga
Gagangeet Singh Aujla
Anish Jindal
Oğuzhan Kalyon

Big Data Analytics

Theory, Techniques, Platforms, and Applications

Springer

Ümit Demirbaga · Gagangeet Singh Aujla ·
Anish Jindal · Oğuzhan Kalyon

Big Data Analytics

Theory, Techniques, Platforms, and Applications

Springer

Ümit Demirbaga
Department of Medicine
University of Cambridge
Cambridge, UK

Department of Computer Engineering
Faculty of Engineering, Architecture, and Design
Bartin University
Bartin, Türkiye

Anish Jindal
Department of Computer Science
Durham University
Durham, UK

Gagangeet Singh Aujla
Department of Computer Science
Durham University
Durham, UK

Oğuzhan Kalyon
Faculty of Medical Sciences
Newcastle University
Newcastle Upon Tyne, UK

ISBN 978-3-031-55638-8 ISBN 978-3-031-55639-5 (eBook)
https://doi.org/10.1007/978-3-031-55639-5

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

Paper in this product is recyclable.

To my beloved wife, Dr. Kübra Kırca Demirbaga, and my cherished son, Asaf Aziz Demirbaga. They are not just coauthors of this book, but the coauthors of my life.

Dr. Ümit Demirbaga

To my daughter, Imanat Kaur Aujla, my son, Avitaj Singh Aujla, my love, Navneet Mann Aujla, my parents (Surjit Kaur and Prof. Kulwant Singh Aujla), and my in-laws (Kuljit Kaur and Datar Singh Mann).

Dr. Gagangeet Singh Aujla

To my late grandparents, lovely parents (Dr. Ashok Jindal and Dr. Anita Jindal) and beloved wife, Ankita.

Dr. Anish Jindal

To my mom, Zeynep, my dad, Mehmet, my sister, Arzu, and my love, Feyza Gül.

Oğuzhan Kalyon

Foreword

The world has changed in the information space over the last two decades due to three main factors: firstly there has been the routine deployment of high content measurement devices, from personal photos to satellite images to DNA sequencing to social media feeds. Secondly we have had the network and disk to store information at scale, often storing information that we don't know the value of. Thirdly increasingly sophisticated computational techniques, given labels such as "data science," "machine learning," and "AI," have been developed. All these phenomena can be collected under the heading of "Big Data."

This book provides an overview of these trends and the practical ways to handle this. Much of the complexity of dealing with data at this scale is about engineering—the practicalities about whether one can manage data flows robustly and cheaply—as well as the more statistically and algorithmically sophisticated analysis schemes. Here the reader can learn about both, and see this from a generic perspective of how to transmit, store, and organise data through to more subject-specific topics such as an introduction to Big Data approaches in bioinformatics. The book is designed for a broad audience, applicable to seasoned computational and data scientists as well as people at the start of their careers. The authors have provided both overviews and practical examples.

The world has already been changed by the advent of big data, and there is no doubt this will be part of this century. I recommend this book to everyone who wants to be part of this future.

December 2023

Prof. Ewan Birney, CBE, FRS, FMedSci
Deputy Director General
of the European Molecular Biology Laboratory (EMBL)
Director of European Bioinformatics Institute (EMBL-EBI)
Nonexecutive Director of Genomics England
Chair of the Global Alliance for Genomics and Health
Honorary Professor of Bioinformatics
University of Cambridge
Cambridge, UK

Preface

The deep significance of big data analytics is a beacon that helps enterprises navigate the challenges of making data-driven decisions in the ever-changing and quickly evolving field of information technology. Businesses now have a strategic imperative: to harness, evaluate, and draw useful insights from an unprecedented influx of data from varied sources. This book opens up as a thorough manual by exploring the fundamental ideas and methods that make up the formidable core of big data analytics and shedding light on its transformative potential. The importance of big data analytics stems from its ability to handle enormous amounts of data and its potential to reveal hidden connections, patterns, and trends that are missed by more conventional analytical techniques. Organisations that fully utilise big data have a competitive advantage in a world where data is being generated at a rate never seen before. Effective big data analytics has far-reaching consequences for various industries, from improving operational efficiency and resource allocation to facilitating data-driven innovation.

The revolutionary potential of big data analytics has become a key component of strategic efforts for organisations globally in the ever-expanding digital landscape. Opportunities and difficulties arise from the sheer amount and diversity of data collected as the globe grows more linked. Big data analytics emerges as the compass that leads decision-makers through the complexities of this data-rich terrain. Big data analytics stimulates innovation and advances artificial intelligence, machine learning, and predictive modelling in addition to its function in revealing insights. This book acts as a bridge to this ever-changing world, guiding readers through the fundamental ideas and innovative uses that shape the field of big data analytics.

This collaborative endeavour, authored by experts in the field, serves as your comprehensive guide by offering a multifaceted exploration of big data analytics. Tailored to the reader's unique role, expertise, and aspirations in the dynamic landscape of information technology, each chapter is crafted by a specialised contributor, which provides in-depth insights and expertise on specific aspects of the subject matter. This book is divided into 12 chapters. Chapter 1 establishes the groundwork by elucidating the essential properties of big data analytics, which explores the diverse range of techniques and provides an overview of the subsequent chapters. Chapter 2 provides a comprehensive guide to understanding big

data, unravelling its definition, characteristics, the renowned 5 Vs, challenges, and future directions. Transitioning seamlessly, Chap. 3 introduces the realm of big data analytics that delves into the pivotal role big data analytics plays in risk management, cost reduction, data-driven decision-making, and product development. As the narrative unfolds, Chap. 4 extends the discussion to the intersection of big data and cloud computing by offering a historical backdrop and elucidating cloud computing units. Chapter 5 immerses the reader in the expansive landscape of big data analytics platforms. The chapter dissects the components of systems by delving into the main characteristics and desired properties. It also provides practical case studies through real-world applications. Navigating further, Chap. 6 addresses the critical aspect of big data storage solutions, which explores traditional systems, such as relational databases and data warehouses, and presents contemporary solutions and cloud storage. Chapter 7 focuses on the pivotal realm of big data monitoring. Understanding the nuances of proactive and reactive monitoring, readers explore the monitoring components. The chapter concludes by acquainting readers with a spectrum of monitoring tools. As the reader ventures into Chap. 8, the book unfolds the intricate world of debugging big data systems. Acknowledging real-world performance problems, the chapter outlines systematic debugging steps and addresses common issues by providing a comprehensive guide to root cause analysis. Closing the narrative, Chap. 9 explores the synergy between machine learning and big data analytics. Diving into supervised and unsupervised machine learning, readers gain insights into challenges, preprocessing techniques, and popular algorithms. The chapter paints a holistic picture of the role machine learning plays in extracting meaningful patterns and predictions from massive datasets. Exploring the diverse applications of big data analytics across various sectors unveils a tapestry of innovation and strategic advancements. Chapter 10 delves into real-world case studies and applications that underscore the transformative impact of big data analytics, where each case study unveils the intricate interplay between big data analytics and sector-specific challenges. The examination of big data analytics for smart grids is expanded in Chap. 11, which also explains the intricacies of smart grids and illustrates how big data analytics is essential to improving their functionality. Finally, Chap. 12 delves into bioinformatics, which explores the intersection of big data and genomics. From understanding the challenges posed by big data in bioinformatics to examining frameworks for handling big genomic data, this section provides a holistic view of big data analytics in bioinformatics and a detailed case study in genomic medicine.

Whether you are an undergraduate student embarking on your academic journey, a master's student exploring advanced concepts, or a Ph.D. candidate or postdoctoral researcher delving into the nuanced intersections of big data analytics, this book serves as your comprehensive guide that offers a multifaceted exploration of big data analytics that is tailored to your unique role, expertise, and aspirations in the dynamic landscape of information technology. As a lecturer or educator, you will find valuable resources to support your teaching endeavours by incorporating real-world case studies and practical insights into your curriculum.

Moreover, this book offers a practical handbook for software engineers navigating the evolving IT landscape, which provides in-depth insights into platforms, storage solutions, and monitoring tools. Whether steering established companies or launching startups, business professionals will discover strategic guidance in risk management, cost reduction, and product development. The diverse range of applications is also explored, spanning government, health care, entertainment, banking, retail, and energy, which ensures relevance across industries. No matter your role or expertise, this book equips you with the knowledge to harness the transformative power of big data analytics for innovation, efficiency, and strategic decision-making.

Cambridge, UK — Ümit Demirbaga
Durham, UK — Gagangeet Singh Aujla
Durham, UK — Anish Jindal
Newcastle Upon Tyne, UK — Oğuzhan Kalyon
December 2023

Acknowledgements Umit extends his heartfelt gratitude to the Republic of Turkey and the Turkish Ministry of National Education for their unwavering financial and emotional support, which was pivotal in facilitating the successful stages of his academic journey.

Contents

List of Figures

Introduction

1

The world is being overrun by an unprecedented amount of data in the twenty-first century. This data comes from various sources, ranging from the subtle clicks of a mouse to the complicated data streams obtained via satellite technologies. Big data analytics is a discipline positioned to unearth priceless insights, spur innovation, and revolutionise decision-making paradigms due to the exponential growth of data. This book thoroughly introduces the complex field of big data analytics.

Big data analytics is fundamentally distinguished by its innate ability to uncover hidden possibilities inside the enormous data reservoirs inherent to our digital era [1]. It goes beyond merely managing huge datasets; instead, it explores the worlds of data interpretation, pattern identification, and predictive analysis, all of which lead to the support of crucial judgements. Big data analytics has permeated numerous industries, from the healthcare sector's pursuit of better diagnostics to the finance sector's search for data-driven strategies, offering competitive advantages, operational efficiency, and concrete benefits to various stakeholders.

This journey embarks upon an exploration of the quintessential attributes that delineate big data analytics. It reveals the complex methods and necessary equipment data scientists employ to decipher complex information. The journey continues into machine learning, Artificial Intelligence (AI), and data mining, where models and algorithms are the keys to revealing significant discoveries. A detailed summary of this book's contents is essential to our journey since it aims to give the reader a thorough understanding of big data analytics so they may successfully navigate the complex terrain of this diverse field.

Thus, this odyssey traversing the annals of big data analytics extends an invitation to all, irrespective of their status as seasoned data professionals, driven by the pursuit of honing their expertise or enthusiastic novices, harbouring an intrinsic curiosity regarding the uncharted territories of data exploration. In unison, this sojourn pledges not only to unveil the enigmatic intricacies inherent in the realm of big data analytics but also to scrutinise its expansive potential meticulously. In doing so, it endeavours to endow its passengers with the comprehensive skill set and profound knowledge

Ü. Demirbaga et al., *Big Data Analytics*,
https://doi.org/10.1007/978-3-031-55639-5_1

for harnessing the boundless power latent within the ever-pervasive data domain in our contemporary, data-centric milieu.

1.1 Essential Big Data Analytics Properties

Big data analytics is a multidisciplinary field that uses large, complex datasets to promote innovation across industries, extract useful insights, and influence decision-making. Six key components support its efficacy: it allows for secure parameter modifications, streamlined data integration, sophisticated data exploration, scalable data analysis, strong identity management, and extensive reporting features [2]. These features collectively constitute the fundamental components of a data-driven world. Combining these traits gives individuals and businesses the means to succeed in the data-centric landscape of the twenty-first century.

- **Scalability**: A fundamental quality in big data analytics is scalability, which concerns how well an analytics model can handle enormous amounts of data while maintaining controllable prices for hardware and cloud services [3]. Data scientists often face the difficulty of scaling their models to handle much larger, more complex data because they usually start their analytical journey with smaller datasets. Scalability is a basic design feature of an ideal big data analytics platform, making shifting from small-scale data analysis to large-scale operations easier. In today's data-intensive world, scalability plays a critical role in facilitating the effective management and extraction of relevant insights from the large amounts of data collected.
- **Version Control**: In big data analytics, version control plays an important role in managing iterative modifications to analytics models. Version control enables safe and reversible software alterations by systematically monitoring and documenting different versions [4]. In the event of unanticipated problems or system failures, this feature allows data scientists to quickly return to an earlier iteration of the analytics model, minimising project delays and guaranteeing budgetary compliance. However, fine-tuning model parameters is common for data scientists and comes with inherent hazards. Modest changes have the potential to seriously upset the system and cause major delays and cost overruns in projects.
- **Simple Integration Process**: Big data analytics technologies usually collect information from various sources, including business systems, cloud apps, and data warehouses. Intricate modifications are frequently required for these integrations to guarantee flawless communication and data processing. Data scientists work much more efficiently when analytics tools provide an easy-to-use integration method [5]. These solutions take up valuable time by simplifying the complexities involved in data source integration, enabling data scientists to concentrate on other important activities, such as improving analytics models. Simplifying the integration process makes preparing data smoothly and effectively easier, enabling data scientists to extract valuable insights more quickly.

- **Better Data Exploration**: Another important step in big data analytics is data exploration, in which data scientists thoroughly examine the gathered information to find previously undiscovered relationships, comprehend the context of business problems better, and create relevant analytical questions [6]. Significantly, faster data exploration is achieved with analytics tools that support it. They make it easier to test hypotheses quickly, spot weak or ambiguous data points rapidly, and offer tools for data visualisation. These kinds of capabilities enable data scientists to find important insights faster, which enhances decision-making in general.
- **Identity Management**: Identity management is essential to the overall data protection and cybersecurity strategy, an extensive database containing data about certain computer systems, software, and hardware. Identity management, which carefully regulates access, is essential to data security [7]. Identity management systems play a major role in data security by controlling and limiting access to particular data for systems or individuals. Access must be restricted to authorised individuals or devices to protect sensitive information's confidentiality and integrity and improve an organisation's overall cybersecurity posture.
- **Reporting Features**: The picture of big data analytics is incomplete without reporting features, which include real-time reporting, dashboard management, and location-based insights. These attributes facilitate enterprises to uphold a watchful and knowledgeable posture concerning their data assets [8]. Businesses are notified when significant data linkages or practical insights are discovered. Thanks to this real-time access, organisational leaders can effectively respond to important events, enabling them to make prompt and informed decisions. Reporting features would allow firms to stay flexible, adaptable, and responsive to new possibilities and trends in a data-driven environment.

1.2 Big Data Analytics Techniques

Big data analytics techniques include a broad range of complex approaches and instruments carefully designed to negotiate the complex terrain of large and complex datasets. They act as the cornerstone for businesses and organisations that offer a crucial way to seize important insights and establish a competitive edge in the fast-paced world of modern data-driven ecosystems. These methodologies are invaluable to enterprises and institutions, allowing them to identify trends, deconstruct intricate systems, and extract significant understandings from vast data.

Some of the key big data analytics techniques include:

- **Data Mining**: One of the most important methods in big data analytics is data mining, which uses sophisticated algorithms to find hidden relationships and patterns in enormous datasets [9]. Its main job is to break down and analyse complicated datasets to extract priceless information and spot new trends. Using this approach, data must first be gathered and prepared. Then, specific algorithms must be applied

to search for latent links and recurring patterns within the data environment. Due to its adaptability, data mining is used in many disciplines, including scientific research, financial analysis, market research, and health care.

- **Machine Learning**: Machine learning is the cutting edge of advanced analytics. It is a set of methods that allow computers to learn independently and improve their performance via experience [10]. This capacity is extremely useful in many areas, such as recommendation systems, categorization problems, and predictive analytics. With machine learning algorithms, computers can now identify complex patterns, anticipate outcomes based on data, and adjust to changing conditions. Using historical data, models are trained in this autonomous learning process, improving their capacity for precise prediction or decision-making. Essentially, machine learning elevates the potential of data-driven decision-making and gives businesses the tools to improve consumer experiences, automate processes, and streamline workflows. It is revolutionising how businesses function and interact with data in the modern day, with applications ranging from health care and banking to e-commerce and autonomous driving.
- **Natural Language Processing (NLP)**: NLP is a vital component within big data analytics that analyses and interprets unstructured textual data by exploring the intricacies of human language [11]. By bridging the gap between computational analysis and human communication, NLP techniques allow useful insights to be extracted from large amounts of textual data. NLP is a broad field that includes tasks such as sentiment analysis, which measures emotional tone; language translation, which fills in linguistic gaps; and the interpretation of linguistic nuances, such as idioms, slang, or contextual signals [12]. Organisations can automate language-based processes, generate a deeper knowledge of the content in documents, social media, or customer interactions, analyse textual information, and identify trends in consumer feedback by implementing NLP approaches. The increasing presence of digital material has increased the importance of this sector, making it a vital resource for fields like content suggestion, customer service automation, and social media analytics. Ultimately, these fields will influence how companies use and interact with textual data.
- **Data Visualisation**: Data visualisation plays a crucial role in big data analytics by bridging complex datasets and human understanding. Using interactive dashboards, graphs, and charts, this art form goes beyond simple data display and turns raw data into forms that are easy to understand [13]. When data is unprocessed, it can be confusing and overwhelming. Data visualisation addresses this by offering an understandable and simple way to work with complicated datasets. Decision-making becomes faster and more precise when patterns, trends, and anomalies concealed in the data become instantly visible through visualisation. It democratises access to insights and allows analysts, data scientists, and decision-makers to successfully convey their results to a larger audience. This leads to a more educated, data-driven approach to addressing business challenges.
- **Predictive Analytics**: A key strategy in big data analytics is predictive analytics, which uses historical data to forecast future patterns and results. This method uses advanced predictive modelling and analysis to provide priceless insights that

facilitate proactive strategy development and well-informed decision-making [14]. Using past data patterns and correlations, predictive analytics gives businesses a powerful tool to foresee future events, spot possible hazards, and grab opportunities. Doing this greatly improves their ability to adjust and react strategically to a business environment that is changing quickly. Whether used in marketing, finance, health care, or other industries, predictive analytics is essential for streamlining processes, allocating resources, and controlling risk. All of these factors contribute to more effective and profitable company outcomes.

- **Statistical Analysis**: In big data analytics, statistical analysis is a rock-solid foundation of scientific rigour that provides a systematic framework for hypothesis testing, data inference, and probabilistic inference-based decision-making [15]. Using this technique, data scientists and analysts can make significant findings by revealing hidden patterns, correlations, and trends in large datasets by applying mathematical and statistical methodologies. Statistical analysis is a lighthouse of impartiality that helps practitioners make evidence-based judgements by carefully analysing data's inherent uncertainties and variabilities [16]. It supports the validity and trustworthiness of findings in fields including experimental research, risk assessment, and quality control.
- **Clustering and Segmentation**: Clustering and segmentation techniques are indispensable for organising and extracting insights from complex big datasets that enable grouping similar data points into clusters or segments, illuminating inherent patterns and structures within the data [17]. With the unsupervised learning approach, clustering identifies hidden relationships and categorises data points based on their similarities, shedding light on intricate connections that might otherwise remain concealed [18]. In market analysis and targeted marketing, segmentation techniques divide datasets into distinct groups, which allows businesses to tailor their strategies and offers to specific customer segments [19]. These techniques enhance decision-making and streamline marketing efforts, leading to more effective and personalised approaches.
- **Real-time Analytics**: Real-time analytics enable organisations to extract immediate value from large-scale complex streaming data [20]. Fast, data-driven decision-making is essential in today's fast-paced, constantly changing digital environment, and this approach is motivated by its necessity. Organisations can make quick decisions responding to events, trends, or new information by processing data in real time. This enables them to optimise short-term prospects, reduce possible hazards, and improve overall operational effectiveness. When it comes to financial trading, cybersecurity, and e-commerce, for example, real-time analytics is very helpful when responding quickly to threats [21].
- **Distributed Computing**: When processing large datasets presents a barrier, distributed computing is a powerful paradigm in big data analytics. This method uses distributed systems' capabilities, which are best demonstrated by Hadoop and Spark, to enable the smooth transfer of data and computing workloads over a network of linked nodes or clusters [22]. The main benefit is that it may divide large, complicated data analysis jobs into smaller, more manageable chunks that can be performed simultaneously on several nodes, thereby reducing the time

needed for analysis [23]. From batch processing to real-time data streaming, distributed computing provides the computational power and scalability to handle complex calculations. It speeds up data analysis and improves fault tolerance by ensuring that the workload may be moved to another node without causing a lot of interruption in the case of a hardware breakdown.

1.3 Overview of This Book

This section provides an insightful glimpse into the content and structure of the book, offering a comprehensive framework of the core themes and subjects to be elucidated in subsequent chapters.

Foundations of Big Data Analytics: A fundamental investigation of big data analytics is at the core of this work. Data now serves as the foundation for innovation and decision-making across industries in the twenty-first century. It is essential to comprehend where data comes from and what it means in the modern world. This section examines the historical development of data analytics, tracing its origins and displaying its profound social impact. Readers will understand deeply how we arrived at today's data-centric society by exploring the process from data gathering to analysis.

Core Concepts and Technologies: Understanding big data analytics's fundamental ideas and tools is necessary for navigating the field. This section provides a thorough overview of data analytics's essential components. It introduces readers to the basic elements, tools, and frameworks necessary for successfully handling and analysing huge datasets. The foundation of contemporary data-driven decision-making, scalable data architecture, and fundamental analytics methodologies are highlighted. Readers will thoroughly understand the technology underlying big data analytics by the end of this part.

Practical Applications: Big data analytics is a dynamic force actively transforming industries; it is not merely a theoretical endeavour. This section illustrates practical applications from various fields, putting theory into action. Readers will see how data analytics fosters creativity, process optimisation, and priceless insights through interesting case studies and examples from health care, finance, and marketing. These real-world examples highlight the transformative potential of big data analytics and help readers imagine how it will affect their particular fields of interest.

Fostering Expertise Development: The journey through this book offers readers a way to develop competence in big data analytics. This book meets a range of learning needs, whether the reader is a beginner in the field or an experienced practitioner looking for significant insights. It offers a disciplined process for obtaining the knowledge and skills necessary for successfully navigating our modern, data-driven landscape. With the help of foundational concepts, technological know-how, practical application, and interactive activities, readers will leave the book prepared to take advantage of data's pervasive effect in the twenty-first century.

References

1. G.S. Aujla, N. Kumar, A.Y. Zomaya, R. Ranjan, Optimal decision making for big data processing at edge-cloud environment: An sdn perspective. IEEE Trans. Ind. Inf. **14**(2), 778–789 (2018)
2. 6 features that make big data analytics vital for businesses. Selerity. [Online]. Available: https://seleritysas.com/2019/05/06/6-features-that-make-big-data-analytics-vital-for-businesses/
3. H. Hu, Y. Wen, T.-S. Chua, X. Li, Toward scalable systems for big data analytics: A technology tutorial. IEEE Access **2**, 652–687 (2014)
4. D.L. Atkins, T. Ball, T.L. Graves, A. Mockus, Using version control data to evaluate the impact of software tools: A case study of the version editor. IEEE Trans. Soft. Eng. **28**(7), 625–637 (2002)
5. X.L. Dong, D. Srivastava, Big data integration, in *IEEE 29th International Conference on Data Engineering (ICDE)*. (IEEE, 2013), pp. 1245–1248
6. A. Wasay, M. Athanassoulis, S. Idreos, Queriosity: Automated data exploration, in *IEEE International Congress on Big Data*. (IEEE, 2015), pp. 716–719
7. P. Jain, M. Gyanchandani, N. Khare, Big data privacy: a technological perspective and review. J. Big Data **3**, 1–25 (2016)
8. P. Mikalef, M. Boura, G. Lekakos, J. Krogstie, Big data analytics capabilities and innovation: the mediating role of dynamic capabilities and moderating effect of the environment. British J. Manage. **30**(2), 272–298 (2019)
9. X. Wu, X. Zhu, G.-Q. Wu, W. Ding, Data mining with big data. IEEE Trans. Knowl. Data Eng. **26**(1), 97–107 (2013)
10. Challenges and approaches, A. L'heureux, K. Grolinger, H.F. Elyamany, M.A. Capretz, Machine learning with big data. IEEE Access **5**, 7776–7797 (2017)
11. R. Sharma, P. Agarwal, A. Arya, Natural language processing and big data: a strapping combination, in *New Trends and Applications in Internet of Things (IoT) and Big Data Analytics*. (Springer, 2022), pp. 255–271
12. J.C. Eichstaedt, M.L. Kern, D.B. Yaden, H.A. Schwartz, S. Giorgi, G. Park, C.A. Hagan, V.A. Tobolsky, L.K. Smith, A. Buffone et al., Closed-and open-vocabulary approaches to text analysis: A review, quantitative comparison, and recommendations. Psychol. Methods **26**(4), 398 (2021)
13. D. Keim, H. Qu, K.-L. Ma, Big-data visualization. IEEE Comput. Graph. Appl. **33**(4), 20–21 (2013)
14. A. Shi-Nash, D.R. Hardoon, Data analytics and predictive analytics in the era of big data, in *Internet of Things and Data Analytics Handbook* (2017), pp. 329–345
15. C. Wang, M.-H. Chen, E. Schifano, J. Wu, J. Yan, Statistical methods and computing for big data. Stat. Interface **9**(4), 399 (2016)
16. A. Der Kiureghian, Analysis of structural reliability under parameter uncertainties. Probab. Eng. Mech. **23**(4), 351–358 (2008)
17. F. Yoseph, N.H. Ahamed Hassain Malim, M. Heikkilä, A. Brezulianu, O. Geman, N.A. Paskhal Rostam, The impact of big data market segmentation using data mining and clustering techniques. J. Intell. Fuzzy Syst. **38**(5), 6159–6173 (2020)
18. O. Nasraoui, C.-E.B. N'Cir, Clustering methods for big data analytics. Tech. Toolbox. Appl. **1**, 91–113 (2019)
19. P. Monil, P. Darshan, R. Jecky, C. Vimarsh, B. Bhatt, Customer segmentation using machine learning. Int. J. Res. Appl. Sci. Eng. Technol. (IJRASET) **8**(6), 2104–2108 (2020)
20. R. Ranjan, Streaming big data processing in datacenter clouds. IEEE Cloud Comput. **1**(01), 78–83 (2014)
21. M. Cao, R. Chychyla, T. Stewart, Big data analytics in financial statement audits. Account. Horizons **29**(2), 423–429 (2015)

22. K. Kambatla, G. Kollias, V. Kumar, A. Grama, Trends in big data analytics. J. Parall. Distrib. Comput. **74**(7), 2561–2573 (2014)
23. U. Demirbaga, G.S. Aujla, Federated-ann based critical path analysis and health recommendations for mapreduce workflows in consumer electronics applications. IEEE Trans. Consum. Electron. (2023)

Big Data

2

Machine learning will automate jobs that most people thought could only be done by people.

—Dave Waters

This chapter introduces the fundamental concepts of big data, offering a comprehensive understanding of its definition, key characteristics, and the widely recognised 5 Vs. The multifaceted challenges associated with realising the enormous potential of big data are explored, encompassing issues related to data collection, storage, privacy, security, and the complexities of deriving value from this extensive resource. In addition, avenues for harnessing the power of big data are investigated, including applying advanced analytics and machine learning, utilising data visualisation techniques, and implementing communication strategies. Lastly, a glimpse into the future of big data is provided, shedding light on emerging trends and directions that will shape its ongoing evolution and influence across various domains.

2.1 Definition of Big Data

Big data is a term that emerged in astronomy and genetics but has now been used on the Internet and has become a part of our everyday lives without our awareness or actively contributing to it. Much data has been stored, processed, and managed due to the computer's effectiveness in every part of our lives. The growing Internet use by businesses, corporations, and individuals has resulted in the circulation, processing, and dissemination of these data in electronic media [1]. The data we have described comprises information entered and stored as a condition of service, as well as a large

Ü. Demirbaga et al., *Big Data Analytics*,
https://doi.org/10.1007/978-3-031-55639-5_2

amount of information that appears superfluous and worthless and it's overgrowing. According to the 2019 numbers, the globe produces about 2.5 quintillion bytes of data daily, and the total data size is expected to exceed 45 times the present time in 10 years [2]. As a result, we have a garbage heap of unstructured data. It wasn't long before it was realised that this occurrence, which was called "information dump," was initially a big treasure that couldn't be utilised due to its lack of structural integrity. This dump, which contains data such as web server logs, social media sharing and publishes, blogs, microblogs, and Internet statistics, effectively holds a potential advantage. These data, if read correctly using the relevant analysis tools, should have been able to assist in making crucial decisions, minimising risks, and signing new discoveries and discoveries. Big data analytics is becoming increasingly significant in several essential industries, such as government, health care, entertainment, weather patterns, cyber-physical systems, finance, IoT technologies, and natural disaster management, for these critical reasons [3].

In the past decade, the amount of data being created is truly mind-boggling. The amount of data produced in the last 2 years corresponds to 90 percent of all data worldwide. Since the beginning of the COVID-19 pandemic, we have become increasingly dependent on the Internet. As a result, broadband data usage has increased by 47%, and there are now more gigabit and terabyte subscribers, leading to skyrocketing growth in data generation. The total data in the world in 2021 was 74 zettabytes, and the total data size is expected to exceed 45 times the current time in 10 years. With this huge data, most of which is unstructured data, big data analytics has become an even more important and indispensable segment of the digital age.

Figure 2.1 shows the global volume of data/information created, recorded, duplicated, and consumed from 2010 to 2025 [4].

This phenomenal increase in data has had a significant impact on enterprises. Traditional databases, such as relational databases, have been pushed to their limits. The strain of "Big Data" has caused many of these systems to fail. Traditional systems, as

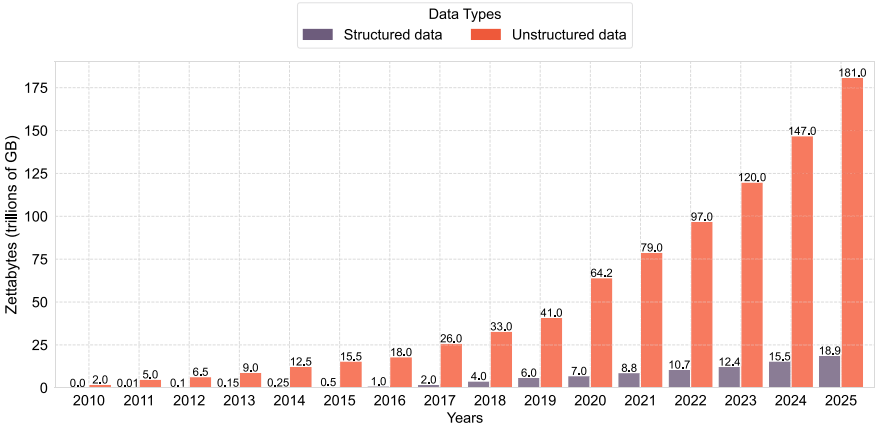


Fig. 2.1 Data storage growth in enterprises worldwide

well as the data management strategies that go with them, haven't been able to keep up with big data, and as a result, they have completely failed to tackle the challenges of big data.

2.2 Characteristics of Big Data

The enormous scope and complexity of the data collected across numerous businesses and areas define big data's characteristics [5]. This data influx is distinguished by its volume because it is amassing at an unprecedented rate and frequently reaches terabytes, petabytes, or even exabytes in size [6]. Furthermore, big data has high velocity, which means that it is produced and processed in real time or very near real time, enabling prompt analysis and decision-making. Big data includes structured, semi-structured, and unstructured data from social media, sensors, logs, and multimedia material. Because there are so many different forms and structures, traditional data management strategies frequently fall short in integrating, storing, and analysing the data that results from this diversity [7].

Big data also has variability and veracity, where "variability" alludes to the dynamic aspect of the data, whose volume, velocity, and variety can quickly vary over time. This dynamic nature necessitates flexible and scalable processing frameworks to manage the changing data landscape properly. Big data veracity also refers to the data's dependability, accuracy, and credibility. Ensuring the integrity and authenticity of the data is essential for insightful analysis and decision-making as data sources expand and become more varied. Data cleaning, preprocessing, and validation procedures address the veracity dilemma, enabling organisations to gain insightful knowledge from the huge sea of big data. The next section details the characteristics of big data called 5 Vs.

2.3 The 5 Vs of Big Data

The defining qualities of big data analytics are encapsulated in the "5 Vs" of big data. These factors act as cornerstones for comprehending and evaluating massive amounts of data. They include the vast volume of data, its potential value for businesses and organisations, the wide variety of data types and sources, the rapid rate at which data is produced and processed, and the requirement for guaranteeing data accuracy and reliability. These five characteristics serve as a framework for understanding the different opportunities and difficulties posed by big data and direct the creation of tools and strategies for efficiently managing and gleaning insights from this vast data. Figure 2.2 depicts the big data generation and its 5 Vs in detail.

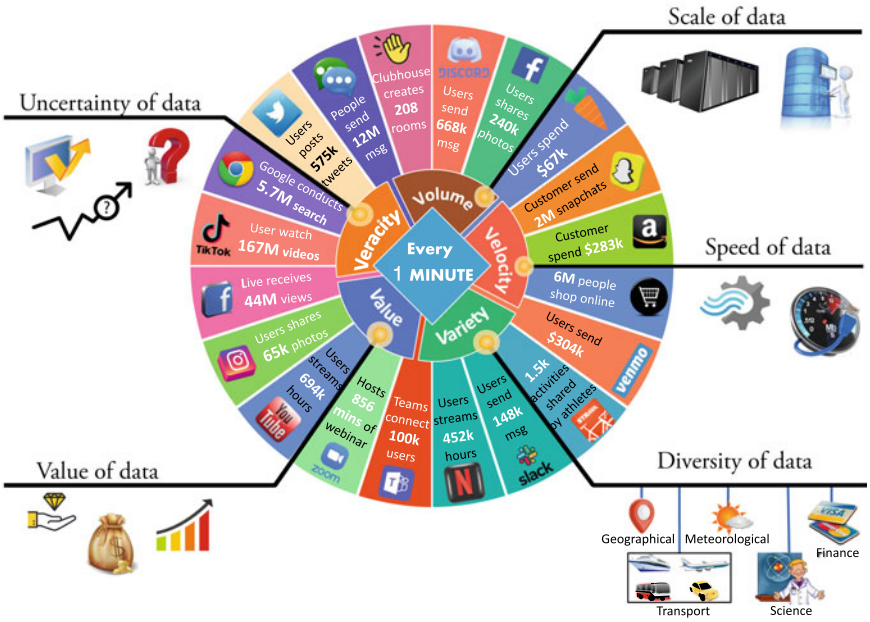


Fig. 2.2 A comprehensive look at big data [8]

Fig. 2.3 Internet users and penetration worldwide [9]

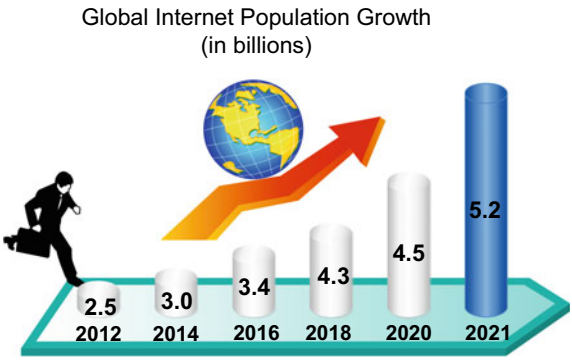


Figure 2.3 illustrates the dynamic and consistent growth of the global Internet population from 2012 to 2021. The chart depicts a steady increase in Internet users worldwide, from 2.5 billion in 2012 to 5.2 billion in 2021. This significant expansion highlights the widespread adoption of digital connectivity across the globe during this period. The upward trajectory indicates the increasing accessibility and relevance of the Internet, which reflects its integral role in shaping our interconnected world over the past decade.

2.3.1 Volume

Big data volume refers to the sheer amount of data produced, gathered, and stored across numerous sources and platforms [10]. It represents the vast amount of information businesses can use for analysis and decision-making, containing terabytes, petabytes, or even exabytes of data, which is astounding. This wealth of data storage, processing, and analysis presents formidable obstacles. Traditional data management systems cannot handle such enormous numbers, necessitating the development of cutting-edge methods and tools to efficiently store, access, and extract knowledge from these vast datasets.

There are several causes behind the data volume's exponential rise. First, there has been a massive increase in data generation due to the broad adoption of digital technology and the proliferation of Internet-connected gadgets. Data is continually being created unprecedentedly, from social media interactions and online purchases to sensor readings and machine-generated records. Additionally, organisations are becoming increasingly aware of the importance of gathering and keeping data for later analysis, which has led to the development of enormous data banks.

Scalable and distributed storage solutions are required to handle the volume of big data. The requirement for scalable, affordable solutions has given rise to technologies like distributed file systems and cloud storage platforms. By distributing data among several nodes or servers, these systems make it possible to store and retrieve large datasets, enabling parallel processing and quick data access. Utilising big data effectively brings both opportunities and challenges. On the one hand, the volume may be too great for conventional data processing and analysis techniques, resulting in problems like data latency and information overload. Conversely, much data offers businesses many insights and chances for data-driven decision-making. Organisations can find important patterns, trends, and correlations hidden within the enormous volumes of data by using advanced analytics techniques, such as machine learning and data mining [11].

2.3.2 Value

The value of big data lies in its immense potential to deliver meaningful insights, drive informed decision-making, and unlock transformative opportunities for organisations [12]. Big data consists of vast and diversified informational collections that provide insightful knowledge about various business operations, customer behaviour, market trends, and more if properly analysed and assessed. Big data is significant for the insights it can offer, which go beyond just its bulk.

Organisations can obtain a competitive edge in their respective sectors by utilising the potential of big data, discovering unachievable latent patterns, correlations, and trends with sophisticated data analysis tools like advanced analytics, machine learning, and predictive modelling [13]. These insights can result in increased operational effectiveness, better decision-making procedures, focussed marketing initiatives, individualised consumer experiences, and the creation of novel goods and services.

Additionally, big data's usefulness resides in its capacity to support evidence-based tactics and ease decision-making based on that data. Organisations may stop relying on gut instinct or anecdotal evidence and base their decisions on real facts and insights by utilising extensive and current information. A culture of accountability and openness is fostered inside organisations thanks to this change towards data-driven decision-making, which also helps to reduce the risks brought on by uncertainty.

Big data also has the potential to open up new revenue sources and business models. Businesses can monetize their data assets by providing data-driven goods and services, developing data markets, or using data partnerships and collaborations. Big data has value beyond internal usage since it allows organisations to collaborate and generate insights for businesses, academics, and society by sharing anonymised and aggregated data with outside parties.

2.3.3 Variety

The wide diversity of data types, formats, and sources in the rich informational environment available for analysis and decision-making is called the variety of big data [14]. Big data includes a variety of data types, including structured, unstructured, and semi-structured data, in contrast to traditional data sources, which often only contain structured data. Big data's variability presents considerable hurdles regarding data integration, storage, processing, and analysis.

Structured data, frequently derived from relational databases, is the conventional and well-defined data structure that may be arranged into rows, columns, and tables. Contrarily, unstructured data is generally created in text documents, emails, social media postings, photos, videos, and audio files and has no established structure. There is semi-structured data between these two types of data, which has some organisational structure but does not follow a preset format. The wide range of data sources from which big data is derived further increases its variability. These sources may include social networking websites, electronic commerce sites, mobile phones, sensors, log files, etc. Each data source offers its special qualities, offering chances to learn insightful things but also causing problems with data governance, data integration, and quality [15].

Big data management demands specialised tools, technologies, and processes to handle the diversity of the data. Data from many sources and formats are transformed and standardised into a single structure appropriate for analysis using data preparation techniques. Unstructured data is mined for useful information using advanced analytics techniques like sentiment analysis, natural language processing, and picture identification [16]. Additionally, varied data kinds are successfully managed and stored using technologies like data lakes and NoSQL databases.

Organisations have both possibilities and challenges due to the diversity of big data. On the one hand, including unstructured and semi-structured data offers priceless insights and context-rich information that may result in more thorough analysis and better judgements. The diversity of data, however, poses difficulties for data

integration, data quality control, and guaranteeing consistency among various data sources [17].

2.3.4 Velocity

Big data is created, gathered, and processed quickly—in real time or very close to it—and this is referred to as its velocity. Big data is a constant flow of information that calls for quick collection, analysis, and reaction, unlike traditional data sources, which sometimes need batch processing and irregular updates. In terms of data input, storage, processing, and decision-making, the velocity dimension of big data presents difficulties and possibilities for organisations. Data is produced at an unprecedented rate due to the spread of digital technology and systems' interconnection. Online transactions, social media interactions, sensor readings, and machine-generated logs influence the continual flow of data that organisations must manage [18]. Effective techniques for capturing and storing data as it is created are required to prevent information loss and enable real-time analysis due to the velocity of big data.

Real-time or almost real-time processing is required to use the velocity of big data properly. Organisations may acquire timely insights, spot patterns, and act fast on new trends or anomalies by analysing data as it is created [19]. This velocity-driven analysis enables organisations to make quick decisions, reduce risks, and adjust to quickly shifting market situations. Organisations use tools and methods that enable real-time data processing to handle the velocity of big data. The continuous ingestion and processing of data streams are made possible by stream processing frameworks like Apache Kafka[1] or Apache Flink,[2] enabling fast analysis and insights. Near-instantaneous replies to data queries are made possible through distributed computing systems and in-memory databases, which speed up data processing [20].

Organisations have possibilities and difficulties due to the pace of big data. On the one hand, the rapid rate of data production necessitates the development of scalable and effective infrastructure to manage the flood of data. The timeliness and relevancy of insights drawn from the data might be affected by delays caused by data intake or processing bottlenecks. However, the velocity of big data makes it possible for businesses to make data-driven decisions in dynamic contexts that are changing quickly.

2.3.5 Veracity

Big data veracity is the quality, accuracy, and reliability of the data being gathered, handled, and examined [21]. Big data is frequently characterised by its heterogeneous and diversified nature, integrating information from many sources, formats,

[1] https://kafka.apache.org/.
[2] https://flink.apache.org/.

and degrees of reliability. Big data's veracity dimension presents substantial issues for organisations since they must ensure the data's accuracy, validity, and integrity to provide actionable insights. Data accuracy, completeness, consistency, and timeliness are just a few facets of data veracity. Inaccurate or lacking data might result in incorrect inferences and poor decision-making. Effective data integration and analysis can be hampered by data inconsistency, which results from differences in data semantics, formats, or standards. Timeliness is particularly important since stale or old data may not accurately reflect the situation of the world right now and can provide false conclusions.

Organisations need strong data governance procedures and quality assurance methods to overcome the problems with big data's veracity. To do this, frameworks for data quality must be established, metrics must be defined, and processes must be put in place for validating and cleaning up data. Data lineage monitoring allows organisations to track the origin and transformation of data, improving data integrity, while data profiling techniques may be used to spot abnormalities and discrepancies [22]. Furthermore, the trustworthiness and reliability of data sources impact the authenticity of big data. To guarantee the dependability of the data they gather, organisations must evaluate the integrity and reputation of data suppliers. Supplying transparent and dependable data sources, collaborative data-sharing programmes, data provenance, and data certification systems can improve data veracity.

It is imperative to address the validity of big data for organisations to feel confident in the conclusions drawn from data analysis. Organisations can take appropriate actions based on trustworthy information by guaranteeing data correctness, completeness, consistency, and timeliness. A culture of data-driven decision-making must also be established inside organisations, where data veracity is valued and data quality is regularly checked and enhanced.

2.4 Challenges in Big Data

The emergence of big data has presented some challenges for businesses across various industries. The sheer amount, diversity, pace, and authenticity of the data present obstacles, while the opportunity to gain insightful knowledge and influence decision-making current opportunities. The main barriers and possibilities related to big data are covered in this section.

Data management is one of big data's most significant concerns. Effective data storage, processing, and retrieval procedures are required due to the volume of data. Organisations must employ cutting-edge technologies like distributed computing and cloud-based storage solutions since traditional data management methods frequently find handling such vast amounts of data difficult. The diversity of data sources and formats makes data integration and quality control processes more difficult. Organisations must invest in strong data governance structures and cleansing procedures to guarantee consistency and correctness [23]. Another problem with big data analysis is that advanced analytical methods and algorithms are needed to extract relevant

insights from massive volumes of data [24]. To provide practical wisdom, data scientists and analysts must be proficient in statistical modelling, machine learning, and data visualisation. Additionally, organisations struggle with the velocity of data as they aim for real-time or almost real-time analysis to make quick judgements [25]. To overcome these difficulties, scalable and effective processing architectures are essential. Examples include parallel computing and stream processing.

Big data also raises serious issues with data security and privacy. Organisations face the problem of protecting data against unauthorised access, breaches, and abuse due to the gathered volume of sensitive and personal information [26]. Maintaining trust and guaranteeing ethical data practises depends on following data protection laws, establishing strong security protocols, and implementing privacy-preserving measures.

Despite these difficulties, big data also offers a wealth of potential for businesses. Organisations can now acquire unheard-of insights into consumer behaviour, market trends, and operational efficiency thanks to the abundance of data accessible. Organisations may personalise client experiences, streamline operating procedures, and find new sources of income with improved analytics skills. Big data analytics can also help with evidence-based decision-making, resulting in better results and competitive advantages [27]. Additionally, big data also creates opportunities for research and innovation. Industries can use data-driven initiatives to develop new goods and services, improve supply chain management, and promote digital transformation. Big data analytics may speed up discoveries, enable personalised therapy, and promote evidence-based treatments in healthcare and scientific research domains. Furthermore, big data presents chances for social and economic influence through influencing public policy, enhancing public services, and solving societal issues.

2.4.1 Data Collection and Storage Challenges

The process of data collecting and storage is essential for organisations to successfully manage and utilise massive amounts of different and heterogeneous data. Big data refers to vast datasets from various sources, including social media platforms, Internet of Things (IoT) gadgets, sensors, and transactional systems [28]. This data quantity poses opportunities and difficulties regarding gathering, organisation, and storage.

The systematic gathering of raw data from many sources is required for big data collection. Capturing and extracting data in different formats necessitates the deployment of data-collecting technologies such as web crawlers, Application Programming Interface (APIs), and sensors [29]. These processes should be built to deal with the huge volume and high velocity of data generated in real time or very close to real time. Data validation and cleansing procedures must also be used in data-collecting operations to guarantee data quality, accuracy, and consistency.

Data needs to be stored in a system that can handle the size and complexity of big data after it has been collected. As they may have trouble managing the enormous number and variety of data, traditional relational databases frequently fall short. Due

to the need for scalable and fault-tolerant big data storage structures, organisations resort to distributed storage systems like Apache Hadoop and Apache HBase. These systems use NoSQL databases and distributed file systems like Hadoop Distributed File System (HDFS) [30] to store and manage data across several nodes or clusters.

Big data storage requires taking accessibility and retrieval into account as well. Fast and reliable data retrieval is necessary, especially when working with large-scale datasets; therefore, effective indexing and querying procedures are essential. Organisations can use strategies like data splitting, sharding, and indexing to enhance the performance of data storage and retrieval. Big data storage also has to consider data security and privacy considerations. Organisations must put in place strong security measures, including encryption, access control, and data anonymisation techniques, to protect data from breaches and unauthorised access due to the sensitivity of some data types, like personal information and private business data. Maintaining trust and ethical data practises requires strict adherence to data protection laws and privacy rules.

2.4.2 Data Quality and Integrity Challenges

Data quality and integrity are essential in big data because they guarantee the dependability, correctness, and consistency of the data being gathered, stored, and analysed. An in-depth discussion of data quality and integrity is provided in this part, along with an emphasis on its importance and main factors. Data fitness for use in a certain environment or application is called data quality. It includes several characteristics: precision, thoroughness, coherence, timeliness, and relevance. Ensuring data quality is essential since it directly influences how reliable conclusions and judgements drawn from the data will be. Throughout the data lifetime, organisations must use reliable procedures to evaluate and enhance the quality of the data.

Data integrity, on the other hand, means preserving data's continuity, correctness, and dependability over time and across various systems or components. It entails guarding against and spotting data tampering, unauthorised alterations, and corruption. Data integrity safeguards are crucial to maintaining the reliability and validity of data, particularly in important applications where data integrity breaches might have serious repercussions. Organisations use a range of tactics and procedures to ensure the integrity and quality of their data. Techniques for data validation are used to verify the precision and completeness of data at the moment of collection [31]. Correcting mistakes, eliminating duplication, and addressing inconsistencies can entail data profiling, cleansing, and transformation. By defining and enforcing data quality standards, data governance frameworks ensure adherence to data management policies and practices. Implementing data encryption is another component of data integrity measures that guards against unauthorised access to or manipulation of data while it is being sent or stored [32]. Data integrity is checked using hashing algorithms and digital signatures, which provide distinctive identifiers or signatures that may be used to spot changes or manipulation in the data.

Data quality and integrity are crucial in industries like health care, banking, and critical infrastructure, where accuracy and dependability are crucial. For example, HIPAA (Health Insurance Portability and Accountability Act) in the healthcare sector or PCI DSS (Payment Card Industry Data Security Standard) in the finance sector are just two examples of the strict requirements and standards organisations frequently follow to ensure data quality and integrity. Additionally, monitoring, evaluating, and improving data quality and integrity are continual procedures. To assess the success of data quality programmes and pinpoint areas for development, organisations use key performance indicators (KPIs) and metrics for data quality. Routine audits and data profiling efforts aid in identifying data abnormalities, inconsistencies, or departures from established quality standards.

2.4.3 Privacy and Security Concerns

Big data privacy issues stem from the dangers of unauthorised access, use, and exposure to personal information [33]. Data breaches, data mining, profiling, and re-identification methods can jeopardise individuals' privacy owing to the expansion of digital technologies and the interconnectedness of data ecosystems [34]. For the rights of people to be upheld and ethical data practices to follow, privacy protection is essential.

Data protection against unauthorised access, malicious attacks, and data breaches are security considerations with large amounts of data [35]. Big data platforms are appealing targets for hackers because they frequently store and handle enormous volumes of sensitive and valuable data. To minimise threats, including unauthorised data access, data theft, data manipulation, or interruption of data operations, efficient monitoring, access control, encryption, and authentication methods, which improve the security of big data systems, need to be implemented [36].

In addition to these general concerns, there are particular difficulties and issues regarding privacy and security with big data. One difficulty is the inherent complexity and variety of big data ecosystems, which include numerous stakeholders, data sources, and data flow. Comprehensive strategies, regulations, and technologies are required to ensure the privacy and security of these interconnected networks. Additionally, it is not easy to protect privacy while maintaining the usefulness and value of the data for analytical purposes when they have been anonymised and de-identified. It is essential to balance data anonymisation and usability to safeguard privacy without sacrificing insights drawn from the data.

Legal and regulatory frameworks address big data privacy and security issues. Data protection legislation, such as the General Data Protection Regulation (GDPR),[3] should be adopted by organisations to ensure the lawful and transparent processing of personal data [37,38]. Compliance with these standards is essential for preserving confidence in big data platforms and safeguarding people's privacy rights [39]. Fur-

[3] https://gdpr-info.eu/.

thermore, ethical issues related to the security and privacy of big data are becoming more prominent. The use of ethical data practices by organisations, such as informed permission, data reduction, purpose limitations, and openness, is becoming increasingly required. Big data methods incorporating ethical standards help guarantee that privacy and security problems are handled responsibly and accountable.

2.4.4 Issues with Extracting Value from Big Data

Data analysis is essential for obtaining large amounts of data. Companies use different tools and procedures to identify patterns, trends, and correlations within enormous amounts of data. While diagnostic analytics aids in identifying the causes of certain results, descriptive analytics offers a summary of the data. While prescriptive analytics provides suggestions and insights for decision-making, predictive analytics uses statistical modelling and machine learning algorithms to forecast future occurrences.

Organisations use sophisticated data-mining techniques to derive values from massive data. These methods include exploratory data analysis, grouping, classification, regression, and association rule mining [40]. They assist in identifying anomalies, hidden patterns, customer segments, and predictive models that may be used to inform strategic choices, streamline processes, and boost corporate performance. Furthermore, extracting value from large datasets relies heavily on applying machine learning techniques and AI. Massive volumes of data can be automatically analysed using machine learning algorithms, revealing important insights and patterns that cannot be seen using conventional analytical techniques. Unstructured data sources, such as text documents and multimedia information, can provide insights extracted from them using AI techniques, such as natural language processing and picture identification.

Problems with data quality, integration, scalability, and computing demand hamper extracting significant data values. Data quality, completeness, and consistency must be guaranteed to provide trustworthy insights. Data harmonisation and integration are complicated by integrating many data sources, including structured and unstructured data [41]. Advanced computing infrastructure and techniques are required to scale up data analysis operations to accommodate the volume and velocity of big data.

2.5 Harnessing the Potential of Big Data

Big data offers organisations in a variety of industries both opportunities and problems. Utilising the potential of big data may open up a wealth of options for organisations, even though managing the amount, velocity, and diversity of data can be challenging. Making decisions based on data and deriving useful insights represent a significant potential. Organisations may use enormous amounts of data to analyse patterns, trends, and correlations to comprehend better their consumers, market

dynamics, and operational procedures. Then, by applying this knowledge, company plans can be optimised, operational effectiveness can be increased, and customer experiences can be improved.

Big data also makes way for new forms of innovation and income. Organisations may find previously undiscovered patterns and trends using sophisticated analytics approaches like machine learning and predictive modelling. This allows them to discover new market possibilities, create individualised goods and services, and conduct focussed marketing efforts. In addition, big data enables the investigation of unstructured and semi-structured data sources, including posts on social media, client evaluations, and sensor data, which can offer insightful information for product creation, sentiment analysis, and real-time decision-making [42].

Now, the opportunities of big data will be discussed in detail.

2.5.1 Advanced Analytics and Machine Learning Opportunities

Advanced analytics and machine learning provide tremendous prospects in big data analytics. These methods allow businesses to dive deeper into their data, find intricate patterns, and make precise forecasts. Companies can extract important insights from massive datasets using modern analytics methods like clustering, classification, and regression. On the other hand, machine learning algorithms enable the creation of predictive models that can foretell future events and spot abnormalities. These possibilities allow businesses to streamline various business processes, including customer segmentation, fraud detection, inventory forecasting, and supply chain management. Additionally, advanced analytics and machine learning may be used in various fields, such as health care, finance, marketing, and cybersecurity, to promote innovation, improve decision-making, and raise overall corporate performance [43]. Organisations can open new doors for development, innovation, and competitive advantage by leveraging the power of advanced analytics and machine learning.

2.5.1.1 Harnessing Predictive and Prescriptive Analytics

A key component of big data analytics is utilising predictive and prescriptive analytics. Making accurate forecasts about upcoming events or outcomes requires using historical data, statistical algorithms, and machine learning approaches. Organisations may spot future opportunities, predict market trends, and streamline their decision-making procedures by analysing patterns and trends in massive databases. With these predictive insights, businesses may proactively handle issues, allocate resources more effectively, and increase operational effectiveness. On the other hand, prescriptive analytics goes a step further by offering useful suggestions and advice for making decisions based on predictive models [44]. Organisations can make decisions that lead to the intended business outcomes by considering various alternative situations and their potential effects. Organisations are equipped to make data-driven choices, reduce risks, and exploit opportunities in real time by integrating predictive

and prescriptive analytics. They can streamline business operations, raise customer satisfaction, and gain a competitive advantage in today's data-driven environment.

2.5.1.2 Implementing Machine Learning Algorithms

Algorithms for machine learning are created to learn automatically from massive amounts of data, spot patterns, and make predictions or judgements without being explicitly programmed. These algorithms are essential for obtaining insightful information from complex and unstructured data. Organisations may create models that can precisely identify, forecast, or cluster new data instances by training these algorithms on past data. Data preparation, feature selection or extraction, model training, and assessment are just a few of the processes involved in putting machine learning algorithms into practice [45]. Based on the characteristics of their data and the particular issue they hope to resolve, organisations must carefully choose the best algorithm.

2.5.1.3 Discovering Patterns and Trends Through Data Mining

Data mining is a key big data analytics approach focussing on unearthing important patterns and trends buried inside actual, complicated databases. Organisations may get valuable insights from enormous volumes of data by using sophisticated algorithms and statistical approaches. Using traditional data analysis techniques, organisations may use data mining to find patterns, connections, and correlations that would not be visible. It incorporates several procedures, including feature selection, exploratory data analysis, data preparation, and model construction. Organisations can discover useful information through data mining, such as patterns in consumer behaviour, market trends, fraud detection patterns, or forecasting patterns for future events [46]. This knowledge can be utilised to reduce risks, create focussed marketing efforts, streamline corporate operations, and improve customer experiences. Association rule mining, classification, clustering, and regression are a few data-mining techniques that may extract useful information from large volumes of data. Organisations may gain a competitive edge, spur innovation, and open up new opportunities in various fields and businesses by using the potential of data mining.

2.5.1.4 Enabling Real-Time Decision-Making with AI-Driven Analytics

AI-driven analytics is crucial for creating real-time decision-making systems in today's big data world. Organisations can handle and analyse huge amounts of data in real time, gleaning useful insights and patterns instantly by utilising the power of AI and machine learning algorithms. This makes it possible for decision-makers to act quickly and precisely when making informed, data-driven judgements. NLP, deep learning, and cognitive computing are popular AI-driven approaches that automatically extract useful information from various data sources, including text, photos, and videos [47]. Through real-time streaming data analysis, organisations may monitor and react to dynamic events and shifting situations in their operating environments.

Through these capabilities, businesses can discover abnormalities, forecast trends, spot new opportunities, and reduce risks in real time. In addition, AI-driven analytics can automate decision-making by offering sage advice and insights based on past data and predictive modelling [48]. This increases outcomes in terms of overall accuracy, precision, decision-making speed, and efficiency.

2.5.1.5 Unlocking the Potential of Deep Learning and Neural Networks

An important development in data analytics is the possibility of deep learning and neural networks, which will completely change how intricate patterns and connections in vast datasets are found. Deep learning, a branch of machine learning, is distinguished by its capacity to build hierarchical data representations using many artificial neural network (ANN) layers [49]. Deep learning models may autonomously extract detailed characteristics and recognise sophisticated patterns from unstructured and high-dimensional information by utilising the computing capacity of contemporary hardware and algorithms. This innovative technique has displayed outstanding performance in several fields, including computer vision, natural language processing, and speech recognition. Neural networks are the core of deep learning, imitating the structure and operation of the human brain to process and understand massive volumes of data effectively [50]. Neural networks enable the discovery of subtle dependencies, complicated interactions, and nonlinear correlations that conventional statistical models may find challenging to identify because of their capacity to learn from experience and adapt. Organisations can access a full toolkit for advanced data analysis, prediction, and decision-making, promoting innovation and generating revolutionary insights in the age of big data by realising the potential of deep learning and neural networks.

2.5.2 Data Visualisation and Communication Opportunities

One of the most important aspects of the big data environment is the data visualisation and communication options, which provide insightful information and enable the efficient transmission of detailed information. Data visualisation helps the natural study and understanding of enormous and varied datasets via visual representations, such as charts, graphs, and interactive dashboards. It allows for recognising trends, patterns, and outliers while giving a thorough picture of the underlying data. The distance between technical specialists and non-technical stakeholders is also bridged through data visualisation, which also acts as a potent communication tool. It improves the accessibility and impact of data-driven insights by making data visually appealing and simple to grasp, assisting in making well-informed decisions in various contexts. Users can interactively examine and edit data, allowing real-time exploration and discovery, thanks to the development of powerful analytics tools and interactive visualisation techniques. The seamless sharing and dissemination of visualisations across teams and organisations is another way data visualisation fosters collaboration and knowledge exchange. Organisations can unleash the full potential

of their big data assets by embracing the opportunities given by data visualisation and communication. This will enable stakeholders to make data-informed choices and spur innovation in a quickly changing digital environment.

2.5.2.1 Visualising and Presenting Complex Data

A key component of big data is the visualisation and presentation of complex data, which tries to convert convoluted and massive information into understandable and approachable representations. To portray complicated relationships, patterns, and trends within the data includes using various visualisation tools, including charts, graphs, maps, and interactive visualisations [51]. Organisations can improve data exploration, analysis, and understanding by visually depicting data. A strong data presentation also guarantees that big data analytics findings are properly conveyed to various stakeholders, including technical and non-technical ones. This encourages collaboration across teams and departments, information exchange, and the ability to make well-informed decisions. Furthermore, innovations in visualisation tools and technologies, including augmented and virtual reality, offer new methods to engage with and visualise complex data, opening up new vistas for investigation and comprehension. Organisations can unlock the full potential of their big data assets by concentrating on the art and science of visualising and presenting complex data, encouraging innovation, and obtaining a competitive edge in the age of information overload.

2.5.2.2 Storytelling with Data to Drive Actionable Insights

The art of storytelling and the analytical power of big data are combined in the powerful method of "*storytelling with data to drive actionable insights*." It entails creating engaging stories based on data to convey insights and motivate action effectively [52]. Organisations can provide their audience with meaningful and lasting experiences by presenting data in a narrative format, enabling people to relate to the data emotionally and comprehend its consequences. Decision-makers may derive useful insights and make well-informed decisions using storytelling to make complex data accessible, interesting, and simpler to understand. Effective data storytelling entails identifying the most pertinent data points, organising the tale logically and cogently, and relying on visualisation methods to aid the storytelling procedure. It aims to express the underlying narrative and message that the data carries and goes beyond merely providing facts and statistics. By using the power of storytelling, businesses can realise the full potential of their big data by turning it into actionable insights that support innovation, drive strategic choices, and establish a data-driven culture inside the company.

2.5.2.3 Interactive and Exploratory Data Visualisation Techniques

By enabling users to interact actively with the data and derive valuable insights, interactive and exploratory data visualisation techniques play a significant role in

unleashing the potential of big data. These methods enable dynamic manipulation and data exploration, enabling a greater comprehension of intricate linkages and patterns. Users can interactively explore various data elements by zooming in on certain information or out to gain a comprehensive perspective thanks to interactive features like filtering, zooming, and drill-down capabilities. In addition to facilitating hypothesis testing and data exploration, exploratory visualisation approaches allow users to analyse and visualise the data from various viewpoints, revealing hidden trends or outliers that would not have been visible otherwise [51]. Real-time data interaction encourages exploration and allows consumers to ponder fresh ideas and viewpoints. Additionally, interactive visualisations improve insight communication by enabling users to customise the presentation to their audience, resulting in a more compelling and personalised storytelling experience. Organisations can use interactive and exploratory data visualisation approaches to foster discovery and innovation, ease collaborative decision-making processes, and obtain deeper insights from their large data.

2.5.2.4 Communicating Insights to Stakeholders and Decision-Makers

Providing insights to decision-makers and stakeholders is crucial to utilise big data properly. To ensure their effectiveness and support well-informed decision-making, it is essential to convey insightful data findings clearly, succinctly, and engagingly. Various tools can be used, including data visualisations, reports, dashboards, and presentations, as messages must be tailored to the target audience and convey detailed information clearly and practically to be effective. For instance, visualisations make complicated patterns and trends visible to stakeholders at a glance, making the insights more approachable and memorable. Reports and dashboards offer thorough summaries and analyses, giving the most important conclusions and suggestions in a well-organised manner [53]. Presentations allow users to share their views face to face and encourage dialogue and clarity. Storytelling approaches are also used to engage the audience and effectively convey insights. Data is presented narratively, including arresting images, amusing tales, and practical examples. By using effective and clear communication techniques, organisations may guarantee that big data insights have a significant impact. This will enable stakeholders and decision-makers to make well-informed decisions, leading to favourable results.

2.5.2.5 Augmented Reality and Virtual Reality for Data Visualisation

The innovative technologies, Augmented Reality (AR) and Virtual Reality (VR), hold great promise for massive data visualisation, where VR immerses users in a virtual world, while AR enhances the real-world experience by overlaying digital data [54]. These technologies provide the extraordinary potential to improve the comprehension and study of intricate datasets when used for data visualisation. Through AR, data visualisations can be immediately projected onto physical objects or surroundings, providing users with a more intuitive and immersive way to interact with and manage the visualisations. One's understanding of the spatial relationships and

patterns in the data can be considerably improved by doing this. Contrarily, VR enables users to enter a virtual setting where they may explore and engage with data visualisations in a three-dimensional world. The sensation of presence offered by this immersive experience can encourage greater engagement with and comprehension of the facts. Furthermore, independent of a user's physical location, AR and VR may offer cooperative data visualisation experiences, allowing real-time interaction and sharing of insights. Utilising the possibilities of AR and VR, businesses may open new vistas for data visualisation, providing users with more immersive, dynamic, and captivating experiences that will eventually increase user comprehension of the data, decision-making, and problem-solving.

2.5.3 Future Directions and Emerging Trends

The techniques and methods of big data analytics are continuously changing, influencing many new trends and directions. One of the important trends is performing and implementing machine learning and AI techniques. AI and ML algorithms may reveal deeper insights and hidden patterns within enormous datasets, allowing for more precise forecasts and improved decision-making [55]. In addition to these areas, edge computing, which includes processing and analysing data closer to its source to reduce latency and enable real-time analytics, is another developing trend. This method is on-demand, while low latency and high availability have high priority.

The development of quantum computing also opens intriguing possibilities for big data analytics since quantum algorithms are more equipped than conventional computer approaches to handle difficult optimisation and pattern recognition problems. The ethical issues surrounding big data, such as data privacy, security, and bias, are also receiving a lot of attention, which has prompted the creation of frameworks and rules to ensure data's ethical and responsible use [56]. These new trends and directions will greatly influence the future of analytics as big data continues to develop, opening up fresh opportunities for data value extraction and fostering innovation across various industries.

Learning Outcomes

- **Understanding the Definition of Big Data:** Defining the concept of big data and its significance in modern data analytics and exploring the key characteristics distinguishing big data from traditional datasets.
- **Exploring the 5 Vs of Big Data:** Comprehending the five essential dimensions of big data—Volume, Value, Variety, Velocity, and Veracity—and analysing how each V contributes to the challenges and opportunities presented by big data.
- **Identifying Challenges in Big Data:** Recognising the major challenges associated with big data, including data collection, storage, quality, integrity, privacy, and security issues, and understanding the difficulties in extracting valuable insights from large and diverse datasets.

- **Addressing Challenges and Harnessing Potential:** Exploring strategies for overcoming challenges in big data, particularly in data collection, storage, quality, and privacy, and investigating opportunities for harnessing the potential of big data through advanced analytics, machine learning, data visualisation, and communication.
- **Exploring Future Directions and Emerging Trends:** Envisioning the future landscape of big data analytics and emerging trends, and understanding the role of advanced analytics, machine learning, and data visualisation in shaping the future of big data applications.

References

1. G.S. Aujla, R. Chaudhary, N. Kumar, A.K. Das, J.J.P.C. Rodrigues, Secsva: secure storage, verification, and auditing of big data in the cloud environment. IEEE Commun. Mag. **56**(1), 78–85 (2018)
2. N. Khan, A. Naim, M.R. Hussain, Q.N. Naveed, N. Ahmad, S. Qamar, The 51 v's of big data: survey, technologies, characteristics, opportunities, issues and challenges, in *Proceedings of the International Conference on Omni-layer Intelligent Systems* (2019), pp. 19–24
3. R. Chaudhary, G.S. Aujla, N. Kumar, J.J. Rodrigues, Optimized big data management across multi-cloud data centers: software-defined-network-based analysis. IEEE Commun. Mag. **56**(2), 118–126 (2018)
4. Total data volume worldwide 2010-2025. Statista. [Online]. https://www.statista.com/statistics/871513/worldwide-data-created/
5. S. Garg, A. Singh, K. Kaur, G.S. Aujla, S. Batra, N. Kumar, M.S. Obaidat, Edge computing-based security framework for big data analytics in VANETs. IEEE Netw. **33**(2), 72–81 (2019)
6. S. Sagiroglu, D. Sinanc, Big data: a review, in *2013 International Conference on Collaboration Technologies and Systems (CTS)*. IEEE (2013), pp. 42–47
7. D. Kaur, G.S. Aujla, N. Kumar, A.Y. Zomaya, C. Perera, R. Ranjan, Tensor-based big data management scheme for dimensionality reduction problem in smart grid systems: SDN perspective. IEEE Trans. Knowl. Data Eng. **30**(10), 1985–1998 (2018)
8. How much information is generated and stored in the world? Fundación MAPFRE. [Online]. https://www.fundacionmapfre.org/en/blog/how-much-information-is-generated-and-stored-in-the-world/
9. Internet users and penetration worldwide. Insider Intelligence. [Online]. https://www.insiderintelligence.com/chart/212610/internet-users-penetration-worldwide-2016-2021-billions-change-of-population
10. N. Elgendy, A. Elragal, Big data analytics: a literature review paper, in *Advances in Data Mining. Applications and Theoretical Aspects: 14th Industrial Conference, ICDM 2014*, St. Petersburg, Russia, July 16–20, 2014. Proceedings 14. Springer (2014), pp. 214–227
11. K. Vassakis, E. Petrakis, I. Kopanakis, Big data analytics: applications, prospects and challenges. *Mobile Big Data: A Roadmap from Models to Technologies* (2018), pp. 3–20
12. N. Deepa, Q.-V. Pham, D.C. Nguyen, S. Bhattacharya, B. Prabadevi, T.R. Gadekallu, P.K.R. Maddikunta, F. Fang, P.N. Pathirana, A survey on blockchain for big data: approaches, opportunities, and future directions. Future Gener. Comput. Syst. **131**, 209–226 (2022)
13. J. Ranjan, C. Foropon, Big data analytics in building the competitive intelligence of organizations. Int. J. Inf. Manag. **56**, 102231 (2021)

14. H. Nozari, M. Fallah, H. Kazemipoor, S.E. Najafi, Big data analysis of IOT-based supply chain management considering FMCG industries, -, vol. 15, no. 1 (eng) (2021), pp. 78–96
15. Q.A. Nisar, N. Nasir, S. Jamshed, S. Naz, M. Ali, S. Ali, Big data management and environmental performance: role of big data decision-making capabilities and decision-making quality. J. Enterp. Inf. Manag. **34**(4), 1061–1096 (2021)
16. T. Nijhawan, G. Attigeri, T. Ananthakrishna, Stress detection using natural language processing and machine learning over social interactions. J. Big Data **9**(1), 1–24 (2022)
17. A. Davoudian, M. Liu, Big data systems: a software engineering perspective. ACM Comput. Surv. (CSUR) **53**(5), 1–39 (2020)
18. A. Sharma, H. Pandey, Big data and analytics in industry 4.0. *A Roadmap to Industry 4.0: Smart Production, Sharp Business and Sustainable Development* (2020), pp. 57–72
19. R.A.A. Habeeb, F. Nasaruddin, A. Gani, I.A.T. Hashem, E. Ahmed, M. Imran, Real-time big data processing for anomaly detection: a survey. Int. J. Inf. Manag. **45**, 289–307 (2019)
20. M.H. Javed, X. Lu, D.K. Panda, Characterization of big data stream processing pipeline: a case study using flink and kafka, in *Proceedings of the Fourth IEEE/ACM International Conference on Big Data Computing, Applications and Technologies* (2017), pp. 1–10
21. I. Taleb, M.A. Serhani, R. Dssouli, Big data quality: a survey, in *2018 IEEE International Congress on Big Data (BigData Congress)*. IEEE (2018), pp. 166–173
22. A. Vogelsang, M. Borg, Requirements engineering for machine learning: perspectives from data scientists, in *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*. IEEE (2019), pp. 245–251
23. R. Mahanti, R. Mahanti, *Data Governance and Compliance* (Springer, 2021)
24. M.K. Saggi, S. Jain, A survey towards an integration of big data analytics to big insights for value-creation. Inf. Process. Manag. **54**(5), 758–790 (2018)
25. M.J. Marquardt, S. Banks, P. Cauwelier, N.C. Seng, *Optimizing the Power of Action Learning: Real-Time Strategies for Developing Leaders, Building Teams and Transforming Organizations* (Hachette, UK, 2018)
26. N.J. Ogbuke, Y.Y. Yusuf, K. Dharma, B.A. Mercangoz, Big data supply chain analytics: ethical, privacy and security challenges posed to business, industries and society. Prod. Plann. Control **33**(2–3), 123–137 (2022)
27. G.S. Aujla, A. Jindal, D.B. Rawat, C. Jiang, Deep neuro-fuzzy analytics for intelligent big data processing in smart ecosystems. *Neural Computing and Applications* (2023), pp. 1–3
28. E. Ahmed, I. Yaqoob, I.A.T. Hashem, I. Khan, A.I.A. Ahmed, M. Imran, A.V. Vasilakos, The role of big data analytics in internet of things. Comput. Netw. **129**, 459–471 (2017)
29. M. Younan, E.H. Houssein, M. Elhoseny, A.A. Ali, Challenges and recommended technologies for the industrial internet of things: a comprehensive review. Measurement **151**, 107198 (2020)
30. K. Shvachko, H. Kuang, S. Radia, R. Chansler, The Hadoop distributed file system, in *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*. IEEE (2010), pp. 1–10
31. S. Campana, Accuracy, precision and quality control in age determination, including a review of the use and abuse of age validation methods. J. Fish Biol. **59**(2), 197–242 (2001)
32. S. Duggineni, Impact of controls on data integrity and information systems. Sci. Technol. **13**(2), 29–35 (2023)
33. M. Barati, G.S. Aujla, J.T. Llanos, K.A. Duodu, O.F. Rana, M. Carr, R. Ranjan, Privacy-aware cloud auditing for GDPR compliance verification in online healthcare. IEEE Trans. Ind. Inform. **18**(7), 4808–4819 (2022)
34. R. Kitchin, Getting smarter about smart cities: improving data privacy and data security (2016)
35. A. Gulati, G.S. Aujla, R. Chaudhary, N. Kumar, M.S. Obaidat, Deep learning-based content centric data dissemination scheme for internet of vehicles, in *2018 IEEE International Conference on Communications (ICC)* (2018), pp. 1–6
36. G.S. Aujla, M. Singh, N. Kumar, A.Y. Zomaya, Stackelberg game for energy-aware resource allocation to sustain data centers using res. IEEE Trans. Cloud Comput. **7**(4), 1109–1123 (2019)

37. D. Marikyan, J. Llanos, M. Barati, G. Aujla, Y. Li, K. Adu-Duodu, S. Tahir, O. Rana, S. Papagiannidis, R. Ranjan, M. Carr, Privacy & cloud services: are we there yet? in *2021 IEEE International Conference on Service-Oriented System Engineering (SOSE)* (2021), pp. 11–19
38. H. Ahmad, G.S. Aujla, GDPR compliance verification through a user-centric blockchain approach in multi-cloud environment. Comput. Electr. Eng. **109**, 108747 (2023)
39. G. Singh Aujla, M. Barati, O. Rana, S. Dustdar, A. Noor, J.T. Llanos, M. Carr, D. Marikyan, S. Papagiannidis, R. Ranjan, Com-pace: compliance-aware cloud application engineering using blockchain. IEEE Internet Comput. **24**(5), 45–53 (2020)
40. X. Shu, Y. Ye, Knowledge discovery: methods from data mining and machine learning. Social Sci. Res. **110**, 102817 (2023)
41. S. Al-Yadumi, T.E. Xion, S.G.W. Wei, P. Boursier, Review on integrating geospatial big datasets and open research issues. IEEE Access **9**, 10 604–10 620 (2021)
42. F. Amalina, I.A.T. Hashem, Z.H. Azizul, A.T. Fong, A. Firdaus, M. Imran, N.B. Anuar, Blending big data analytics: review on challenges and a recent study. IEEE Access **8**, 3629–3645 (2019)
43. J.P. Bharadiya, Leveraging machine learning for enhanced business intelligence. Int. J. Comput. Sci. Technol. **7**(1), 1–19 (2023)
44. T. Susnjak, A prescriptive learning analytics framework: beyond predictive modelling and onto explainable AI with prescriptive analytics, arXiv preprint arXiv:2208.14582 (2022)
45. J. Brownlee, *Data Preparation for Machine Learning: Data Cleaning, Feature Selection, and Data Transforms in Python* (Machine Learning Mastery, 2020)
46. M.A. Khder, I.A. Abu-Alsondos, A.Y. Bahar, The impact of implementing data mining in business intelligence. Int. J. Entrep. **25**, 1–7 (2021)
47. P. Tadejko, Cloud cognitive services based on machine learning methods in architecture of modern knowledge management solutions. *Data-Centric Business and Applications: Towards Software Development (Volume 4)* (2020), pp. 169–190
48. S. Gupta, A. Leszkiewicz, V. Kumar, T. Bijmolt, D. Potapov, Digital analytics: modeling for insights and new methods. J. Interact. Mark. **51**(1), 26–43 (2020)
49. I.H. Sarker, Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions. SN Comput. Sci. **2**(6), 420 (2021)
50. Y. Matsuo, Y. LeCun, M. Sahani, D. Precup, D. Silver, M. Sugiyama, E. Uchibe, J. Morimoto, Deep learning, reinforcement learning, and world models. Neural Netw. **152**, 267–275 (2022)
51. A. Protopsaltis, P. Sarigiannidis, D. Margounakis, A. Lytos, Data visualization in internet of things: tools, methodologies, and challenges, in *Proceedings of the 15th International Conference on Availability, Reliability and Security* (2020), pp. 1–11
52. Y. Zhang, M. Reynolds, A. Lugmayr, K. Damjanov, G.M. Hassan, A visual data storytelling framework, in *Informatics*, vol. 9, no. 4 (MDPI, 2022), p. 73
53. K. Börner, A. Bueckle, M. Ginda, Data visualization literacy: definitions, conceptual frameworks, exercises, and assessments. *Proceedings of the National Academy of Sciences*, vol. 116, no. 6 (2019), pp. 1857–1864
54. G.A. Koulieris, K. Akşit, M. Stengel, R.K. Mantiuk, K. Mania, C. Richardt, Near-eye display and tracking technologies for virtual and augmented reality, in *Computer Graphics Forum*, vol. 38, no. 2 (Wiley Online Library, 2019), pp. 493–519
55. C. Li, Y. Chen, Y. Shang, A review of industrial big data for decision making in intelligent manufacturing. Eng. Sci. Technol. Int. J. **29**, 101021 (2022)
56. S.S. Gill, A. Kumar, H. Singh, M. Singh, K. Kaur, M. Usman, R. Buyya, Quantum computing: a taxonomy, systematic review and future directions. Softw. Pract. Experience **52**(1), 66–114 (2022)

Further Reading

57. I. Yaqoob, I.A.T. Hashem, A. Gani, S. Mokhtar, E. Ahmed, N.B. Anuar, A.V. Vasilakos, Big data: from beginning to future. Int. J. Inf. Manag. **36**(6), 1231–1247 (2016)
58. A.K. Bhadani, D. Jothimani, Big data: challenges, opportunities, and realities. *Effective Big Data Management and Opportunities for Implementation* (2016), pp. 1–24

Big Data Analytics

3

Without big data analytics, companies are blind and deaf, wandering out onto the web like deer on a freeway.

—Geoffrey Moore

This chapter introduces the dynamic domain of big data analytics, illuminating its multifaceted aspects and profound significance. It commences by furnishing a comprehensive definition of big data analytics and delves into the taxonomy of this discipline, encompassing descriptive, diagnostic, predictive, prescriptive, and cognitive analytics, each underscored by its distinctive applications. Furthermore, this chapter elucidates the manifold advantages that big data analytics affords, notably its pivotal role in bolstering risk management, effecting cost reduction, facilitating informed decision-making, and catalysing advancements in product development. In parallel, it conscientiously scrutinises the challenges endemic to this field, encompassing the dearth of proficient practitioners, misconceptions, concerns about escalating data volumes, intricacies associated with tool selection, and the salient issues of data security and privacy. The essential stages inherent to big data analytics are methodically expounded to facilitate a comprehensive understanding, encompassing data acquisition, preprocessing, storage, and analysis, thereby furnishing a nuanced appreciation of the foundational principles and intricate nuances intrinsic to this pivotal discipline.

Ü. Demirbaga et al., *Big Data Analytics*,
https://doi.org/10.1007/978-3-031-55639-5_3

3.1 What Is Big Data Analytics?

Big data analytics studies enormous and complex datasets of structured, unstructured and semi-structured data, particularly datasets from new data sources, to uncover hidden patterns and correlations by applying advanced analytical techniques such as regression, clustering, classification, and correlation. Along with these statistical analysis techniques, other data analytics methods are applied with the help of the latest system big data tools to datasets that are too large or complex to be dealt with by traditional methods. Since the early 2000s, when technology revolutionised, many technological devices, from smartphones to smart home devices, have contributed to the creation of big data [1]. This data explosion led to the development of Hadoop, Spark, and NoSQL projects. Big companies, mainly in marketing, have focussed on big data analysis to increase their profits. Data engineers have focussed on looking for an answer to the question: "How to extract more meaningful insights from large and complex data readily and quickly?". Emerging technologies such as machine learning are frequently used in big data analysis techniques to discover complex insights from data and make faster and more accurate decisions. In recent years, many companies, including cloud companies, have implemented different data analytics systems, some of which are paid, to store and analyse batch and streaming data. Apache Hadoop, the most popular one, is available as an open-source that provides many powerful projects for storing data securely and analysing it efficiently.

3.2 The Types of Big Data Analytics

Big data analytics applications are descriptive, diagnostic, predictive, prescriptive, and cognitive. Figure 3.1 elaborates the types of big data analytics with their features.

3.2.1 Descriptive Analytics

Descriptive analytics is the first step in analysing big datasets by performing simple mathematical operations to reveal essential existing data patterns. It seeks to answer "what happened" by summarising the raw data in a human-understandable format. Descriptive analytics scrutinise company data to perform statistical analyses in day-to-day Business Intelligence, such as a company's production, product sales, and customer purchasing preferences. It also uses aggregation functions and methods to reveal complex relationships between variables. Graphs are commonly used to visualise data to make them more understandable.

Use case: A chemical company, called Dow, analysed historical data to see how it could improve plant utilisation in its offices and laboratories. Using descriptive analytics, the company identified underutilised space, saving approximately $4 million per year [2].

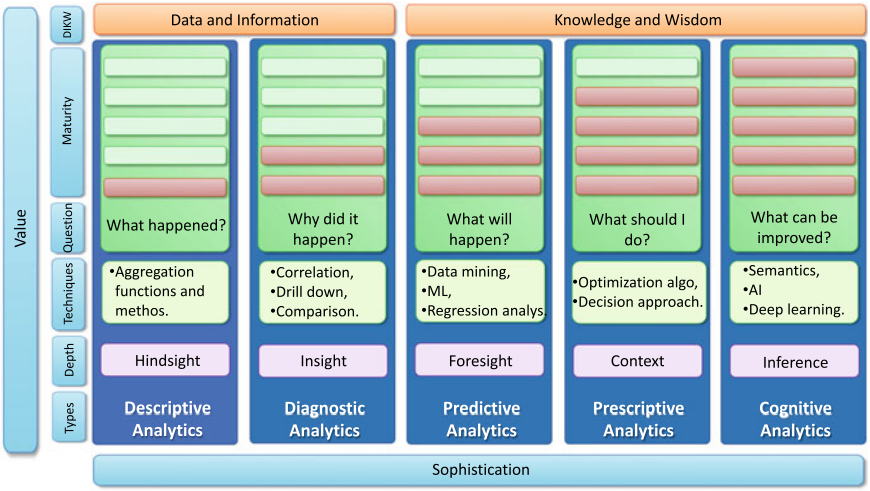


Fig. 3.1 A comprehensive look at the types of big data analytics

3.2.2 Diagnostic Analytics

Diagnostic analytics reviews data about a past event or situation to understand what caused a problem. Organisations gain in-depth insight into the case by using correlation, drill-down, and comparison techniques to identify customer trends. This technique, which data scientists frequently use, focusses on causal relationships and sequences.

Use case: HelloFres, the meal kit subscription company, collects millions of data points from consumers worldwide, including geographic location, demographic data, food type, taste preferences, and normal ordering cadence and timing. The HelloFresh team analyses this information to look for consumer characteristics and behaviour patterns. As an example, let's say the HelloFresh team notices an increase in orders for fish-based recipes. After completing the diagnostic analysis, they discovered that identifying as female and residing in the Northeastern United States were the traits most strongly associated with ordering fish recipes [3].

3.2.3 Predictive Analytics

After finding the answer to what happened in the past, organisations use predictive analytics to predict what will happen. This analytical method analyses big data from different sources using data mining, machine learning, and regression analysis to predict future trends or events. Predictive analytics predict the future state of organisations and plan for the future. Examples of predictive analytics include best offers, churn risk analysis, etc.

Use case: Purdue University estimated academic and behavioural concerns using big data analytics [4]. Using data from various sources, such as student information and course management systems, the system predicts and generates a risk profile for each student, indicating how far a student succeeds in a course and labels the risk levels as green (high probability of success), yellow (potential problems), and red (risk of failure).

3.2.4 Prescriptive Analytics

Prescriptive analytics helps organisations evaluate the impact of different possible decisions by asking, "How can we make it happen?" and "What should we do to make it happen?". Prescriptive analytics aims to develop the best potential suggestions for a scenario based on what the analyst can deduce from the available data.

Use case: Health care is one of the industries which benefits the most from prescriptive analytics. Prescriptive analytics can recommend diagnoses and treatments to physicians for the treatment of similar newcomers by analysing past medication and timelines administered to patients. Aurora Health Care Center applies big data analytics to recommend the best possible treatment to doctors, resulting in savings of approximately six million USD per year [5].

3.2.5 Cognitive Analytics

It combines next-generation technologies such as cognitive analytics, semantics, artificial intelligence, and deep learning to create and use human-like intelligence for specific tasks. The intelligence created through this analytics can become smarter and more effective by learning new patterns from new data and interacting with people.

Use case: Many organisations use cognitive analytics to leverage unstructured data sources such as emails, text documents, and social media content. Microsoft's Cortana, Apple's Siri, and IBM's Watson monitor customer behaviour and trends using cognitive analytics [6].

3.3 The Advantages of Big Data Analytics

Big data analytics is a subset of business intelligence that uses advanced analytical tools to analyse enormous amounts of data from various sources, including social media, the web, and databases. Companies analyse it in depth to identify patterns and patterns within it. By this way, businesses gain leverage from their advertising and marketing dollars. Some of these advantages are listed below.

3.3.1 Risk Management

Big data analytics provides critical insights into consumer behaviour and market trends that help businesses evaluate their position and progress. Moreover, it helps companies identify and anticipate risks that could harm their business regarding financial risks. With the prevalence of cybercrime, big data analytics allows businesses to identify patterns that point to a potential cybersecurity threat.

3.3.2 Cost Reduction

Big data analytics provides insights that help businesses reduce business and improve operational efficiency. Quality assurance and testing processes may be complicated, particularly in biopharmaceuticals and nanotechnologies. Big data analytics can reveal the effects of many variables in the manufacturing process, allowing businesses to make better decisions.

3.3.3 Advanced Data-Driven Decision-Making

Big data analytics helps make predictions by analysing past data. Thus, businesses can make better decisions for the current situation and prepare for the future, giving them a competitive advantage and providing a more agile framework for decision-making and risk management.

3.3.4 Improving New Product Development

Businesses can launch better products in the future by analysing historical data about product launches and customer feedback using big data analytics. In addition, they can understand real-time market analysis, changes in the supply and demand of business products, and changes in consumer behaviour that help customer-focussed marketing. The growing demand for personalised services can also be strengthened by analysing consumer needs, preferences, and purchasing behaviour.

3.4 The Challenges of Big Data Analytics

3.4.1 Lack of Knowledge Professionals

The shortage of specialists with the appropriate knowledge and abilities is one of the biggest problems facing big data analytics. It takes specific knowledge and skills in statistical analysis, data mining, machine learning, and programming to analyse

vast and complicated datasets. It is challenging for firms to attract and retain skilled individuals for their big data analytics initiatives due to a lack of experts with these capabilities.

Several factors cause the absence of big data analytics experts [7]. First of all, big data analytics is a relatively new area, and few school programmes concentrate on it. There are not enough experts with the required education and training to fulfil the need for qualified workers. Second, new technologies and methodologies are emerging quickly in big data analytics, which is constantly changing. Because of this, professionals must continuously upgrade their knowledge and abilities to stay abreast of new advancements, which may be difficult and time-consuming. Lastly, there is fierce rivalry for qualified experts in big data analytics since businesses from various sectors understand how vital analytics are to achieving success in the marketplace. Due to the strong demand for competent experts, companies must entice them with appealing compensation plans and advancement prospects.

Ultimately, overcoming the difficulty of a skills gap in big data analytics involves a mix of deliberate investments in training and development, cooperation with educational institutions, and partnership with outside service providers. Organisations may position themselves to benefit from big data analytics and propel business success by following these actions.

3.4.2 Misunderstanding of Big Data

The misunderstanding of big data is a significant barrier to big data analytics. Big data, its applications, and the advantages it may provide are frequently unclear. Consequently, people may have inflated expectations and misinterpret data analysis findings, which might eventually reduce the efficacy of big data analytics.

The notion that more data is always better is among the most significant misunderstandings about big data. Although having access to a lot of data might be helpful, it is not always required or enough to provide insightful knowledge. In some circumstances, a smaller, more robust dataset could be better suited for resolving particular queries or tackling specific business difficulties. Another misunderstanding about big data is that technology can handle any analytical problem independently. Technology is essential to big data analytics, but it is only one component. For big data analytics to be effective, experts must understand the data, spot patterns and trends, and produce pertinent and valuable insights. Additionally, it's a common misunderstanding that big data analytics is a one-time procedure that produces conclusive results. Big data analytics is an ongoing, iterative process that needs to be constantly monitored and modified to take account of changes in data sources, organisational objectives, and market circumstances.

Addressing the challenge of misunderstanding big data necessitates a mix of education, strategic planning, and continuous improvement culture. By following these measures, organisations may harness the benefits of big data analytics and promote economic success.

3.4.3 Data Growth Issues

Data growth is another significant challenge in big data analytics. It gets more challenging and complex to manage and analyse the growing amounts of data businesses produce and gather. This may result from data overload, poor quality, and restricted processing and storage space [8].

Data overload is one of the most critical problems with data increase in big data analytics. Finding and extracting the most pertinent and valuable data for analysis is harder as data quantities rise. As a result, analysts may find themselves inundated with data yet unable to provide genuinely useful and meaningful insights. Data quality issues are a related concern to data growth. Data quality can worsen as data quantities rise, becoming more prone to mistakes, inconsistencies, and inaccuracies. This may result in inaccurate or misleading insights, undermining data analysis outcomes' validity and accuracy. Limitations in storage and processing are another critical problem for big data analytics. Organisations may experience limitations regarding storage space, computing power, and network bandwidth as data quantities rise. This can make it challenging to quickly and effectively store, retrieve, and analyse massive data quantities, which can cause delays and higher costs.

A mix of cutting-edge technology, effective data management procedures, and scalable computing infrastructure are needed to address the data growth problem in big data analytics. By following these actions, businesses may overcome the difficulties presented by data expansion and maximise the benefits of big data analytics.

3.4.4 Confusion on Big Data Tool Selection

The uncertainty that businesses may have while choosing the best solutions for managing and analysing their data is another problem presented by big data analytics. There are various tools on the market, each with a particular set of features and functions [9]. Making the incorrect option can have serious adverse effects, and selecting the wrong instrument can be difficult and time-consuming.

The wide variety of tools on the market is one of the key factors making it challenging to choose the best one for big data analytics. Many tools and technologies are accessible, each with unique features and capabilities. Organisations may struggle to select the technology that best suits their needs. Choosing the best big data tool can be difficult for various reasons, including the fact that these technologies are sometimes sophisticated and require specific skills to utilise correctly. Organisations may need to spend a lot of money on training and development to ensure their personnel can use the tools successfully and provide insightful data.

Clear criteria, training and development, and outside direction are all necessary to overcome the problem of misunderstanding around the choice of big data technology. With the help of these actions, businesses may overcome the difficulties associated with choosing the best technologies for big data analytics and maximise the value of their data.

3.4.5 Data Security and Privacy

Big data analytics provide two crucial problems businesses must handle to preserve and utilise their data morally and responsibly. It is essential to develop robust security measures and privacy rules when firms gather and analyse ever-larger amounts of data since doing so raises the danger of data breaches, cyberattacks, and other security concerns [10].

Ensuring data security across the whole data lifecycle—from collection to storage to analysis—is one of the significant difficulties of big data analytics, which calls for implementing specific security mechanisms that shield data from unwanted access, disclosure, or change [11]. The GDPR[1] in the European Union, the California Consumer Privacy Act (CCPA)[2] in the United States, and other national or regional legislation must all be complied with by organisations to secure customer data. Making sure that data is handled responsibly, ethically, and with the proper privacy regulations and controls to preserve individual privacy rights is another difficulty presented by big data analytics. Organisations must get adequate consent from people whose data is being collected and processed and be open about how they gather and use that data. This necessitates the adoption of efficient data governance rules and practices, such as data anonymisation, pseudonymisation, and other methods that can assist in safeguarding personal information while still allowing for the production of insightful knowledge from the data.

Data security and privacy challenges in big data analytics must be addressed with robust security mechanisms, efficient data governance policies and procedures, and outside counsel and experience. Organisations may overcome the problems with data security and privacy problems and ensure that their data is utilised morally and responsibly by following these measures.

3.5 The Steps of Big Data Analytics

Figure 3.2 depicts the four stages of big data analytics: data acquisition, data preprocessing, data storage, and data analysis, and it shows the technologies used in each stage. These technologies will be discussed in detail in Chap. 5.

3.5.1 Big Data Acquisition

Collecting structured and unstructured large amounts of data from diverse sources is known as big data collection. Big data aggregation is conducted during stream processing such as message queuing operations, publish/subscribe paradigm, or event

[1] https://gdpr-info.eu/.
[2] https://oag.ca.gov/privacy/ccpa.

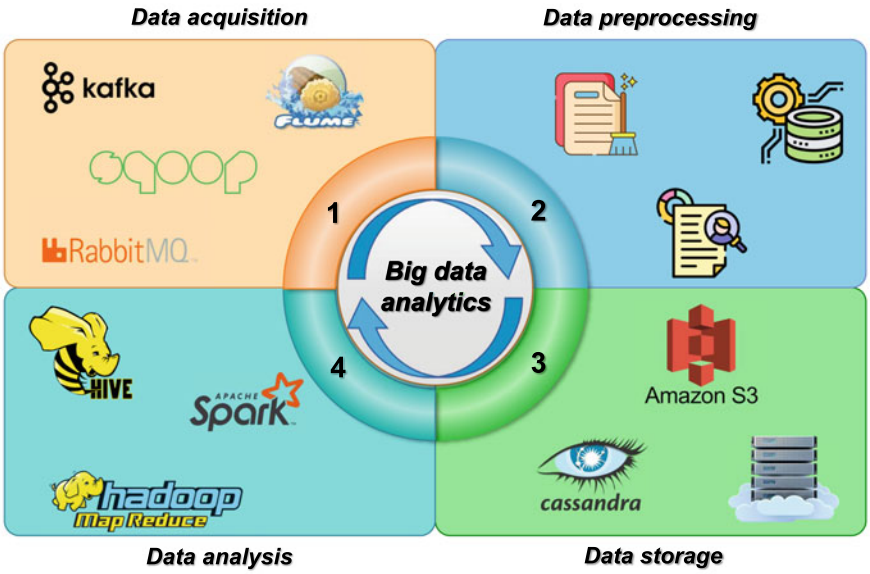


Fig. 3.2 Big data analytics lifecycle

processing paradigm [12]. The new data, produced by the data sources, is sent to the data preprocessing stage through components integrated into the big data acquisition stage (i.e. Apache Kafka,[3] Apache Flume,[4] Apache Sqoop,[5] RabbitMQ.[6] Based on the companies' purposes, this collected raw data can also be sent directly to the data warehouse. Examples of big data sources are structured data, semi-structured data, unstructured data, sensors, real-time devices, mobile phones, social media platforms, companies, and various service organisations such as hospitals.

3.5.2 Big Data Preprocessing

Data preprocessing, the first step in the data knowledge discovery process, is making the collected raw data analysable. Raw data collected from various sources are often flawed and contain inconsistencies and redundancies. Considering the amount of data produced per second, preprocessing is an indispensable part of data analysis. Data preprocessing is the stage of providing the minimum requirements of data analysis techniques on the collected data, which is also a crucial first step in feature extraction since it helps to create cleaner, more manageable, and consistent datasets [13].

[3] https://kafka.apache.org/.
[4] https://flume.apache.org/.
[5] https://sqoop.apache.org/.
[6] https://www.rabbitmq.com/.

Data preprocessing includes the following steps below [14]:

- Data integration,
- Data imputation (managing missing values),
- Categorical encoding (converting categorical data into numerical data; label encoding/one-hot encoding),
- Scaling (MinMax scaling) and normalisation (range 0–1),
- Outlier detection (handling noisy data).

Data preprocessing is also an essential step to enable feature extraction. Feature extraction, also known as dimensionality reduction, is a method for creating a new and smaller set of features from large-scale datasets and is a critical element in creating better machine learning models [15]. Feature extraction has numerous advantages, such as reducing data size, decreasing needed storage, improving prediction accuracy, reducing executing and training time, and improving data visualisation [16].

3.5.3 Big Data Storage

Big data storage systems aim to provide a scalable, secure place for data by meeting applications requiring access to data. An ideal big data storage system should have fast read and write access while enabling the storage of increasing amounts of data and allowing all types of data structures (structured and unstructured). Big data storage systems use a distributed and shared architecture to address the volume problem. This architecture allows new nodes to be added in parallel with increasing data, where new machines can be seamlessly added to the existing data warehouse set. Speed is also one of the essential criteria in big data systems. Storing streaming data and responding to queries without delay is of great importance. In big data systems, it is common to save new data in the database and respond to queries simultaneously. For this reason, companies need to choose databases suitable for their purposes. Some examples can be given for big data storage solutions: HDFS, NoSQL databases (i.e. key-value databases, columnar databases, document-based databases, graph databases), and cloud databases solutions.

3.5.4 Big Data Analysis

Big data analysis is gleaning important patterns and insights from vast quantities of intricate and varied datasets. The goal is to find hidden connections, trends, and important information that conventional data processing methods could miss using cutting-edge analytical techniques and technology. Big data analysis uses techniques and processes like machine learning, data mining, NLP, and predictive analytics to make sense of the enormous volumes of structured and unstructured data that organisations can acquire important insights, make data-driven choices, spot opportunities, streamline operations, and gain a competitive edge in today's data-driven world by releasing the potential of big data.

Learning Outcomes of the Chapter

- **Understanding Big Data Analytics:** Defining the concept of big data analytics and its role in modern data processing.
- **Exploring the Types of Big Data Analytics:** Examining various types, including descriptive analytics, diagnostic analytics, predictive analytics, prescriptive analytics, and cognitive analytics.
- **Analysing the Advantages of Big Data Analytics:** Investigating the benefits, such as risk management, cost reduction, advanced data-driven decision-making, and improving new product development.
- **Recognising the Challenges of Big Data Analytics:** Identifying challenges, including the lack of knowledge professionals, misunderstanding of big data, confusion on tool selection, and data security and privacy concerns.
- **Understanding the Steps of Big Data Analytics:** Exploring the key steps involved in big data analytics, including acquisition, preprocessing, storage, and analysis.

References

1. C. Lutz, Digital inequalities in the age of artificial intelligence and big data. Hum. Behav. Emerg. Technol. **1**(2), 141–148 (2019)
2. E. Siegel, Descriptive, predictive, prescriptive: transforming asset and facilities management with analytics. New Jersey, Hoboken (2016)
3. What is diagnostic analytics? 4 examples. Harvard Business School. [Online]. https://online.hbs.edu/blog/post/diagnostic-analytics
4. Purdue university achieves remarkable results with big data. Datafloq. [Online]. https://datafloq.com/read/purdue-university-achieves-remarkable-results-data/
5. The future of big data? three use cases of prescriptive analytics. Datafloq. [Online]. https://datafloq.com/read/future-big-data-use-cases-prescriptive-analytics/
6. Real-life applications of cognitive analytics. orbit. [Online]. https://www.orbitanalytics.com/cognitive-analytics/
7. P. Russom et al., Big data analytics. *TDWI Best Practices Report, Fourth Quarter*, vol. 19, no. 4 (2011), pp. 1–34
8. D. Bumblauskas, H. Nold, P. Bumblauskas, A. Igou, Big data analytics: transforming data to action. Bus. Process Manag. J. **23**(3), 703–720 (2017)
9. I. Lee, Y.J. Shin, Machine learning for enterprises: applications, algorithm selection, and challenges. Bus. Horiz. **63**(2), 157–170 (2020)
10. S. Garg, K. Kaur, G. Kaddoum, P. Garigipati, G.S. Aujla, Security in IoT-driven mobile edge computing: new paradigms, challenges, and opportunities. IEEE Netw. **35**(5), 298–305 (2021)
11. K. Crawford, J. Schultz, Big data and due process: toward a framework to redress predictive privacy harms. BCL Rev. **55**, 93 (2014)
12. G. Cugola, A. Margara, Processing flows of information: from data stream to complex event processing. ACM Comput. Surv. (CSUR) **44**(3), 1–62 (2012)
13. A. Famili, W.-M. Shen, R. Weber, E. Simoudis, Data preprocessing and intelligent data analysis. Intell. Data Anal. **1**(1), 3–23 (1997)

14. S.B. Kotsiantis, D. Kanellopoulos, P.E. Pintelas, Data preprocessing for supervised leaning. Int. J. Comput. Sci. **1**(2), 111–117 (2006)
15. I.H. Sarker, Machine learning: algorithms, real-world applications and research directions. SN Comput. Sci. **2**(3), 160 (2021)
16. S. Khalid, T. Khalil, S. Nasreen, A survey of feature selection and feature extraction techniques in machine learning, in *2014 Science and Information Conference*. IEEE (2014), pp. 372–378

Further Reading

17. V. Rajaraman, Big data analytics. Resonance **21**, 695–716 (2016)
18. D. Fisher, R. DeLine, M. Czerwinski, S. Drucker, Interactions with big data analytics. Interactions **19**(3), 50–59 (2012)

Cloud Computing for Big Data Analytics

4

Cloud computing is about how you do computing, not where you do computing.

—Paul Maritz

In this chapter, the exploration unfolds within the domain of cloud computing, emphasising its instrumental role in empowering the realm of big data analytics. Commencing with a comprehensive exposition, the historical evolution of cloud computing is meticulously traced across various computing generations, culminating in its contemporary manifestation as a transformative and indispensable component of the Information Technology (IT) landscape. Subsequently, cloud computing service models are systematically elucidated in conjunction with an exhaustive examination of deployment models, including public, private, hybrid, and community clouds. Furthermore, multi-cloud strategies are explored, with an in-depth exploration of key cloud computing platforms. A thorough comparison of these renowned cloud providers is offered to aid in making well-informed decisions and provide stakeholders with the necessary knowledge to effectively use cloud computing's promise to enhance big data analytics.

4.1 What is Cloud Computing?

Cloud computing is the combination of servers spread over the Internet and used to store, manage, and process data in bulk [1]. Cloud computing provides a large-scale computing environment by providing computing resources to Internet users [2,3]. The main features of cloud computing that make it increasingly popular are

Ü. Demirbaga et al., *Big Data Analytics*,
https://doi.org/10.1007/978-3-031-55639-5_4

excellent accessibility, improved collaboration, low maintenance cost, mobility, unlimited storage capacity, and data security. One of the things that makes cloud computing attractive is that users can only get resources from the cloud as they need and pay based on usage (pay as you go). Instead of temporarily purchasing the resources they need, users can use them by paying cloud providers for usage time. The flexible architecture of cloud computing enables storing large amounts of data and adding new resources easily based on the increasing amount of data and computing power requirements [4]. Cloud computing offers high-capacity servers to different users simultaneously using virtualization technology. In this way, it provides more efficient use of resources by providing a virtual machine instead of presenting a physical machine to the user. To this end, cloud providers perform big data analysis on system logs to improve the scalability and flexibility of their systems [5,6].

The cloud computing revolution has changed the use of computer infrastructure. Cloud computing platforms are widely used to deal with huge amounts of data called "Big Data." This term is a data volume exceeding a single system's storage and processing capacity. More specifically, big data is defined as more than hundreds of gigabytes. Big data is challenging to store and handle using traditional data processing systems due to the size and complexity of the data [7]. Cloud computing is leading in addressing the massive storage capacity problem and giving many flexible ways for managing massive amounts of data [8]. Parallel processing is key in handling complicated or large data, improving big data's performance and scalability. One of the most significant benefits of big data analysis is that it enables individuals to assess and spot dangers, allowing users to keep their data safe and secure. Google's MapReduce framework [9] and Apache Hadoop[1] are two extensively used software systems for large data applications [10].

4.2 The History of Cloud Computing

The history of cloud computing is unique, spans several decades, and has seen both substantial technological developments and the swiftly changing demands of the industry. Cloud computing has undergone several stages during its development, changing and strengthening to meet the ever-expanding needs of the sector. Let's now explore a complete review of the development of cloud computing to offer a thorough comprehension of this intriguing history.

- **Origins (1950s–1960s)**: The development of mainframe computers in the 1950s and 1960s laid the groundwork for cloud computing. Large-scale systems were used for data processing and storage during this period, and consumers accessed them through "dumb terminals." This centralised architecture made the idea of remote access to computer resources possible.

[1] https://hadoop.apache.org/.

- **Early Networking (1970s–1980s)**: Computer networks were developed in the 1970s and 1980s, allowing for resource sharing and communication across various systems. Distributed computing was made possible by technologies like Local Area Networks (LANs) and Wide Area Networks (WANs), which allowed users to access resources from far-off places.
- **Utility Computing (1990s)**: The 1990s witnessed the birth of "utility computing," which took its cue from offering computer resources as a utility, similar to electricity or water services. Providing processing power and software via the Internet on a pay-as-you-go basis was a pioneering effort by businesses like Amazon and Salesforce. The cloud computing paradigm was officially launched at this point.
- **Virtualization (Early 2000s)**: The development of Virtual Machines (VMs), which allowed different operating systems and applications to run on a single physical server, helped virtualization technologies gain popularity in the early 2000s. This innovation improved resource scalability, flexibility, and utilisation, providing the groundwork for cloud architecture.
- **Emergence of Cloud Providers (Mid-2000s)**: Major businesses began to enter the cloud computing sector in the mid-2000s. The Amazon Elastic Compute Cloud (EC2), which provides scalable computing resources over the Internet, was created by Amazon Web Services (AWS) in 2006. After that, Google introduced its Google Cloud Platform (GCP), and Microsoft released Azure, both offering cloud services, escalating the rivalry.
- **Cloud Service Models (Late 2000s)**: Different service models arose as cloud computing gained popularity to meet the diverse demands of users. Platform as a Service (PaaS) offered development platforms and tools, Software as a Service (SaaS) supplied software applications over the Internet, and Infrastructure as a Service (IaaS) allowed customers to rent virtualized infrastructure resources. These versions provided users with varying degrees of freedom and control.
- **Rapid Expansion and Adoption (2010s)**: Cloud computing had tremendous growth and industry acceptance in the 2010s. Organisations realised the benefits of the cloud's scalability, economy of scale, and agility. The cloud computing revolution was further accelerated by cloud providers expanding their products and adding new services, including serverless computing, containerization, and data analytics tools.
- **Hybrid and Multi-Cloud Environments (Present)**: Environments that use multiple clouds and hybrid clouds have received attention recently. Many businesses mix on-premises resources with private cloud infrastructure or public cloud services to exploit each environment's advantages. Multiple cloud providers are used in multi-cloud methods to minimise vendor lock-in and boost redundancy.
- **Advancements and Future Trends**: Cloud computing is still developing quickly. Cloud services are using recent developments in technologies like AI, machine learning, edge computing, and the IoT. Some subsequent developments are serverless computing, edge computing, and a stronger focus on security and privacy.

4.2.1 Computing Generations

Over the years, computing has made incredible strides, giving rise to unique computing generations that have influenced how we process, analyse, and manage data. The computer industry has undergone a revolution thanks to each generation's new paradigms and capabilities, which constitutes a tremendous technological advance. Each age of computing has offered its breakthroughs and possibilities, from the centralised power of mainframe computing to the decentralised control of personal computers and the collaborative nature of network computing to the worldwide connection of Internet computing. Additionally, grid computing made distributed resource management possible, and on-demand access to scale computing infrastructure was made possible by cloud computing, which changed the game.

To fully comprehend the context and relevance of cloud computing in big data analytics, one must thoroughly understand the evolution of computing generations. Figure 4.1 depicts the steps of generations of computing, where each has played a crucial role in shaping the evolution of big data analytics. Understanding their progression provides valuable context for utilising cloud computing in this field.

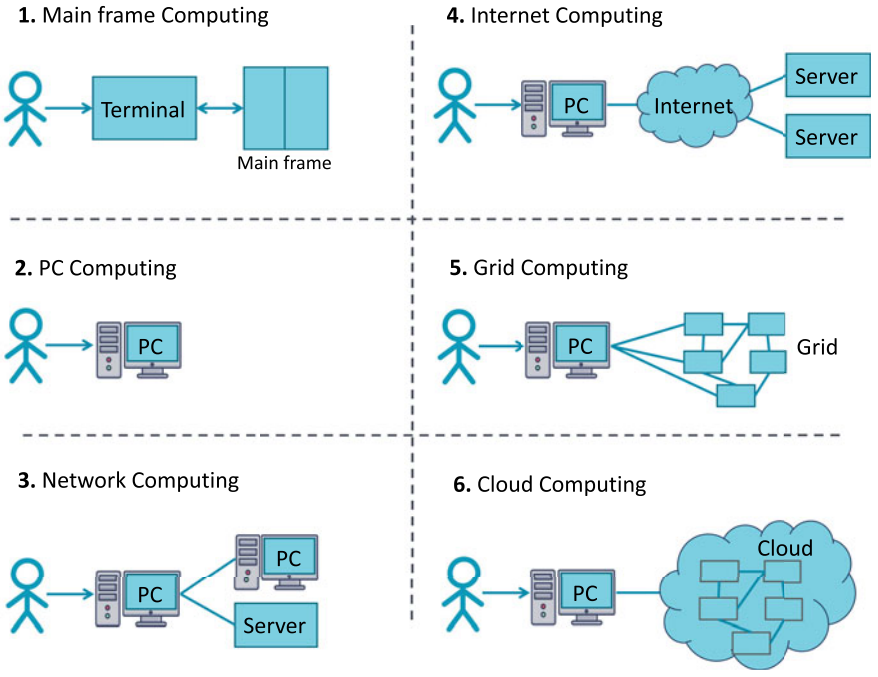


Fig. 4.1 Computing generations

4.2.1.1 Mainframe Computing

Mainframe computing is the term used to describe the early days of computing when mainframes, big, strong central computer systems, were employed to process and store enormous quantities of data [11]. Large organisations largely used these systems for crucial processes, including financial transactions, scientific research, and government operations, since they could perform complicated computations. The key characteristics of mainframe computing were its centralised architecture, tremendous processing power, and restricted accessibility, with users often connecting to the mainframe through terminals.

4.2.1.2 PC Computing

The introduction of Personal Computers (PCs) transformed the computing environment by giving everyone access to their computing capabilities [12]. Due to the decentralised nature of PC computing, individuals and small enterprises may now process and analyse data on their computers. This change in computer generation gave users more influence over their data and made it easier to create user-friendly software programmes, which helped boost individual productivity and analytical skills.

4.2.1.3 Network Computing

Network computing emerged as a paradigm with the development of networking technology, enabling the connection of several computers and sharing resources and collaborative computation [13]. This allowed users to access information and software on distant computers, promoting distributed computing settings. Network computing significantly improved data sharing, allowed businesses to utilise their pooled computer capacity and laid the foundation for upcoming advancements in the Internet and cloud computing.

4.2.1.4 Internet Computing

The expansion of the Internet brought about a significant change in computing. Internet computing uses networked computers to access and deliver services through the Internet [14]. The emergence of web-based apps, e-commerce platforms, and online services throughout this generation's lifetime allowed for smooth international data interchange, communication, and cooperation. By making it possible to gather, store, and retrieve enormous volumes of data from numerous sources, Internet computing significantly contributed to the development of data analytics.

4.2.1.5 Grid Computing

Grid computing is a type of distributed computing infrastructure that combines processing power from many sites to tackle challenging computational issues [15]. Across several administrative domains, it entails coordinating and sharing computing resources, storage, and applications. Grid computing focuses on on-demand

resource allocation and the ability to tap into extra processing power from other systems. Because technology allows extensive data analysis and simulations, grid computing has significantly impacted scientific and research groups.

4.2.1.6 Cloud Computing

The provisioning, delivery, and management of computer resources have all been revolutionised by cloud computing [16]. The Internet offers on-demand access to reconfigurable computing resources, including servers, storage, and applications. Cloud computing enables scalability, flexibility, and cost-effectiveness by allowing customers to pay for help based on utilisation. It has significantly aided big data analytics by offering the infrastructure, platforms, and tools required for processing huge amounts of data. The foundation of contemporary big data analytics is cloud computing, which enables businesses to use strong computing capabilities without substantial upfront expenditures in hardware and infrastructure.

4.3 Cloud Computing Units

Cloud computing units are essential to sorting and classifying cloud computing services. The service model and deployment model are two aspects that act as basic building blocks for the distinguishing characteristics, features, and workings of cloud-based solutions. Organisations can learn everything there is to know about the many subtleties and complexity of cloud computing by considering these components.

The service model dimension is an indispensable pillar that controls the quality of resources and services cloud providers offer consumers. This model clearly defines the range of obligations placed on users and the degree of control they have over the cloud infrastructure and apps. Users can decide the amount of engagement needed in controlling different areas of their cloud computing endeavours since it effectively defines the limits within which they function. Three main service models with other features and offerings arise within this dimension. Cloud computing units are shown in Fig. 4.2: *Service Model* and *Deployment Model*. The details are discussed in the next sections.

4.3.1 Cloud Computing Service Models

Service models for cloud computing cover a wide range of products and degrees of abstraction that cloud providers offer customers. These models provide a framework for identifying the level of accountability and control granted to users concerning the computer resources they utilise. In essence, these service models offer consumers the freedom to customise their cloud computing experiences to meet their needs. This paradigm is best shown by the four primary cloud computing service

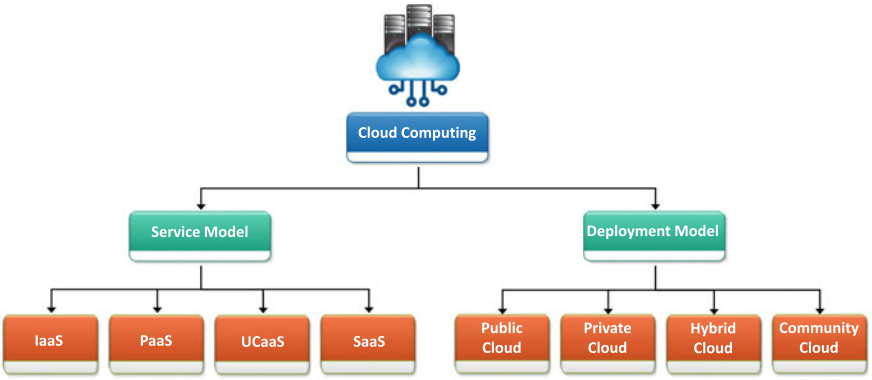


Fig. 4.2 Cloud computing units

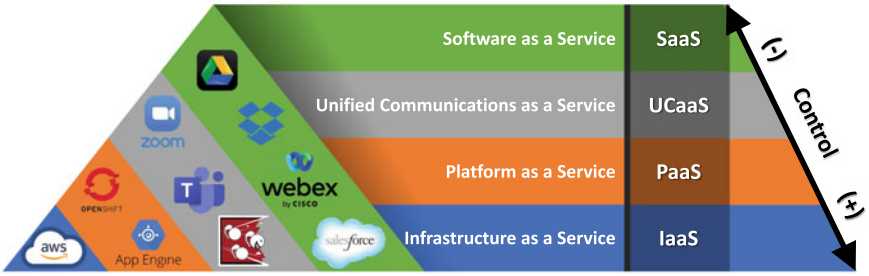


Fig. 4.3 Hierarchy of cloud computing service levels

models: IaaS, PaaS, Unified Communications as a Service (UCaaS), and SaaS [17]. Figure 4.3 depicts the types of cloud service offerings.

4.3.1.1 Infrastructure as a Service (IaaS)

This flexible cloud computing paradigm provides services such as virtual machines, storage drives, networks, IP addresses, and operating systems that create the infrastructure system [18]. Across the network, all these infrastructures are accessible via the cloud. Instead of purchasing the entire system, users only use the infrastructure needed and pay accordingly. Users here have access to a distributed cloud environment to run their applications. For instance, Amazon's EC2, Azure's Virtual Machine, and Google's Compute Engine are the services where the required hardware tools are available to meet users' requirements in the cloud.

4.3.1.2 Platform as a Service (PaaS)

In this service, a web application in the cloud is provided to the user within a platform by the cloud provider without the need to purchase. PaaS over the web provides necessary platforms such as an operating system, database management systems, programming language, and Web servers required for the user to create any software [19]. This provides a remote environment where customers can develop, build, and operate their software products. PaaS is a subtype of SaaS that uses the same infrastructure as IaaS. PaaS services include AWS Elastic Beanstalk, Windows Azure Heroku, Google App Engine, and OpenShift.

4.3.1.3 Unified Communications as a Service (UCaaS)

This service model provides communication continuity and remote collaboration services to users worldwide through the cloud network [20]. The remote workforce works seamlessly in a virtualized cloud environment with this service. This model, which has risen especially during the COVID-19 pandemic, has been adopted by various institutions and organisations to keep remote teams together. UCaaS helps the personnel working out of the office connect via phone/video calls and allows the workflow to continue quickly by allowing them to share files, documents, or resources quickly and securely. Zoom, Microsoft Teams, Fuze, Google Meet, and Jive are examples of UCaaS.

4.3.1.4 Software as a Service (SaaS)

SaaS, an "on-demand software application," refers to a complete product managed and maintained by a cloud service provider. SaaS is a cloud-based service that extends a whole software suite in a pay-per-use model [21]. It is made available to end-users via the Internet. SaaS is a superset of both PaaS and IaaS since it includes everything from infrastructure, middleware, and operating systems to web-based applications that can be accessible at any time, from any location, and on any platform. This fully built cloud service paradigm helps enterprises get started amid the COVID-19 epidemic, adapt quickly, and grow company operations remotely. Some examples of SaaS are Google Docs, Google Apps, Hotmail, Online Payroll, Dropbox, Hubspot, Salesforce, Slack, and DocuSign.

Figure 4.4 demonstrates the management of the resources in cloud computing service models. The key difference between on-premise and cloud is where the software is hosted. The software is hosted on the vendor's server in the cloud system and accessed via a web browser. In contrast, on-premise software is hosted locally and on the company's server and managed by themselves or a third party.

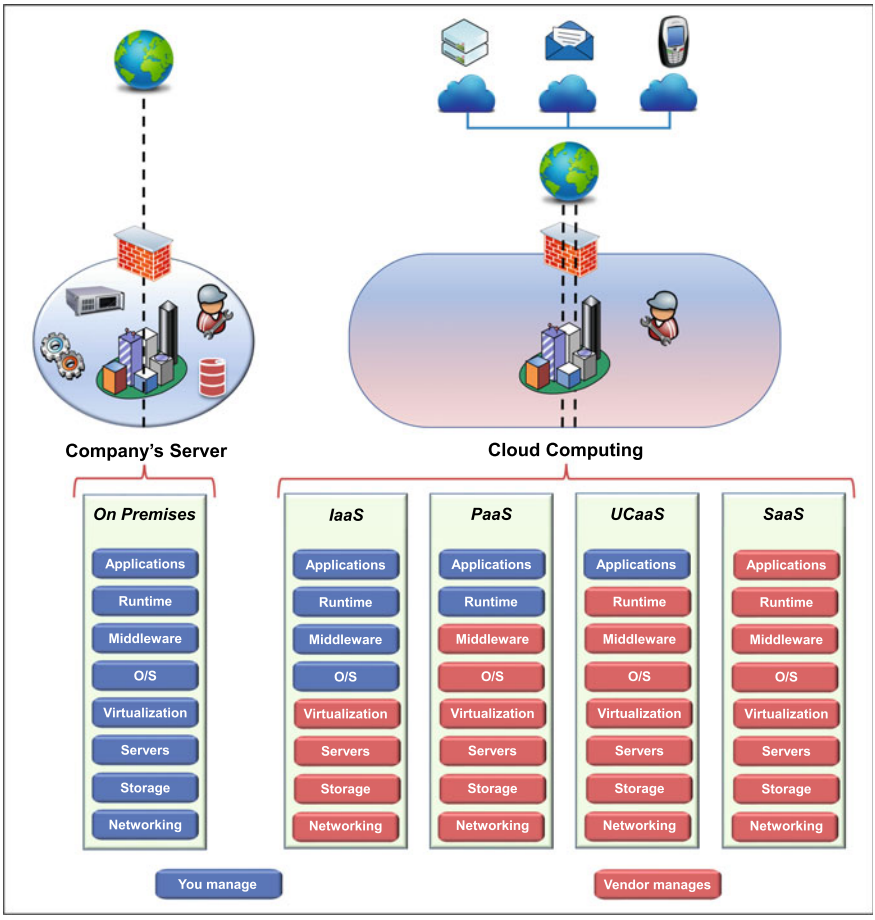


Fig. 4.4 Cloud services control comparison

4.3.2 Cloud Computing Deployment Models

Each user or organisation has different needs, such as how much control they want over the environment and the services they will use based on their needs. Different service models are offered to users within the cloud system to meet these requirements. These different service models have different purposes and requirements. Four different deployment models for cloud services explain the nature and purpose of cloud applications. Figure 4.5 shows cloud computing deployment models.

4.3.2.1 Public Cloud

In this model, where the cloud service is publically available to the user, businesses can quickly access computing resources without an upfront cost [22]. Cloud services are offered directly to the user without any third-party involvement. Companies

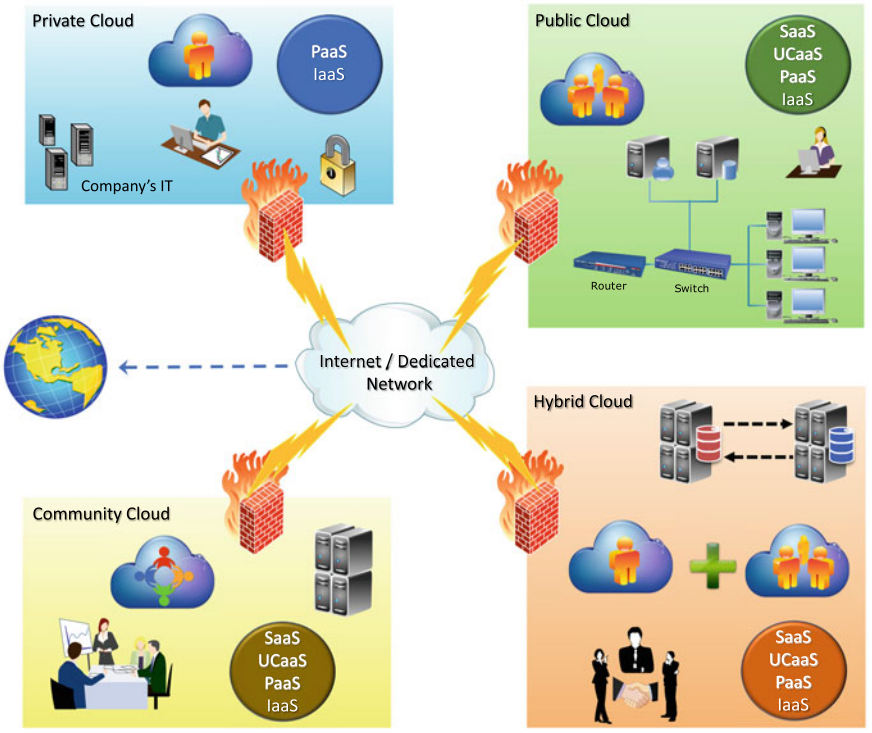


Fig. 4.5 Cloud computing deployment models

purchase the computing, storage, and network services they need directly over the public Internet from a cloud provider in a virtualized manner. With this model, businesses get quick access to resources and can bring their products to market faster and try new services. Companies do not need their IT team to maintain hardware and deal with hardware-related issues as the necessary infrastructure is provided by this model. Some examples of public cloud facilities are AWS, Microsoft Azure, Alibaba Cloud, Google Cloud, and IBM.

Benefits: The public cloud is a desirable business alternative due to its many benefits. First, public cloud services are extremely scalable, enabling companies to quickly scale up or down their resources per their demands, assuring optimum resource utilisation and cost-efficiency. Additionally, the public cloud eliminates the need for upfront capital investment in infrastructure as users only pay for their use on a pay-as-you-go basis, resulting in considerable cost savings. Moreover, a variety of services and features are provided by public cloud providers, allowing businesses to use cutting-edge technology without worrying about maintaining the underlying infrastructure. Furthermore, the public cloud providers' stringent security protocols and compliance certifications ensure the confidentiality and privacy of data. Public cloud services' extensive geographic reach ensures high availability and

dependability, making data available at all times and from any location. Last but not least, the provider regularly updates public cloud services, freeing organisations of the duty of infrastructure maintenance and enabling them to concentrate on their primary business goals.

Drawbacks: This model comes with limitations. First, companies can face heavy bills, especially midsize to large ones, due to a lack of cost control in public clouds. Secondly, the public cloud is inherently the least secure, so it's not a good choice for sensitive, personal, and mission-critical workloads. Lastly, companies have low visibility and technical control over the cloud system, which prevents fulfilling compliance requirements.

4.3.2.2 Private Cloud

Companies construct their cloud-native structure via a private cloud system. All layers of the private cloud are created and managed by the company. The resources that build the system infrastructure, such as processing power, storage, and networking, are located within the company and are usually accessed over the local LAN or WAN [23]. A Virtual Private Network (VPN) provides external access to system resources.

Benefits: Private cloud is a popular option for businesses as it provides many benefits. First and foremost, private clouds boost security and data privacy since they are only used by one organisation, lowering the possibility of unauthorised access or data breaches. This is especially useful for businesses that handle sensitive data or must adhere to strict legal regulations. Moreover, private clouds give companies more flexibility and control to adjust the infrastructure and services to suit their requirements and preferences. This helps organisations fully manage their data, applications, and infrastructure while enabling seamless connection with already installed programmes and systems. Furthermore, as resources are not shared with other organisations, the private cloud offers predictable performance and dependability, providing reliable and constant performance, which offers more deployment flexibility because private clouds can be hosted on-premises or in a dedicated data centre, allowing businesses to select the setting that best meets their needs. A private cloud is an excellent option for companies with particular compliance or security demands since it gives organisations more control, security, and flexibility overall.

Drawbacks: The private cloud has certain limitations that may restrict its applications. First, it is a rather expensive solution than the public cloud, especially for companies needing short-term use. In addition, due to the high-security measures implemented in the private cloud system, access to the system is quite tricky. Finally, the limited resources of private cloud infrastructure cannot offer the scalability to meet the unpredictable data size and the required processing power needs.

4.3.2.3 Hybrid Cloud

This model combines private and public cloud models that emerge when public cloud resources are used when private cloud services are insufficient [24]. Similarly, critical

services such as corporate data processing are accessed from the private cloud while using the public cloud. It overcomes the limitations of the other two models and offers the advantages of all models.

Benefits: Hybrid cloud provides unrivalled flexibility by fusing the advantages of both public and private clouds. Organisations can use the public cloud's scalability and affordability for non-sensitive or cyclical workloads while retaining sensitive data and crucial applications inside the private cloud's protected boundaries. Due to this flexibility, businesses can optimise resource allocation, cut expenses, and adjust to shifting needs. A hybrid cloud also allows data and applications to move easily between environments, supporting smooth workload transfer and avoiding vendor lock-in. As businesses can create a hybrid cloud architecture that closely matches their unique needs, it provides more control and customization. By offering redundant and geographically dispersed infrastructure, hybrid clouds also improve disaster recovery capabilities while maintaining business continuity and data resilience. Overall, the benefits of hybrid clouds lie in their capacity to find a balance between control, security, scalability, and cost-effectiveness, enabling businesses to benefit from the best of both worlds.

Drawbacks: It is challenging to achieve harmony and integration between private model and public cloud infrastructure spread over different locations and categories. In consequence, additional infrastructure and software are required to overcome these difficulties. Switching between private and public clouds is difficult to monitor, resulting in wasted and increased costs.

4.3.2.4 Community Cloud

This model is a collaborative effort maintained by the community that built it, where several organisations with the same computing concern come together to share their infrastructure [25]. A modernised version of the private cloud, this model addresses security concerns while reducing companies' implementation costs.

Benefits: Community cloud is a good option for groups or organisations with similar objectives or interests since it has some special advantages. First, it encourages cooperation and information exchange among community members, allowing them to cooperate on achieving shared goals or overcoming shared difficulties. The community cloud infrastructure is created to address the community's needs, offering specialised services, programmes, and valuable tools to every member. Community cloud also frequently provides improved security protocols and legal compliance that are tailored to the particular company or sector the community belongs to. The community cloud is a financially attractive choice since it provides cost reductions for all users by sharing infrastructure expenses and pooling resources. Furthermore, compared to the public cloud, the community cloud enables more customization and control since the community may jointly impact the features and functions of the cloud environment through a shared governance architecture. Overall, the benefits of the community cloud are found in its capacity to encourage community member cost savings, boost community member cooperation, offer personalised solutions, and offer increased security and compliance.

Drawbacks: Companies must arrange additional Service Level Agreements (SLAs) when sharing resources and data in the community cloud model, although it provides better security than the public cloud. It is also more costly than the public cloud as it requires on-premises cloud experts. Moreover, similar to the private cloud, scalability is insufficient due to limited storage, processing power, and bandwidth.

4.4 Multi-Cloud Strategies in Big Data Analytics

As discussed above, while cloud computing offers different models, eliminating the supply needs of small and medium-sized enterprises and offering options such as scaling for their growth, cloud systems have some issues and challenges. These problems create significant problems in terms of time and budget, especially in long-term cloud use. **Security** is at the forefront of these problems with various subheadings, including security mechanisms, cloud monitoring, data confidentiality, and avoidance of malicious transactions. **Legal concern** about data storage is also one of the problems. User data may be in another country in the cloud system, as users prefer cheap and available servers. Although service level agreements have been made between cloud providers and their customers, there is no standard worldwide. Therefore, data storage rules and laws vary from country to country. Users must follow and obey different laws and be more careful when designing their systems. Data management also differs from one to another one. **Interoperability**, which indicates the ability of various systems and organisations to work together, is another prominent challenge for cloud computing systems. Another problem is the **high latency**, known as the time between a user request and a response from a cloud service provider. This problem is usually caused by problems with the components that make up the cloud system communicating with each other. Lastly, in cloud computing, the **vendor lock-in** problem occurs when clients rely on a single cloud service provider's technological implementation and cannot simply switch vendors without incurring significant expenses or technical incompatibilities.

The definition of multi-cloud started to come to the forefront towards the end of 2018 with technology development. Multi-Cloud is a cloud computing paradigm in which two or more types of cloud (including private cloud) come together and form a combination, considering a user's or organisation's specific needs. Figure 4.6 shows a use case diagram for a multi-cloud management system using Aviatrix,[2] a cloud networking platform that provides a common solution for managing multi-cloud computing platforms.

Given all these problems, the Multi-cloud strategy offers organisations the following [26]:

[2] https://aviatrix.com/.

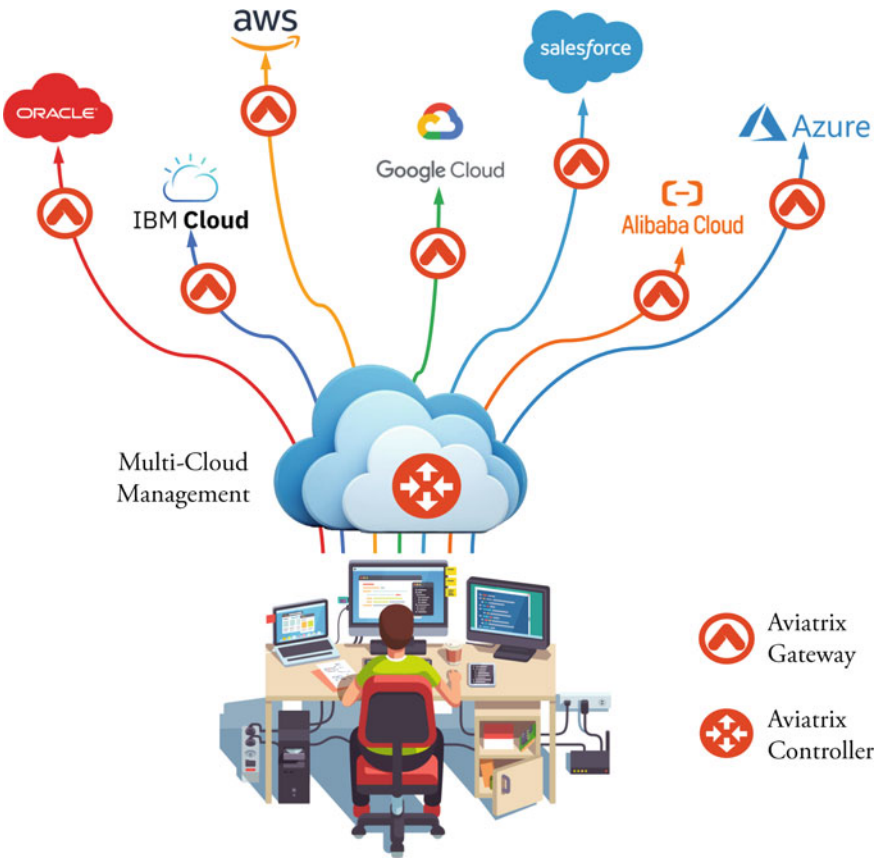


Fig. 4.6 A use case diagram for a simplified multi-cloud management system

- Multi-cloud provides the user with the most convenient choice of geolocation while also giving the flexibility to choose cloud services from different cloud providers based on pricing, performance, security, and compliance requirements.
- Multi-cloud usage offers the convenience of rapidly adopting "best-of-breed" technologies from any vendor as needed or as they arise, rather than limiting customers to any offering or functionality from a single vendor at a given time.
- Multi-cloud offers reduced vulnerability to unplanned downtime because an outage in one cloud will not necessarily affect services in other clouds.

4.5 Cloud Computing Platforms for Big Data Analytics

Cloud computing platforms enable the distribution of computing resources, storage, and applications via the Internet by offering a wide range of services, tools, and resources. These platforms provide a scalable and adaptable infrastructure that enables users to access and use computing power, storage, databases, and software whenever they need it without buying more infrastructure or physical gear. Cloud computing platforms allow organisations to focus on their primary business goals rather than managing and maintaining complicated IT infrastructure. These platforms also enable effective resource allocation, seamless scalability, and cost optimisation. By enabling agility, accessibility, and the capacity to install and expand apps and services quickly, they have completely changed how organisations operate. Organisations leverage cloud computing platforms to use pooled computing resources, data storage, and cutting-edge technology to spur innovation, boost productivity, and quicken the digital transition. The most commonly used cloud providers are shown in Fig. 4.7.

Cloud computing providers, such as AWS, Google Cloud, Microsoft Azure, and Alibaba Cloud, are robust and all-encompassing systems that provide a wide range of services and resources for consumers and enterprises. Users may instantly access processing power, storage, databases, and various software solutions because of these platforms' scalable and adaptable architecture. With AWS, Google Cloud, Azure, or Alibaba Cloud, businesses can take advantage of the cost-effectiveness, agility, and scalability of cloud computing to meet their unique demands and spur innovation. These platforms let organisations accelerate their digital transformation by utilising cutting-edge technology and services while concentrating on their core capabilities.

Now, we will focus on three favourite cloud computing services in detail:

Fig. 4.7 Top Cloud service providers

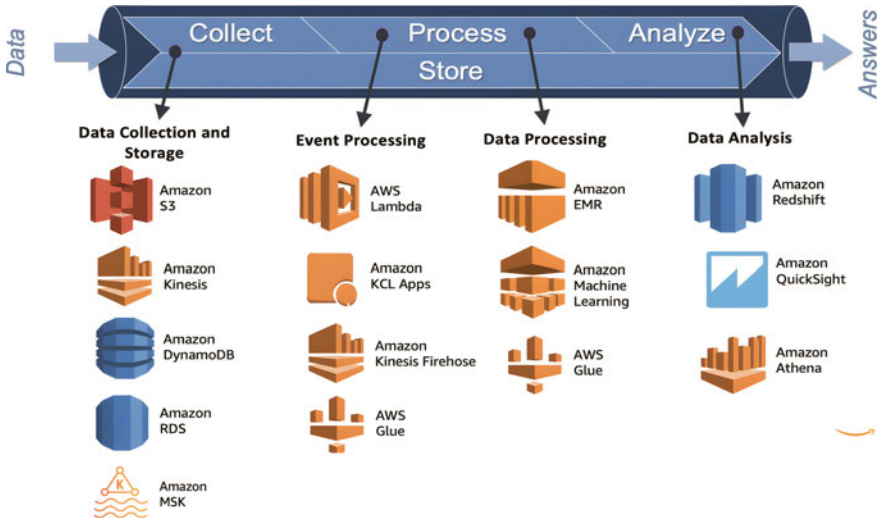


Fig. 4.8 AWS data processing pipeline [27]

4.5.1 Amazon Web Services (AWS)

Amazon.com introduced AWS[3] in 2006 in response to the demand for scalable and affordable cloud computing services. It began by providing essential infrastructure services, such as processing power and storage, but soon broadened its portfolio to encompass a variety of services, including databases, analytics, machine learning, and IoT. By offering on-demand, pay-as-you-go cloud services and removing the need for significant upfront expenditures in hardware and infrastructure, AWS upended the established Information Technology sector. AWS has developed and innovated, becoming the world's top cloud computing provider. Millions of customers, including startups, large businesses, and governmental institutions, rely on AWS to fuel their digital transformation programmes and foster creativity at scale (Fig. 4.8).

AWS is a well known and complete platform in cloud computing for big data analytics. Big data analytics can be implemented using a wide range of AWS services and technologies that are scalable, effective, and affordable. AWS's essential features and services as a cloud computing platform for big data analytics are explained below.

4.5.1.1 Amazon Elastic Compute Cloud (EC2)

AWS provides the highly scalable and adaptable Amazon EC2[4] as a cloud computing solution. Users can provision virtual servers, sometimes called instances, in the cloud using EC2, giving them total control over their computing resources. Users

[3] https://aws.amazon.com/.
[4] https://aws.amazon.com/ec2/.

of EC2 have access to a large selection of instance types that have been optimised for various workloads, including high-performance computing, memory-intensive applications, and storage-oriented jobs. Using EC2 instances, customers may scale their computing capacity up or down in response to demand since they are simple to configure, deploy, and terminate. Furthermore, EC2 offers load balancing, auto-scaling, and virtual networking capabilities that let customers create and maintain sophisticated, highly available, and resilient applications.

4.5.1.2 Amazon Simple Storage Service (S3)

The highly scalable and reliable object storage solution Amazon Simple Storage Solution (S3)[5] is offered by AWS. Businesses and developers can store and retrieve massive volumes of data securely and affordably using S3. With S3, customers can keep and retrieve any volume of data from any online location with great durability and availability. Users may construct buckets (containers) and upload objects (files) inside of those buckets using the user-friendly and straightforward S3 interface. S3 provides several storage classes, including Standard, Intelligent-Tiering, Glacier, and others, to reduce storage costs depending on data access patterns. Additionally, it effortlessly connects with other AWS services, allowing customers to take advantage of its scalability and dependability to run their apps, backup data, and more.

4.5.1.3 Amazon Elastic MapReduce (EMR)

Amazon Elastic MapReduce (EMR)[6] is a cloud-based big data processing service provided by AWS. EMR uses Apache Hadoop and other open-source frameworks to streamline the processing and analysis of massive datasets. To run data-intensive tasks like data transformations, machine learning, and data warehousing, users may deploy and expand a cluster of virtual servers using EMR. EMR offers a managed environment that automatically takes care of the infrastructure's installation, configuration, and tuning, freeing users to concentrate on data analysis rather than infrastructure administration. It effortlessly interfaces with other AWS services to enable data intake from numerous sources, storage in Amazon S3, and connection with Amazon Redshift for data warehousing. EMR offers a scalable, affordable solution for processing large data, enabling businesses to get insightful knowledge and make informed decisions.

4.5.1.4 Amazon Athena

Amazon Athena[7] is an interactive query tool provided by AWS that enables customers to analyse data directly from Amazon S3 using conventional SQL queries.

[5] https://aws.amazon.com/s3/.
[6] https://aws.amazon.com/emr/.
[7] https://aws.amazon.com/athena/.

With Athena, users can quickly extract insights from massive datasets and do ad-hoc analysis without needing complicated infrastructure setup or data processing. Because Athena is serverless, there are no upfront charges or capacity planning necessities; customers pay for the queries they execute. It supports various file formats and compression methods, enabling users to query data saved in multiple forms effectively. Athena is a useful tool for businesses wishing to use their data for research and decision-making since it allows for fast and economical querying of S3 data.

4.5.1.5 Amazon Redshift

AWS offers a cloud-based data warehousing solution called Amazon Redshift.[8] It is completely managed. Redshift provides great speed, scalability, and affordability while being intended to handle large-scale data analytics tasks. It uses parallel query execution and columnar storage to give quick query performance on huge datasets. Redshift interacts effortlessly with well-known BI tools and data integration providers, enabling users to analyse their data quickly using conventional SQL queries. Redshift offers automated backups, data replication, and encryption to guarantee data longevity and security. It lets businesses effectively store and analyse enormous volumes of data, enabling sophisticated analytics and data-driven decision-making at scale.

4.5.1.6 Amazon Glue

Amazon Glue[9] is a fully managed Extract, Transform, and Load (ETL) service offered by AWS. By automating time-consuming operations like schema discovery, data transformation, and job scheduling, Glue simplifies preparing and loading data for analysis. It supports a wide range of data sources and formats, making it simple to combine and transform data from many sources into a single form. The serverless design of Glue expands automatically to accommodate huge datasets and offers a visual interface for generating, managing, and tracking ETL operations. Users may utilise Glue to improve their data preparation workflows, speed up data integration procedures, and prepare their data for usage in downstream analytics and machine learning applications.

4.5.1.7 Amazon Kinesis

Amazon Kinesis,[10] a fully managed real-time streaming data platform, enables businesses to gather, process, and analyse streaming data in real time, providing instant insights and prompt action. It facilitates the input of significant amounts of data from several sources, such as clickstreams, IoT devices, logs, and social media feeds.

[8] https://aws.amazon.com/redshift/.
[9] https://aws.amazon.com/glue/.
[10] https://aws.amazon.com/kinesis/.

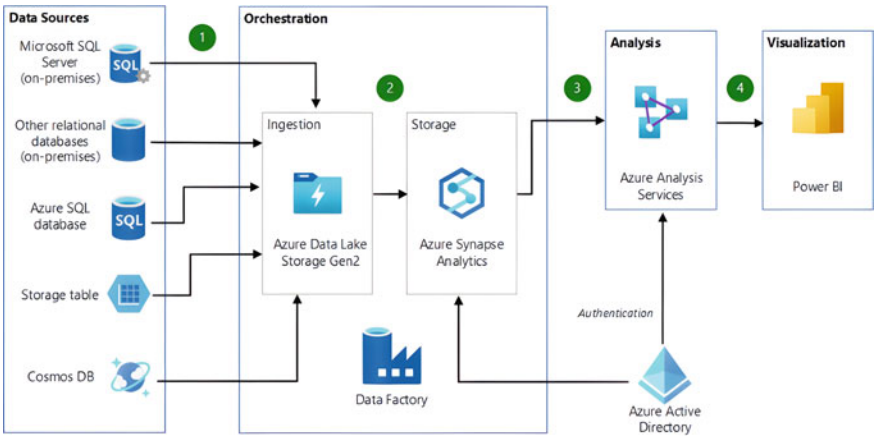


Fig. 4.9 Microsoft Azure data processing pipeline [28]

Kinesis offers data streaming, transformation, and storage features, enabling customers to create real-time analytics and machine learning applications. Kinesis equips organisations to unleash the value of their streaming data and make educated, data-driven choices in real-time thanks to its scalability, dependability, and simplicity of use.

4.5.2 Microsoft Azure

Microsoft Azure,[11] launched in 2010, has expanded quickly to rank among the top cloud computing platforms. Due to Microsoft's strategic move towards cloud computing and its goal to offer a full range of cloud services, it first went by the name Windows Azure. The history of Azure has been marked by constant innovation and growth, and the platform has developed to provide a variety of infrastructure, platform, and software services. Azure has grown significantly in terms of its global data centre footprint, introduced cutting-edge technologies like AI and machine learning, and established itself as a trustworthy and dependable platform for businesses worldwide to develop and deploy their cloud-based applications and services (Fig. 4.9).

Azure provides users with a reliable and scalable infrastructure that enables the on-demand provisioning of virtual machines, storage, and networking resources. Azure's wide-ranging global footprint enables businesses to distribute their applications and services across several areas globally, assuring high availability and minimal latency. Microsoft Azure comprises various crucial parts enabling businesses to utilise cloud computing fully. The following are a few of Azure's essential elements:

[11] https://azure.microsoft.com/.

4.5.2.1 Azure Virtual Machines

Microsoft Azure, a cloud computing platform offered by Microsoft, includes Azure Virtual Machines as an effective service. With the help of Azure VMs, users may set up and administer virtual machines in the Azure cloud environment, allowing them to construct and control cloud-based instances of different operating systems (like Windows or Linux). Azure's scalability and computational capacity enable users to handle and analyse massive data.

The following are some crucial characteristics of Azure Virtual Machines in the context of big data analytics.

1. **Deployment**: Deployment in Azure Virtual Machines is the procedure for quickly creating and setting up virtual instances of operating systems in the Azure cloud environment. It provides versatility by giving users many predefined virtual machine pictures or the choice to import their images. Azure simplifies deployment by offering customers simple-to-use interfaces and APIs that let them select virtual machine sizes, regions, and networking options. At the same time, Azure handles the configuration of the underlying infrastructure. By automating and maintaining consistency, deployment templates significantly simplify the process. Once implemented, virtual machines may be held, watched over, and accessed remotely, allowing users to install software, set up networks, and create security protocols.
2. **Scalability**: Scalability in Azure Virtual Machines allows scaling up or down the assigned Central Processing Unit (CPU), memory, and storage according to the needs of the applications or big data analytics workloads. Users can select the optimum level of resources to optimise performance and cost-efficiency by using Azure Virtual Machines, which provide a variety of VM sizes and configurations. Users can scale manually by modifying the resources themselves as needed, or they can scale automatically using Azure Autoscale, which keeps track of workload metrics and adjusts the resources accordingly. For effectively handling massive information and performing demanding tasks, Azure Virtual Machines must have the capacity to expand their resources. It ensures that the infrastructure can change on the fly to meet the workload's needs, delivering the best performance possible while preventing resource shortages or underutilization.
3. **Performance**: Azure virtual machines aim to achieve the best possible computational efficiency and responsiveness when running cloud applications. There are several choices available with Azure Virtual Machines to enhance performance. Users can choose from various VM sizes with CPU, memory, and storage configurations to accommodate diverse task demands. Azure also provides high-performance storage choices, including Premium Solid State Drives (SSD), Ultra Disc, and Azure NetApp Files, which can greatly speed up data processing and retrieval. Azure Virtual Machines effortlessly connect with other Azure services like Azure Load Balancer or Azure Cache for Redis to further boost speed. Users can track performance indicators and make educated decisions for optimisation using monitoring tools and analytics.
4. **Connectivity**: Azure Virtual Machines provide strong networking choices facilitating efficient data transmission, teamwork, and integration. Customers can

quickly link their virtual machines to Azure Storage, Azure Data Lake Storage, Azure Databricks, Azure HDInsight, or Azure Synapse Analytics, which enables effective big data processing, storing, and analysis. In addition, virtual networks can be connected to Azure Virtual Machines, providing secure communication within the Azure architecture or extending the connection to on-premises networks using Azure VPN Gateway or Azure ExpressRoute. Azure Load Balancer and Azure Application Gateway allow traffic distribution and load balancing across several virtual machines. Azure Virtual Network Service Endpoints also provide private and secure access to Azure services without the need for public IP addresses.

5. **Management and Monitoring**: Users can efficiently manage and monitor their virtual machine instances in the Azure cloud environment by using a full collection of tools and services included with Azure Virtual Machines. Users can do operations like provisioning, configuring, and operating virtual machines using a user-friendly user interface and reliable APIs. Azure offers a variety of administration features, such as network management, automatic scaling rules, backup and disaster recovery solutions, and security setups. Additionally, Azure provides several monitoring tools that let customers track performance indicators, identify problems, and maximise resource use. While Azure Advisor makes proactive suggestions for enhancing the setup and performance of virtual machines, Azure Monitor delivers insights into the performance, logs, and diagnostics of virtual machines.

4.5.2.2 Azure App Service

Microsoft Azure's comprehensive PaaS solution, Azure App Service, makes it simple for developers to create, launch, and grow online and mobile apps. From a research and development standpoint, Azure App Service provides a stable and adaptable environment so researchers and developers can concentrate on creating applications without worrying about maintaining the underlying infrastructure.

Azure App Service supports several programming languages and frameworks, including .NET, Java, Python, and Node.js. This is one of its primary benefits. Because of their flexibility, researchers may use the languages and tools that they find most useful, which makes it easier to design applications in settings they are already comfortable with. Moreover, Azure App Service offers easy connections with other Azure services and tools, enabling researchers to use Azure's ecosystem. For example, connection with Azure SQL Database provides effective data archiving and retrieval, while interaction with Azure Functions empowers the creation of serverless architectures for increased scalability.

Applications can manage various workloads with the automated scaling features of Azure App Service based on customisable performance criteria. This scalability is crucial in academic settings where apps could face abrupt spikes in traffic brought on by increasing user engagement or research needs. Furthermore, Azure App Service offers powerful monitoring and diagnostics capabilities that allow researchers to learn important details about an application's functionality and usage

trends. This information helps enhance user experience, pinpoint bottlenecks, and optimise application performance. In addition, Azure App Service provides integrated authentication and authorisation protocols, making it simple for researchers to safeguard their apps. It also connects with Azure Active Directory for identity and access control to ensure that authorised users can safely access applications.

4.5.2.3 Azure Storage

Azure Storage, a cloud-based storage option provided by Microsoft Azure, offers highly scalable and reliable storage services to fulfil the needs of modern data-intensive applications. It works as a trustworthy and affordable platform for managing and storing enormous amounts of data in many types, from structured to unstructured.

Blob storage, File storage, Queue storage, and Table storage are just a few of the essential services Azure Storage provides; each offers specific functionality and caters to particular storage needs. Large volumes of unstructured data, such as pictures, movies, and documents, can be kept in blob storage. Collaboration and data sharing are made possible by file storage, which enables the establishment of shared file systems that several virtual computers may access. A dependable messaging solution for asynchronous communication between various application components is provided by queue storage. A NoSQL key-value store with table storage can scale up to accommodate enormous volumes of structured data.

The scalability of Azure Storage is one of its key benefits, which eliminates the need for upfront capacity planning by enabling organisations to effortlessly scale their storage resources up or down in response to their demands. Users may store and retrieve terabytes or petabytes of data using Azure Storage, ensuring their applications have the storage space needed to keep up with data growth. Durability and reliability are other important features of Azure Storage, where data stored in Azure Storage is automatically duplicated across different data centres within a region to provide high availability and data resilience. Data durability is ensured even during a disaster or system failure through this replication mechanism's protection against hardware failures.

Additionally, Azure Storage focuses on security, where stringent security measures are deployed to protect data at rest and in transit. Azure Storage provides encryption at rest, where data is automatically encrypted before being saved and can only be decrypted by authorised users. Additionally, Azure Storage interfaces with Azure Active Directory for strong authentication and access control, enabling businesses to manage user identities and permissions efficiently.

4.5.2.4 Azure Cosmos DB

Microsoft Azure offers a globally distributed, multi-model database solution Azure Cosmos DB which provides a highly scalable, low latency, and universally accessible data storage solution created to fulfil contemporary applications' rigorous needs. Built on the idea of NoSQL databases, Azure Cosmos DB offers support for several different data formats, including document, key-value, graph, and columnar.

This multi-model method enables flexibility and adaptability in managing a variety of data kinds and structures by allowing developers to select the most appropriate data model for their application. Azure Cosmos DB provides low-latency access to data, allowing real-time and interactive applications. It combines several approaches for optimum efficiency and effective resource use, including automated indexing, automatic sharding, and automatic load balancing.

Organisations can grow their databases horizontally and vertically using Azure Cosmos DB, which provides a wide range of scalability options. Data is distributed across numerous nodes by partitioning, allowing horizontal scaling to manage growing workloads and storage needs. Vertical scaling entails boosting individual nodes' computational and storage capabilities for bigger data collections or faster throughput. Support for several APIs, including SQL, MongoDB, Cassandra, Gremlin, and Table, is another significant feature of Azure Cosmos DB. As a result, it is simpler to integrate Azure Cosmos DB with existing applications and use developer expertise and familiar programming models and query languages.

Security is of the utmost importance in Azure Cosmos DB, which has strong security features, including encryption both in transit and at rest, role-based access control, and interaction with Azure Active Directory for authentication and identity management. This guarantees that data stored in Azure Cosmos DB is safe from unauthorised access and complies with all applicable data security and compliance regulations. Additionally, Azure Cosmos DB offers a wide range of monitoring and administration tools, including metrics, logs, and diagnostics, to provide administrators with a better understanding of the functionality and state of their databases. Additionally, because of its easy integration with other Azure services, businesses can create comprehensive solutions that include data processing, analytics, and storage.

4.5.2.5 Azure AI Services

Azure AI services provide full AI capabilities, including pre-built AI models, machine learning tools, chatbot creation, and infrastructure support. These services offer natural language processing, computer vision, speech recognition, and machine learning, empowering businesses to integrate AI into their apps and workflows. With Azure AI services, companies can utilise AI to boost innovation, improve user experiences, and extract insightful information from their data.

Some Azure AI services are detailed below, enabling organisations to leverage the power of AI in their applications and workflows by providing a comprehensive suite of AI capabilities, from pre-built models to custom solutions.

1. **Azure Cognitive Services**: Azure Cognitive Services are the pre-built AI models and APIs allowing programmers to quickly add AI functionality to their applications. These services offer text analytics, computer vision, speech recognition, and language comprehension. Without creating intricate AI models, developers can use Azure Cognitive Services to perform tasks like sentiment analysis, picture identification, and natural language processing.

2. **Azure Machine Learning**: Azure Machine Learning, a fully managed cloud service, allows data scientists and developers to create, train, and deploy machine learning models at scale. It offers a variety of tools and capabilities, such as automated machine learning, hyperparameter tweaking, and model deployment, and it supports well-known machine learning frameworks. Making it simpler to design and implement machine learning solutions, Azure Machine Learning simplifies the whole end-to-end machine learning process, from data preparation and model training through model assessment and deployment.
3. **Azure Bot Services**: Developers can build smart chatbots and virtual agents with Azure Bot Services, which uses natural language processing to communicate with users. These bots may be linked with various chat applications, including Facebook Messenger, Slack, and Microsoft Teams, allowing users to access them over many channels. Building conversational agents that can help with customer care, deliver information, and automate processes is made possible for businesses by Azure Bot Services, which offers a full development environment with tools for bot creation, testing, and deployment.
4. **Azure Speech Services**: Azure Speech Services offer speech recognition and synthesis capabilities for developers to add speech-related features to their apps. These services enable applications to process and produce speech-based information by converting spoken language into written text (speech-to-text) and vice versa (text-to-speech). Multiple language support, customisable voice recognition models, speaker identification, and real-time transcription make Azure Speech Services ideal for various speech-related applications.
5. **Azure Computer Vision**: Azure Computer Vision offers sophisticated image analysis features, including Optical Character Recognition (OCR), object identification, and picture recognition. Developers may use these services to detect people or objects in photographs, extract useful information, and transform printed or handwritten language into machine-readable text. With the help of Azure Computer Vision, organisations can extract important insights from visual data in various situations, including content moderation, image-based search, and document digitalization.
6. **Azure Custom Vision**: Azure Custom Vision's feature makes building and altering image recognition models possible to suit certain business requirements. Building incredibly precise and domain-specific image identification systems is made feasible by the ability of developers to train machine learning models using their own labelled picture datasets. In addition to offering tools for model training, assessment, and deployment, Azure Custom Vision includes categorization and object detection activities. When pre-trained models already in use do not completely meet the specific needs of an application, this service is very helpful.

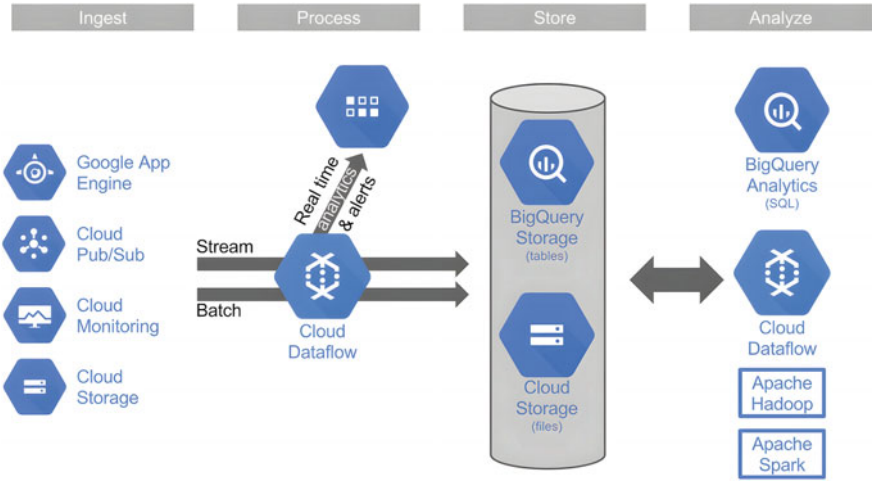


Fig. 4.10 Google cloud data processing pipeline [29]

4.5.3 Google Cloud Platform (GCP)

GCP[12] is a powerful and comprehensive set of cloud computing services. It gives businesses a strong, adaptable infrastructure to create, launch, and expand their apps and services. A wide range of services, including computing, storage, networking, data analytics, machine learning, and more, are available through GCP. GCP is a great option for companies of all sizes because it guarantees high availability, low latency, and scalability through its worldwide network of data centres.

With services built to handle and analyse massive volumes of data, data analytics is one of GCP's major strengths. Through SQL-like searches on enormous datasets, BigQuery, a fully managed data warehouse, provides quick and scalable analytics. The creation of data pipelines is made easier by the real-time stream and batch processing of data provided by Cloud Dataflow. Cloud Dataproc offers managed Apache Spark and Hadoop clusters for large data processing. The services provided by GCP for AI and machine learning, like AutoML, AI Platform, and TensorFlow, enable businesses to build and deploy machine learning models, making it simpler to derive knowledge and intelligence from their data. Figure 4.10 demonstrates the data processing pipeline of GCP.

The essential components of the Google Cloud Platform are detailed below.

4.5.3.1 Compute Engine

Compute Engine provides customers access to VMs to run their applications flexibly and be scalable. It functions as an IaaS offering, allowing customers complete control

[12] https://cloud.google.com/.

over the underlying resources to build, configure, and manage VM instances. Users can select from various VM instance types; each optimised for a particular application and set of performance criteria. As these instances come with different CPU, memory, storage, and networking choices, users can customise their virtual environment to meet the particular requirements of their applications. Furthermore, Compute Engine supports the Linux and Windows operating systems, guaranteeing interoperability with various applications and software stacks.

Scalability is one of the main advantages of the Compute Engine. It allows users to easily increase or decrease their VM instances based on demand, allowing for effective resource utilisation and cost savings. Additionally, Compute Engine offers functions like load balancing and auto-scaling, allowing applications to withstand heavy traffic and unexpected increases in demand without experiencing any downtime. To speed up time to market and simplify deployment, users can utilise predefined machine images or build their images.

Compute Engine gives customers access to various tools and innovations by seamlessly integrating with other Google Cloud services. It provides close connectivity with services like Cloud Storage, Cloud Networking, and Identity and Access Management (IAM), enabling customers to create reliable and secure solutions. Compute Engine allows customers to develop and maintain their applications in the Google Cloud environment easily and productively through its robust architecture, rich customization choices, and scalable capabilities.

4.5.3.2 Kubernetes Engine

Kubernetes Engine is a robust and highly scalable managed service for container orchestration. It simplifies containerized application deployment, administration, and scaling, allowing customers to concentrate on their application logic rather than the supporting infrastructure. With the help of the well-known open-source Kubernetes platform, Kubernetes Engine offers a dependable and durable solution for workload management using containers. Using the Kubernetes engine, users can build and manage node clusters, the building blocks for executing containerized applications. These nodes, arranged into clusters, are responsible for hosting and running containers. Kubernetes Engine manages these clusters automatically, guaranteeing effective resource distribution and node task dispersion.

One of Kubernetes Engine's main advantages is managing and deploying complex containerized applications. It includes capabilities that improve application availability, scalability, and reliability, including rolling updates, auto-scaling, and self-healing. Additionally, Kubernetes Engine has sophisticated networking features, enabling containers inside a cluster to connect safely and effectively. The seamless integration of Kubernetes Engine with other Google Cloud services allows users to use various tools and services to develop and deploy applications. It interfaces with services like Cloud Storage, Cloud IAM, and Cloud Monitoring for improved security, access control, and observability for containerised workloads.

4.5.3.3 Cloud Functions

Cloud Functions, a serverless computing service, allow developers to create and deploy event-driven apps and microservices without managing or setting up servers. It enables developers to concentrate entirely on building application code as underlying infrastructure is abstracted away. The infrastructure of Cloud Functions expands dynamically in response to demand, and functions are activated by specified events or requests, according to the Function-as-a-Service (FaaS) paradigm. Developers can concentrate on building code using popular programming languages like Python, Node.js, and Go that handle particular business logic since this serverless method does away with the requirement for manual infrastructure administration.

Numerous event sources are supported by Cloud Functions, including Hypertext Transfer Protocol (HTTP) requests, Pub/Sub messages, events from cloud storage, and events from Firebase Real-time Database. Developers can specify the triggers that start their functions' execution, enabling smooth connection with several GCP services and external systems. This event-driven architecture enables reactive and scalable application development by allowing functions to be called in response to certain events, assuring effective resource utilisation and cost optimisation. The ability of Cloud Functions to scale automatically is another noteworthy feature. The service immediately sets up the required resources to manage the incoming demand when a function is activated. The programme can withstand unexpected increases in traffic or event processing thanks to its dynamic scalability, preventing manual intervention. Reducing resources when there is little or no activity also saves money. Additionally, Cloud Functions interfaces easily with other GCP services, enabling developers to use various tools and services to create complete applications. To allow for smooth data processing, analysis, and storage within the application processes, it connects with services like Cloud Storage, BigQuery, and Firestore.

4.5.3.4 Cloud Storage

Cloud storage provides customers with a dependable and scalable solution for managing and storing their data on the cloud. Without the requirement for on-premises infrastructure, it gives organisations a central location to store and access their data from anywhere globally safely. Google Cloud Storage delivers a distributed storage infrastructure with high durability, availability, and performance.

Google Cloud Storage offers a variety of features and advantages to accommodate various storage needs. First, it provides multiple storage types, each tailored for a particular use case. The Standard storage class offers low-latency access and high availability, making it suited for often accessed data. Nearline storage is good for data accessed less frequently since it has lower costs but a somewhat greater latency. Coldline storage has even lower prices and faster access times and is intended for long-term archiving. The least expensive storage option, archive storage, has the longest retrieval times and is used for infrequently accessed material. Organisations can easily scale their storage resources up or down based on their needs via Google Storage's scalability, ensuring they only pay for the storage they need. Enormous quantities of data can easily be stored and managed with nearly infinite storage

capacity without worrying about storage space limitations. High workloads can be handled by the storage infrastructure, which can also automatically scale to meet growing data demands.

Google Cloud Storage provides strong security protections for data security. For the protection of cloud-stored data, it incorporates encryption at rest. Businesses can employ their customer-managed encryption keys or Google's management for further control over data access. Data security while being sent across the network is ensured by in-transit encryption. Access controls and permissions can be set up to limit authorised users' access to data, improving data confidentiality. Additionally, Google Cloud Storage offers sophisticated data management features. According to preset guidelines, organisations can build up lifecycle policies to automatically transfer data across storage types. Shifting less-often accessible data to less expensive storage tiers offers cost optimisation. Organisations can easily track changes and manage data versions thanks to versioning and object-level controls, which make data management more effective.

4.5.3.5 Cloud Bigtable

GCP offers Cloud Bigtable, a fully managed, highly scalable NoSQL database service. It is highly suited for applications that need real-time data processing and storage since it is built to manage large-scale and high-throughput workloads. To offer a reliable and effective method of handling enormous volumes of structured data, Cloud Bigtable uses the strength of Google's distributed infrastructure. The Bigtable data model, a distributed, multidimensional sorted map, is the foundation for Cloud Bigtable. It arranges data into tables with rows, columns, and timestamped values in each cell. Applications that deal with quickly altering or unexpected data structures benefit greatly from the schema-less feature of this technology, which enables flexible data storage and retrieval.

Petabytes of data can easily be handled by Cloud Bigtable, which distributes the workload across many nodes to guarantee great performance and availability. Cloud Bigtable can accommodate rising data volumes and traffic without impacting performance by dynamically expanding the underlying infrastructure. Its scalability makes it appropriate for use cases, including time-series data analysis, IoT data processing, and high-traffic web applications.

Low-latency access to data is another feature of Cloud Bigtable that enables real-time processing and analytics. Data is stored on SSDs for quick read-and-write operations, and Google's distributed architecture is used to its advantage. This ensures that applications can swiftly retrieve and analyse data, enabling in-the-moment analysis and decision-making. Data is replicated across many nodes and centres to ensure high availability and fault tolerance, which provides the data is still available even in the case of hardware failures or other disturbances. Additionally, organisations can safeguard and restore their data to an initial state through backups and point-in-time recovery capabilities if necessary. With the help of the integration of Cloud Bigtable with Google Cloud IAM, businesses can manage who has access to their data and resources. Data security and confidentiality are guaranteed, and it is shielded from unauthorised access by encryption both in transit and at rest.

4.5.3.6 BigQuery

BigQuery is a fully managed, serverless data warehouse solution designed to handle large amounts of data and conduct quick and effective data analytics and querying. BigQuery is a potent tool that enables businesses to make data-driven choices and unearths useful business knowledge from their massive databases. It is appropriate for organisations dealing with massive volumes of data since it can easily manage petabytes of data. BigQuery parallelizes queries over several nodes using Google's distributed architecture, ensuring speed and quick processing even with massive datasets. Through this scalability, organisations can run sophisticated analytical queries on their data without worrying about infrastructure or performance constraints.

The simplicity of usage of BigQuery is another noteworthy feature, offering a recognisable SQL-like user interface that enables users to interact with the data using their current SQL knowledge and abilities. Its accessibility to a wide spectrum of users, from business executives to data analysts, allows them to explore and analyse data without learning complicated code or specialised expertise. BigQuery is versatile and adaptive to various data intake requirements since it offers a range of data ingestion techniques, such as batch loading, streaming, and data transfer services.

BigQuery also provides sophisticated querying features, basic SQL operations, nested and repeated fields, and user-defined procedures. It enables organisations to conduct complicated analytical operations to get in-depth insights into their data, including aggregations, joins, and window functions. Users can create end-to-end data processing and analytics pipelines inside the GCP ecosystem by integrating BigQuery with other GCP services like Dataflow, Dataproc, and AI Platform. Furthermore, BigQuery offers strong authentication and access controls, enabling businesses to govern who has access to and what they do with their data. Data confidentiality and integrity are guaranteed by encryption while in storage and transport, shielding it from unauthorised access. To help businesses comply with legal obligations and uphold data governance standards, BigQuery also provides data governance tools, including audits, data lineage, and data classification.

4.5.3.7 Cloud Dataflow

Cloud dataflow is built to handle large-scale data processing jobs. It enables scalable and reliable data processing and analytics by offering a single programming architecture for batch and streaming data processing. Cloud Dataflow makes distributed data processing less complicated, allowing businesses to concentrate on data analysis without worrying about the supporting infrastructure. The capability of Cloud Dataflow to process data in batch and streaming modes is one of its important characteristics. This adaptability enables businesses to manage various data processing scenarios, including batch analysis of historical data or real-time analysis of streaming data. Cloud Dataflow manages the challenges of managing data ingestion, parallel processing, and fault tolerance, which enables organisations to process and analyse data in near real time.

Cloud Dataflow follows the Apache Beam programming paradigm and offers a high-level abstraction for describing data processing pipelines. Using well-known programming languages like Java, Python, and Go, this architecture enables developers to create data processing logic independent of the language used to compose it. Developers using the Apache Beam architecture may concentrate on the business logic of their data processing jobs. At the same time, Cloud Dataflow handles the execution and optimisation of the underlying data processing infrastructure.

Cloud Dataflow has built-in fault tolerance methods, including automated checkpointing and data buffering, to ensure that data processing processes can recover from errors and continue processing without losing data. This dependability aspect is essential for systems operating continuously and with absolute data integrity. Furthermore, Cloud Dataflow's integration with other GCP services like BigQuery, Pub/Sub, and Cloud Storage is easy. Through this interface, businesses may create end-to-end data pipelines that enable data ingestion from multiple sources, process and transform that data using Cloud Dataflow, and store or analyse that data using GCP services. Because of this compatibility, organisations may use the whole GCP ecosystem and streamline their data operations.

4.5.3.8 Cloud Dataproc

Cloud Dataproc is a managed big data processing solution enabling businesses to build and manage Apache Hadoop and Apache Spark clusters quickly. It makes adopting and using these powerful data processing frameworks easier, letting companies concentrate on their data analysis requirements rather than the supporting infrastructure. Organisations can easily construct a cluster with all the essential software and configurations with a few mouse clicks or a single command. Without the need for intricate and drawn-out setup procedures, this agility enables organisations to quickly configure their data processing environment and begin analysing their data.

Cloud Dataproc has automated scaling options that dynamically change the cluster's size according to the workload. This ensures businesses have the computing power to effectively perform their data processing activities. The cluster can scale up at times of high activity and down during low workload to reduce expenses. Organisations can optimise resource use and reduce operating costs via this elasticity. Moreover, integrating Cloud Dataproc with other GCP services is another important aspect. It enables businesses to create end-to-end data pipelines by integrating with services like BigQuery, Cloud Storage, and Cloud Pub/Sub. Data may be imported from multiple sources into Cloud Storage, processed, and examined using Hadoop or Spark clusters from Cloud Dataproc. Then the outcomes can be saved in BigQuery for additional examination or visualisation. Organisations can track the performance and well-being of their clusters with the help of Cloud Dataproc's monitoring and logging features. By giving organisations visibility into resource utilisation, task status, and cluster metrics, they can optimise their data processing workflows and address any problems that may develop.

4.5.3.9 AutoML

AutoML, which stands for Automated Machine Learning, is a cutting-edge feature in GCP that aims to democratise the creation of machine learning models. It uses AI to speed up the creation of high-quality machine learning models by automating several steps in the machine learning pipeline. This makes the pipeline accessible to users with little or no machine learning knowledge.

With the help of AutoML, users can train machine learning models without having to design features or choose challenging algorithms manually. Analysing the incoming data and locating pertinent characteristics that might contribute to precise predictions automates the feature extraction. Users are freed from manually engineering features, which might take time and require subject knowledge. Furthermore, choosing a model is made easier by AutoML. It examines a wide range of machine learning algorithms, architectures, and hyperparameter combinations to determine the ideal configuration that results in optimal performance. By automating this selection procedure, AutoML removes the requirement for users to possess in-depth knowledge of various machine learning approaches, enabling them to concentrate on the problem domain rather than the underlying algorithms.

Finding the ideal values for different parameters that control how machine learning models behave is a difficulty that AutoML also addresses, and it is known as hyperparameter tuning. It uses advanced optimisation approaches like Bayesian optimisation or evolutionary algorithms to explore the hyperparameter space and find the configurations that maximise model performance. By automating the process, manual hyperparameter tweaking takes a great deal less time and effort. AutoML provides customers with a user-friendly interface and clear processes leading them through the entire machine learning pipeline. It provides visualisations and performance indicators to assist users in comprehending and assessing the calibre of their models. Additionally, AutoML offers tools for model deployment, enabling customers to utilise their developed models as APIs quickly or include them in their programmes.

4.5.3.10 TensorFlow

TensorFlow, a powerful open-source machine learning framework, is one of the primary services provided by GCP, enabling researchers and developers to create and deploy machine learning models at scale. TensorFlow, created by the Google Brain team, offers a versatile and effective platform for putting different deep learning algorithms into practice and carrying out extensive distributed training and inference.

TensorFlow offers a computational graph abstraction that enables users to describe intricate mathematical processes as a network of linked nodes. This graph shows the flow of information and calculations, with nodes denoting mathematical operations and edges representing the information that passes across them. This method is ideal for training deep neural networks on huge datasets because it offers flexibility and facilitates effective parallelization. Numerous machine learning tasks, including classification, regression, object identification, natural language processing, and others, are supported by TensorFlow. It offers a complete collection of tools and libraries for data preparation, model building, training, and assessment.

With TensorFlow, users can quickly create intricate neural network designs by utilising high-level APIs like Keras or the low-level TensorFlow API to manage model construction and optimisation precisely.

One of TensorFlow's primary advantages is scaling computations over several devices and distributed systems. Users can train models on huge datasets in parallel because it supports distributed training over clusters of computers. Additionally, TensorFlow interfaces smoothly with the GCP infrastructure, allowing users to expedite training and inference using strong tools like Google's Cloud Tensor Processing Unit (TPUs). Additionally, TensorFlow offers comprehensive model deployment and serving support. It provides solutions like TensorFlow Serving and TensorFlow Lite for delivering models in real-world settings or on devices with limited resources, which enables users to quickly and effectively incorporate their trained models into practical applications and provide large-scale forecasts.

4.5.4 Comparison of Cloud Computing Providers

Organisations looking for the best match for their unique needs must compare cloud computing providers. Three significant cloud computing companies will be examined in this comparison: AWS, Microsoft Azure, and GCP. Each supplier provides various services and features to meet different needs and use cases.

AWS has the broadest selection of services and the highest market share since it invented cloud computing. It provides many services, including computing, storage, databases, networking, machine learning, and more, by offering scalable and dependable infrastructure, including Amazon S3 for object storage and Amazon EC2 for virtual machine instances. Additionally, AWS provides cutting-edge analytics tools like Amazon Athena for querying and analysis and Amazon Redshift for data warehousing. AWS is a good fit for businesses that need flexibility, scalability, and a developed ecosystem, as it offers broad services.

Microsoft Azure, a close competitor to AWS, provides a broad range of services that play to its strengths in hybrid cloud solutions and business integration. Organisations that rely significantly on Microsoft technology can find Azure appealing because of its seamless integration with existing Microsoft tools and services. Azure focuses on hybrid cloud capabilities, allowing businesses to combine on-premises infrastructure with cloud resources. Azure is a good choice for companies moving smoothly from on-premises settings to the cloud due to its strong corporate emphasis and integration features.

GCP has been quite popular in recent years despite being more recent than AWS and Azure, which provides data analytics, AI and machine learning, databases, computation, storage, and various other services. With services like BigQuery for data warehousing, Cloud Machine Learning Engine for training and deploying machine learning models, and Dataflow for stream and batch data processing, GCP distinguishes itself via its data analytics and machine learning competence. GCP also emphasises its infrastructure's performance and ability to connect to worldwide

networks, making it a great option for businesses with data-intensive workloads and an emphasis on analytics and machine learning.

Figure 4.11 compares the cloud providers regarding the number of regions and availability zones, while Fig. 4.12 reveals the services of the three biggest cloud computing services.

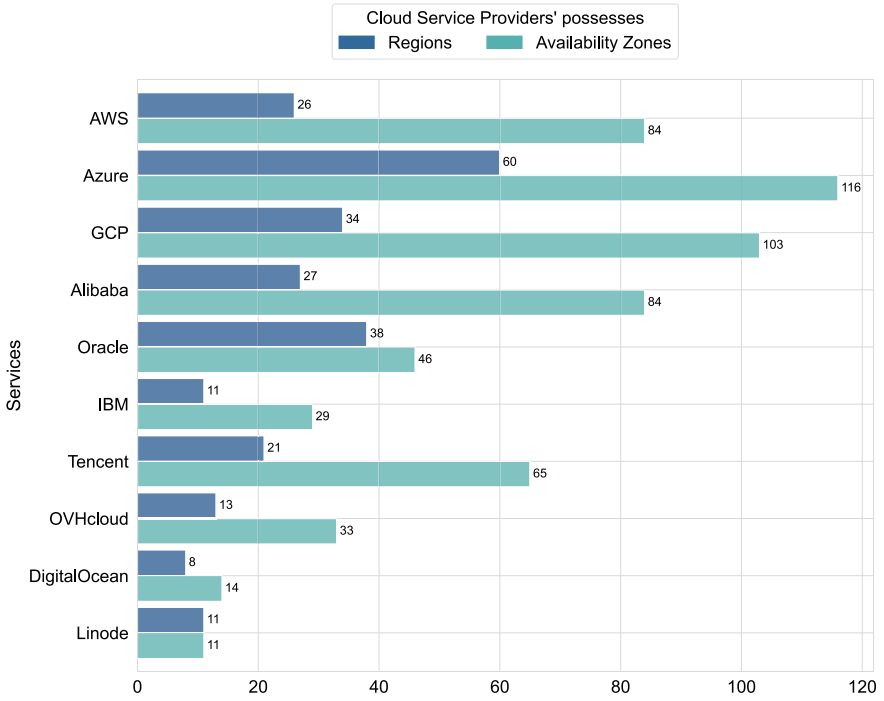


Fig. 4.11 The number of regions and availability zones that each vendor possesses

	Azure	aws	PRODUCT
Google Compute Engine	Azure Virtual Machines	Elastic Compute Cloud (EC2)	Infrastructure as a Service (IaaS)
Google App Engine	Azure Cloud Services	AWS Elastic Beanstalk	Platform as a Service (PaaS)
Google Cloud SQL	Azure SQL Database	Amazon Relational Database Service	Database as a Service (DaaS)
Google Cloud Bigtable	Azure Table Storage	Amazon Dynamo DB	Scalable SQL database services
Google BigQuery	Azure SQL Database	Amazon Redshift	Relational Databases
Google Cloud Functions	Azure Functions	AWS Lambda	Serverless Applications
Google Cloud Datastore	Azure Cosmos DB	Amazon Simple DB	Highly Scalable NoSQL Database Services
Google Storage	Azure Storage	Amazon Simple Storage Service (S3)	Storage of object, blocks and files.

Fig. 4.12 The services of cloud providers

4.6 Learning Outcomes of the Chapter

- **Understanding Cloud Computing**: Defining the concept of cloud computing and its relevance to modern computing architectures.
- **Exploring the History of Cloud Computing**: Examining the evolution of cloud computing, including computing generations that have shaped its development.
- **Understanding Cloud Computing Units**: Investigating cloud computing service models and deployment models that form the foundation of cloud computing units.
- **Exploring Multi-Cloud Strategies in Big Data Analytics**: Analysing the use of multi-cloud strategies in big data analytics and understanding their implications.
- **Evaluating Cloud Computing Platforms for Big Data Analytics**: Assessing prominent cloud computing platforms and comparing cloud computing providers.

References

1. K. Hwang, J. Dongarra, G.C. Fox, *Distributed and Cloud Computing: From Parallel Processing to the Internet of Things*. (Morgan kaufmann, 2013)
2. R. Ranjan, I.S. Thakur, G.S. Aujla, N. Kumar, A.Y. Zomaya, Energy-efficient workflow scheduling using container-based virtualization in software-defined data centers. IEEE Trans. Ind. Inf. **16**(12), 7646–7657 (2020)
3. R. Prodan, E. Torre, J.J. Durillo, G.S. Aujla, N. Kummar, H.M. Fard, S. Benedikt, Dynamic multi-objective virtual machine placement in cloud data centers, in *2019 45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)* (2019), pp. 92–99
4. C. Yang, Q. Huang, Z. Li, K. Liu, F. Hu, Big data and cloud computing: innovation opportunities and challenges. Int. J. Digital Earth **10**(1), 13–53 (2017)
5. Y. Lu, X. Xu, Cloud-based manufacturing equipment and big data analytics to enable on-demand manufacturing services. Robot. Comput. Integr. Manufact. **57**, 92–102 (2019)
6. F. Habeeb, K. Alwasel, A. Noor, D.N. Jha, D. AlQattan, Y. Li, G.S. Aujla, T. Szydlo, R. Ranjan, Dynamic bandwidth slicing for time-critical iot data streams in the edge-cloud continuum. IEEE Trans. Ind. Inf. **18**(11), 8017–8026 (2022)
7. K.A. Kumar et al., Big data characteristics, classification and challenges-a review. Turkish J. Comput. Math. Edu. (TURCOMAT) **12**(12), 4236–4243 (2021)
8. Z.N. Rashid, S.R. Zebari, K.H. Sharif, K. Jacksi, Distributed cloud computing and distributed parallel computing: A review, in *2018 International Conference on Advanced Science and Engineering (ICOASE)*. (IEEE, 2018), pp. 167–172
9. J. Dean, S. Ghemawat, Mapreduce: simplified data processing on large clusters. Commun. ACM **51**(1), 107–113 (2008)
10. U. Demirbaga, D.N. Jha, Social media data analysis using mapreduce programming model and training a tweet classifier using apache mahout, in *IEEE 8th International Symposium on Cloud and Service Computing (SC2)*. (IEEE, 2018), pp. 116–121
11. R. Vaupel, *High Availability and Scalability of Mainframe Environments using System z and z/OS as Example*. (KIT Scientific Publishing, 2014)
12. A.R. Lebeck, X. Fan, H. Zeng, C. Ellis, Power aware page allocation. ACM SIGARCH Comput. Architect. News **28**(5), 105–116 (2000)

13. M. Revett, I. Boyd, C. Stephens, Network computing: a tutorial review. Electron. Commun. Eng. J. **13**(1), 5–15 (2001)
14. A. Sunyaev, A. Sunyaev, *Internet Computing*. (Springer, 2020)
15. J. Joseph, C. Fellenstein, *Grid Computing*. (Prentice Hall Professional, 2004)
16. D. Georgakopoulos, P.P. Jayaraman, M. Fazia, M. Villari, R. Ranjan, Internet of things and edge cloud computing roadmap for manufacturing. IEEE Cloud Comput. **3**(4), 66–73 (2016)
17. M.U. Bokhari, Q.M. Shallal, Y.K. Tamandani, Cloud computing service models: A comparative study, in *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*. (IEEE, 2016), pp. 890–895
18. R. Buyya, J. Broberg, A.M. Goscinski, *Cloud Computing: Principles and Paradigms*. (John Wiley & Sons, 2010)
19. A. Yousif, M. Farouk, M.B. Bashir, A cloud based framework for platform as a service, in *2015 International Conference on Cloud Computing (ICCC)*. (IEEE, 2015), pp. 1–5
20. D. Dziembek, T. Turek, Characteristics and application of unified communications as a service (ucaas) in enterprises. Inform. Ekonomiczna **4**(50), 47–65 (2018)
21. G. Laatikainen, A. Ojala, Saas architecture and pricing models, in *2014 IEEE International Conference on Services Computing*. (IEEE, 2014), pp. 597–604
22. V. Kundra, *State of Public Sector Cloud Computing* (Federal Chief Information, 2010)
23. M. Olowu, C. Yinka-Banjo, S. Misra, H. Florez, A secured private-cloud computing system, in *Applied Informatics: Second International Conference, ICAI*, Madrid, Spain, November 7–9, 2019, Proceedings 2. (Springer, 2019), pp. 373–384
24. A. Srinivasan, M.A. Quadir, V. Vijayakumar, Era of cloud computing: A new insight to hybrid cloud. Procedia Comput. Sci. **50**, 42–51 (2015)
25. A. Marinos, G. Briscoe, Community cloud computing, in *Cloud Computing: First International Conference, CloudCom* (Beijing, China, December 1–4, 2009). Proceedings 1. Springer **2009**, 472–484 (2009)
26. J. Hong, T. Dreibholz, J.A. Schenkel, J.A. Hu, An overview of multi-cloud computing, in *Web, Artificial Intelligence and Network Applications: Proceedings of the Workshops of the 33rd International Conference on Advanced Information Networking and Applications (WAINA-2019) 33*. (Springer, 2019), pp. 1055–1068
27. Analytics pipeline with aws services. AWS. [Online]. Available: https://docs.aws.amazon.com/whitepapers/latest/data-warehousing-on-aws/analytics-pipeline-with-aws-services.html
28. Azure architecture center. Microsoft. [Online]. Available: https://learn.microsoft.com/en-us/azure/architecture/
29. Analytics pipeline with aws services. Google Cloud Platform blog. [Online]. Available: https://cloudplatform.googleblog.com/2015/04/big-data-cloud-way.html

Further Reading

30. R. Ranjan, K. Mitra, P.P. Jayaraman, L. Wang, A.Y. Zomaya, *Handbook of Integration of Cloud Computing, Cyber Physical Systems and Internet of Things*. (Springer, 2020)
31. M. Trovati, R. Hill, A. Anjum, S. Ying Zhu, L. Liu, *Big-data Analytics and Cloud Computing*. (Springer, 2015)

Big Data Analytics Platforms

5

Processed data is information,
Processed information is knowledge,
Processed knowledge is wisdom.

— Ankala V. Subbarao

This chapter explores big data analytics platforms by shedding light on their essential characteristics for processing and deciphering vast datasets. Encompassing distributed computing, data ingestion, storage, processing, and advanced analytics, the chapter underscores the critical significance of scalability, security, and governance in this intricate landscape. The narrative probes into the desired properties of robust big data systems, including fault tolerance, scalability solutions, generalisation, and minimal maintenance. Moving through key processing systems, the focus extends to Hadoop's MapReduce paradigm and the versatile capabilities of Apache Spark, with discussions on Spark's deployment on Yet Another Resource Negotiator (YARN) and practical case studies. Additionally, the chapter delves into data engineering with Apache Hive, data ingestion using Apache Sqoop, and streaming data with Apache Flume. Culminating the exploration is an insightful overview of Apache Mahout, elucidating its role in distributed machine learning for big data analytics through installation, configuration, and a compelling case study.

5.1 Main Characteristics of Big Data Analytics Platforms

Big data analytics platforms are comprehensive software frameworks that provide businesses access to enormous, complicated datasets for processing, analysis, and insight extraction. These systems offer the required capabilities and infrastructure

Ü. Demirbaga et al., *Big Data Analytics*,
https://doi.org/10.1007/978-3-031-55639-5_5

for managing big data's volume, velocity, diversity, and veracity. Big data analytics platforms' main characteristics, elements, and features are discussed below.

5.1.1 Distributed Computing

Distributed computing refers to using several networked nodes but running as a single entity to perform computational tasks and resolve challenging issues [1]. Distributed computing is essential to big data analytics because it allows for the concurrent processing and analysis of enormous amounts of data, making data processing effective and scalable. The underlying idea behind distributed computing is to break down a computational activity into smaller sub-tasks and spread them among the cluster's nodes. Each node processes the allotted data individually, and the outcomes are integrated or aggregated to get the final output. Due to the effort being divided across several workstations, this parallel processing strategy considerably decreases the time needed to compute huge datasets.

One of the main benefits of distributed computing is its capacity to address big data's inherent problems, such as the amount, diversity, and velocity of data. Distributed computing systems may extend horizontally and successfully scale up to meet the increasing needs of data processing by dividing the data and processing it over several nodes. Because of its scalability, the capacity of organisations to handle and analyse ever-larger datasets is not constrained by the capabilities of a single machine. Moreover, distributed computing offers resistance to errors and fault tolerance. In a distributed system, the analytics tasks can run without interruption, even if one or more nodes fail. Data replication, partitioning, and distributed consensus methods are used for fault tolerance.

Many frameworks and technologies, including Apache Hadoop, Apache Spark, and distributed databases like Apache Cassandra, have evolved to support distributed computing. These frameworks include programming paradigms, APIs, and distributed file systems that abstract the difficulties of distributed computing, facilitating the use of distributed systems by developers and data scientists. The intricate details of distributed computing systems are discussed in the following sections, including their underlying architectures, algorithms, and techniques.

5.1.2 Data Ingestion and Integration

Data ingestion and integration are essential steps in big data analytics that make it possible to gather, harmonise, and consolidate data from many sources. Effective data ingestion and integration processes are critical for big data analytics solutions to guarantee data quality, integrity, and usefulness. Organisations may efficiently use their data assets, generate insightful information for decision-making, and achieve a competitive advantage by tackling the issues related to data variety, velocity, and volume.

Data ingestion describes obtaining data from many sources, including databases, files, sensors, social media feeds, and streaming platforms [2]. Data must be extracted from various sources and brought into the big data environment for additional processing. The ingestion procedure should be able to deal with massive amounts of data and accommodate multiple data types and formats. On the other hand, data integration entails merging and unifying data from several sources into a single view. This procedure also involves data cleansing, transformation, and harmonisation to guarantee data compatibility and consistency across various datasets. As part of the integration, handling missing values, fixing data quality problems, and standardising data formats and schemas may also be necessary.

Big data analytics uses various methods and tools for ingesting and integrating data. These include data integration platforms, data streaming frameworks, ETL procedures, and data virtualization methods. Technologies like Apache Kafka, Apache Nifi, and Enterprise Service Buses (ESBs) are often employed for data intake and real-time data integration.

5.1.3 Data Storage and Management

Data storage and management are the fundamental components of big data analytics, which lay the groundwork for effective and dependable data processing and analysis. The sheer amount, speed, and variety of data created make standard storage and administration methods in the big data space frequently ineffective. Specialised storage and administration solutions have been developed to meet the specific needs of big data analytics.

The necessity for scalable and distributed structures to manage the enormous volumes of created data is one of the main issues in big data storage. In big data, distributed file systems—like the HDFS—have become more prevalent. With fault tolerance and fast throughput, these file systems are made to handle and store data across clusters of affordable hardware. Distributed file systems provide efficient data access and parallel processing by dividing data into smaller pieces and spreading them over numerous computers. NoSQL databases have gained popularity as options for huge data storage and administration alongside distributed file systems. NoSQL databases, in contrast to conventional relational databases, allow for horizontal scaling and flexible schema designs, making them ideal for managing unstructured and semi-structured data. Cassandra, Apache HBase, and MongoDB are examples of NoSQL databases. These databases provide dependable data storage and retrieval in big data analytics by offering high availability, fault tolerance, and scalability.

Data integration, cleansing, governance, and other processes are all part of managing data. The practice of merging data from several sources into a single perspective to enable thorough analysis is known as data integration. Effective integration strategies are needed to harmonise and combine data from diverse sources when dealing with the heterogeneous data formats and structures that big data analytics frequently requires. Another key component of data management in big data analytics is data cleaning, also known as data quality management. Due to the enormous amount of

data being processed, it is crucial to guarantee accuracy, completeness, and consistency. To improve the quality and dependability of data, data cleansing procedures encompass deleting duplicates, addressing missing values, and resolving discrepancies. Establishing rules, processes, and controls to guarantee data security, privacy, and compliance requires data governance practices. This includes establishing data access credentials, implementing encryption techniques, and complying with legal mandates like GDPR or HIPAA. The rules and procedures necessary for managing data at every stage of its lifespan, from collection and storage to analysis and disposal, are provided by data governance frameworks.

5.1.4 Data Processing and Analysis

Big data analytics relies heavily on data processing and analysis to help organisations extract valuable information and intelligence from massive and complicated databases. Traditional data processing and analysis methods have failed to keep up with the exponential expansion of data volume and the proliferation of data sources. To this end, specialised strategies and frameworks have been developed to handle the specific difficulties associated with big data analytics.

Data processing is converting and modifying unstructured data in a more organised and practical manner. It includes several operations, such as data gathering, integration, transformation, and cleansing. These procedures are completed to ensure the data is accurate, reliable, and compatible and to make analysis more effective. Scalable processing frameworks like Apache Hadoop and Apache Spark are frequently utilised in big data analytics. These frameworks use parallel processing and distributed computing techniques to manage large datasets and run calculations quickly and effectively. On the other hand, data analysis focuses on drawing conclusions and patterns from the prepared data. Statistical, machine learning, and data mining approaches are used to find hidden connections, practices, and anomalies in the dataset. Data analysis objectives are gaining a deeper knowledge of the underlying data, making predictions, and coming to data-driven choices. Big data analytics uses various methods and models, including regression analysis, clustering, classification, and recommendation systems.

Big data analytics platforms provide tools, libraries, and frameworks to simplify data processing and analysis. These systems provide mechanisms for sophisticated analytics, data exploration, and visualisation. They enable interaction with the data, the execution of complex queries, and using pre-built models and algorithms. Several well-known platforms for big data analytics include Apache Hadoop, Apache Spark, and GCP's BigQuery. Due to the advent of real-time and streaming data, real-time data processing and analysis are now required. Organisations must be able to examine data as it is received to make timely decisions and respond to shifting conditions. Today, frameworks for stream processing, including Apache Kafka and Apache Flink, can manage continuous data streams and offer real-time analytics.

5.1.5 Machine Learning and Advanced Analytics

Big data analytics includes machine learning and advanced analytics, which allows businesses to analyse detailed information better and get useful insights and forecasts [3]. To find hidden patterns, spot abnormalities, and make data-driven decisions in the context of big data, sophisticated analytics techniques, and machine learning algorithms are essential.

Creating algorithms and models for machine learning enables computers to learn from data and make predictions or actions without explicit programming. It uses statistical methods, pattern recognition, and computational algorithms to analyse huge amounts of data automatically and find significant patterns, correlations, and trends. Several categories can be used to categorise machine learning algorithms, including supervised, unsupervised, and reinforcement learning. These algorithms are trained on labelled or unlabeled data to identify patterns and produce predictions or classifications. Machine learning algorithms are used in big data analytics to perform classification, regression, grouping, and recommendation on huge and complicated datasets. These algorithms find hidden connections, forecast outcomes, categorise data into useful categories, and customise user experiences. The deployment of machine learning models to process new data in batch or real-time processing settings begins with training them on past data. On the other hand, advanced analytics refers to the employment of sophisticated analytical methods outside of typical statistical analysis. To extract insights from numerous and unstructured data sources, it entails integrating multiple techniques such as machine learning, natural language processing, text mining, network analysis, and deep learning. Organisations can use advanced analytics approaches to analyse data in great detail, find intricate patterns, and better understand consumer behaviour, market trends, and operational performance.

Big data analytics applies machine learning and sophisticated analytics methods to various datasets, including text, photos, social media data, structured and unstructured data, and structured and unstructured data. These methods assist businesses with sentiment analysis, picture identification, fraud prevention, customer segmentation, predictive maintenance, and customised suggestions. The fast execution of machine learning algorithms and advanced analytics activities on massive datasets is made possible by the scalability and parallel processing capabilities of big data analytics platforms like Apache Spark and TensorFlow. In addition, deep learning, a subfield of machine learning that focuses on developing neural networks with numerous layers, is progressing in big data analytics. Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), two types of deep learning algorithms, have demonstrated outstanding performance in tasks including audio and picture recognition, natural language processing, and anomaly detection. These methods enable extracting subtle patterns and characteristics from multidimensional, complicated data.

5.1.6 Data Visualisation and Reporting

Data visualisation and reporting are important processes in big data analytics, enabling companies to communicate effectively and gain knowledge from detailed information from the bulk of data [4]. Visual representations and reports are effective tools for presenting information in a way that is visually appealing and intuitive, enhancing comprehension, decision-making, and communication.

Data visualisation is graphically representing data using graphs, maps, and other visual components. It tries to offer a visual overview of detailed statistics, making it simpler to spot trends, patterns, and outliers. Users may explore and interact with huge amounts of data using visualisation techniques, which enables them to obtain important insights and find significant links that can be buried in the raw data. Organisations successfully convey complicated information to stakeholders by visualising the data, which makes it simpler for them to understand the main points and make wise decisions. Various large datasets, including structured, unstructured, and semi-structured data, are visualised using different data visualisation techniques provided by platforms and tools like Tableau, Power BI, and D3.js, where users can build dynamic dashboards, reports, and infographics. These tools allow users to alter visualisations, utilise different chart styles, and include interactive elements like filtering, zooming, and drill-down options. Organisations can efficiently investigate and deliver insights from big data analytics using these skills. Furthermore, reporting entails creating summaries of reports, dashboards, and presentations highlighting the most important discoveries and insights from big data analytics. Reports are a useful tool for outlining the most crucial elements of the data analysis process while delivering critical information in a structured and concise manner. They give stakeholders a complete picture of the analytics findings to comprehend the ramifications and make fact-based decisions. Reports can contain various components, including text, tables, charts, and visualisations, giving readers a comprehensive understanding of the analysed data and its ramifications.

Reporting is often automated and produced regularly, offering pertinent updates and insights in big data analytics. The development and delivery of reports may be automated by organisations using reporting tools and platforms, ensuring stakeholders can access the most recent data. Additionally, interactive elements that enable users to explore and engage with the given insights can be incorporated into sophisticated reporting methodologies. Stakeholders can better grasp the underlying trends and patterns through increased user interaction and the ability to go deeper into the data.

5.1.7 Scalability and Performance

Scalability and performance determine the ability of big data systems to manage massive datasets and provide prompt results effectively depending on their scalability and performance. The power of a system or solution to handle growing data volumes,

workloads, and user demands without sacrificing efficiency or dependability is called scalability in the context of big data.

Horizontal and vertical scaling are included in the concept of scalability in big data analytics. To transfer the processing and storage burden throughout a cluster or network, horizontal scalability entails adding computing resources, such as servers or nodes. This strategy makes it easier to process data in parallel and allows the system to manage large amounts of data and expand workloads. On the other side, vertical scalability refers to boosting the capabilities of individual nodes or computers by upgrades to their hardware specifications, such as raising memory or processing power. Systems can handle increasingly complicated algorithms and complete computationally demanding tasks more quickly through vertical scaling.

Performance optimisation greatly impacts big data analytics, emphasising data processing, analysis effectiveness, and speed. Users can get insights and make quick choices because of high-performance systems' ability to execute complex algorithms and queries in reasonable amounts of time. To ensure that analytics jobs are finished within realistic time limitations, performance improvements are essential to reducing latency and increasing throughput. The development of effective data storage and retrieval systems, the use of parallel processing strategies, and algorithm optimisation for distributed computing settings are frequently included in these improvements.

Various techniques and technologies are used in big data analytics to achieve scalability and performance. Scalable and high-performance data processing is made possible by distributed computing frameworks like Apache Hadoop and Apache Spark. These frameworks allow running processes concurrently across a group of computers, speeding up the processing of enormous datasets. Furthermore, by lowering data access latency and accelerating computation, techniques like data partitioning, caching, and in-memory processing contribute to better performance.

Adopting SSDs and high-speed networks, among other hardware developments, may greatly improve the performance of data storage and processing systems. For various types of calculations, especially machine learning algorithms, hardware acceleration technologies offer significant speedups, such as Graphics Processing Units (GPUs) and Field-programmable Gate Arrays (FPGAs). Moreover, achieving scalability and performance in big data analytics relies heavily on algorithm and workflow optimisation. The quantity of data handled can be decreased, and query execution time can be increased using techniques like data compression, indexing, and pruning. Furthermore, parallel algorithms, distributed data structures, and effective data transmission protocols also help improve the performance of distributed computing systems.

5.1.8 Security and Governance

Security and governance are essential to big data analytics to preserve data privacy, guarantee data integrity, and adhere to legal standards. Strong security measures and efficient governance frameworks are crucial to reduce risks and protecting data

confidentiality, availability, and integrity in big data, where enormous volumes of sensitive information are processed and analysed.

Big data analytics security includes various safeguards, such as data encryption, access management, and authentication techniques. Both data in transit and at rest are protected using encryption techniques, such as symmetric and asymmetric encryption, to prevent unauthorised parties from accessing or changing the data. Role-based Access Control (RBAC) and Attribute-based Access Control (ABAC) systems govern and limit data access based on user roles and privileges. Strong authentication techniques are used to confirm the identity of people accessing the system or data, such as Multi-Factor Authentication (MFA) and biometric authentication.

Data governance is essential in guaranteeing the appropriate and ethical use of data in addition to these technological security measures. Guidelines and policies for data processing, storage, sharing, and retention are provided through data governance frameworks. They create accountability, specify data management and compliance processes, and clarify roles and duties. Data quality, data lineage, and data lifecycle management are further challenges that data governance frameworks address, ensuring that data is correct, traceable, and appropriately maintained throughout its existence. Moreover, big data analytics places a high priority on observing legal and privacy standards. Organisations must abide by all applicable laws and regulations, such as the GDPR and the HIPAA, to preserve individuals' rights to privacy and ensure the lawful processing of personal data. Obtaining user consent, pseudonymizing or anonymizing data, setting data retention guidelines, and ensuring accountability and openness in data processing operations are some examples of compliance procedures.

Organisations use a combination of technological solutions and organisational practices to improve security and governance in big data analytics. Effective data security controls are implemented to identify and reduce possible security breaches, such as network firewalls, intrusion detection systems, and data loss prevention techniques. Routine security audits and vulnerability assessments are performed to identify and address security gaps. Organisations also create data governance committees or stewardship positions to supervise data management procedures, uphold regulations, and promote best practices.

5.2 Desired Properties of a Big Data System

The term big data, which emerged with the increase in the amount and complexity of data, raised the question of "*How can I process big data at a reasonable cost and time?*" and paved the way for the emergence of big data systems. The size and complexity of data have increased the number of challenges big data systems must overcome. Scalability, one of the main challenges of big data, forms the backbone of the design of big data systems. In addition, a big data system's performance and resource efficiency are also expected features. The desired properties in an ideal big data system are detailed.

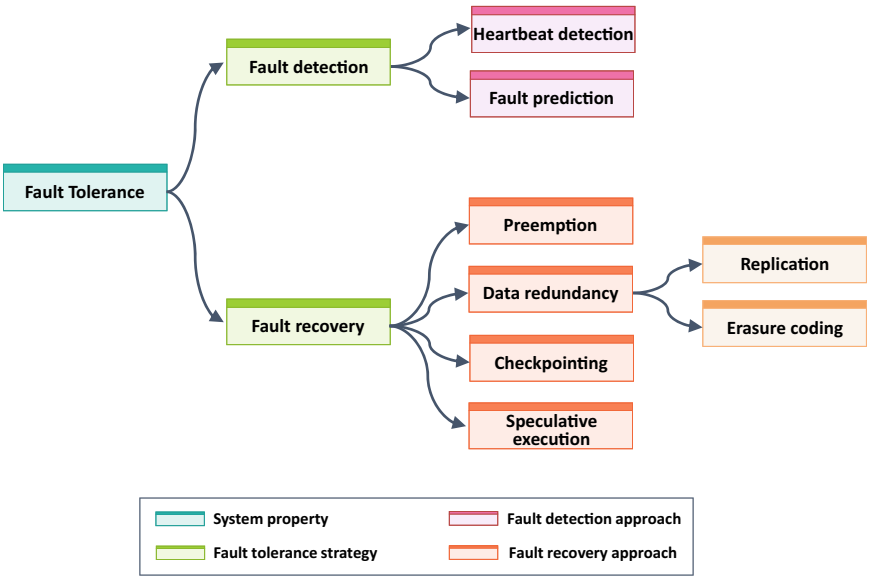


Fig. 5.1 Classification of fault tolerance techniques

5.2.1 Robustness and Fault Tolerance

Fault tolerance is the ability to continue operating despite faults occurring during execution [5]. As the number of machines increases, so does failure probability, making fault tolerance one of the essential components in big data systems [6]. An error-free execution prevents performance degradation and provides excellent savings in matters such as time and energy. The study [7] has shown that a system consisting of 1000 super reliable computers with an average lifespan of 30 years is expected to fail daily. The recovery period for the problems can take approximately two days. Big data systems (e.g. Hadoop,[1] Spark[2]) based on the principle of processing data in a parallel and distributed manner have aimed to manage fault tolerance by offering various fault tolerance approaches such as data redundancy, checkpoint and speculative execution providing fault tolerance. However, even these suggested solutions cannot wholly prevent the frequent errors from reducing performance.

A system with maximum performance and fault tolerance is based on two main concepts, fault detection, and recovery [8], represented in Fig. 5.1 with their subclasses. Fault detection, the first step of a fault-tolerant system, allows for quickly identifying flaws as soon as they reveal themselves. Stable fault detection techniques, such as heartbeat and fault prediction, are utilised in large-scale systems. The heartbeat technique is built around the explicit and routine transmission of heartbeat

[1] https://hadoop.apache.org/.
[2] https://spark.apache.org/.

signals between two components during periods of error-free operation [9]. Fault recovery returns the faulty component to normal behaviour after detecting a fault. Data replication and erasure coding approaches ensure data availability and reliability in case of data redundancy in storage systems. Multiple copies of the original data are simply created and stored on different disks, which is adopted by Google File System (GFS),[3] HDFS,[4] and Windows Azure Storage (WAS),[5] to provide high availability and to ensure fault tolerance [10]. Facebook uses Reed-Solomon coding (RSC), a widely used type of erasure coding [11]. Besides, preemption, checkpointing, and speculative execution are the methods used to provide fault tolerance in big data processing systems [12]. Checkpointing is used to record the latest status of a process or an active node on another standby node in case of failure for quick and effective recovery. Checkpointing is commonly adopted in processes with low latency, such as real-time or stream processing. Next, preemption is another fault recovery strategy the task scheduler typically employs to offer effective fault recovery when the cluster reaches its maximum resource capacity. A task prevention policy should be defined to terminate low-priority jobs and allow room to re-run high-priority tasks if they fail. As a final, speculative execution works by duplicating the active tasks that perform poorly compared to other tasks detected by a certain threshold. The speculative task is executed in another node from where the actual task is running. The task is terminated if the counterpart is completed before it. Hadoop YARN[6] uses this approach for fault tolerance.

5.2.2 Scalability

Scalability is another key factor for big data systems, which refers to maintaining system performance by adding new resources to the existing system in the context of increased data or load.

Big data requires storage, traffic, and processing capacity (for real time/stream processing). In this case, not expanding the system infrastructure will cause bottlenecks in data operations. Several reasons indicate the need for system scaling, which causes poor performance.

- **High CPU usage:** Big data projects are data processing operations requiring high computing power, often CPU-bound. Compressed columnar storage architecture, complex data structure, different types of data (i.e. structured, unstructured data, and semi-structured data), and in-memory data formats are increasingly used for data analytics, increasing CPU utilisation. That is why high CPU usage is the most common and visible bottleneck in big data systems.

[3] https://cloud.google.com/blog/products/storage-data-transfer.
[4] https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html.
[5] https://docs.microsoft.com/en-us/azure/storage/.
[6] https://hadoop.apache.org/docs/stable/hadoop-yarn/hadoop-yarn-site/YARN.html.

- **Low memory:** Servers that do not have enough memory to handle the workload greatly slow down the overall system and cause frequent system crashes. In-memory databases, commonly used for big data, store data collections directly in the working memory of machines to provide high-speed access. Moreover, Apache Spark, one of the most popular frameworks for big data analysis, does in-memory processing, which helps to run incredibly iterative algorithms and requires high memory capacity. With the increasing amount of data, systems with insufficient memory may come to a standstill and require a Random Access Memory (RAM) upgrade.
- **High disk I/O:** Four basic operations (i.e. create, read, update, and delete) are frequently executed every second on the same disks in big data systems. Hadoop MapReduce, a programming model for processing huge amounts of data across thousands of clusters, performs disk read/write in each iteration, which reads data from the disk and again writes temporary data back to the disk. When these data operations and processing are performed on the same disk, high I/O latency occurs, creating a bottleneck.
- **High disk usage:** Maxed-out server disks create bottlenecks as they slow down I/O operations. In addition, servers with insufficient storage capacity cause incoming data not to be recorded, thus data loss, especially in systems where streaming data is stored. Servers reaching a certain storage capacity can cause bottlenecks and need to be scaled.

5.2.2.1 Scaling Solutions for Big Data

There are two types of scaling commonly used to meet the growing need for hardware systems: scaling up and scaling out.

Vertical scaling (scaling up): It is the way to get a machine with more powerful processors, more memory, and storage capacity. With this method, the software or programming language used on the existing computer can be used. However, this method offers a short-term solution and only allows storing and analysing a certain amount of data.

Horizontal scaling (scaling out): Horizontal scaling is a method of adding servers to the existing server for parallel computing instead of replacing the basic hardware of the computer, such as CPU, memory, and disk. It provides a long-term solution by adding machines to the server in use according to the increasing need. Although it brings some difficulties, it is a magnificent solution for big data over vertical scaling.

Figure 5.2 presents how these two methods are applied, along with their advantages and disadvantages.

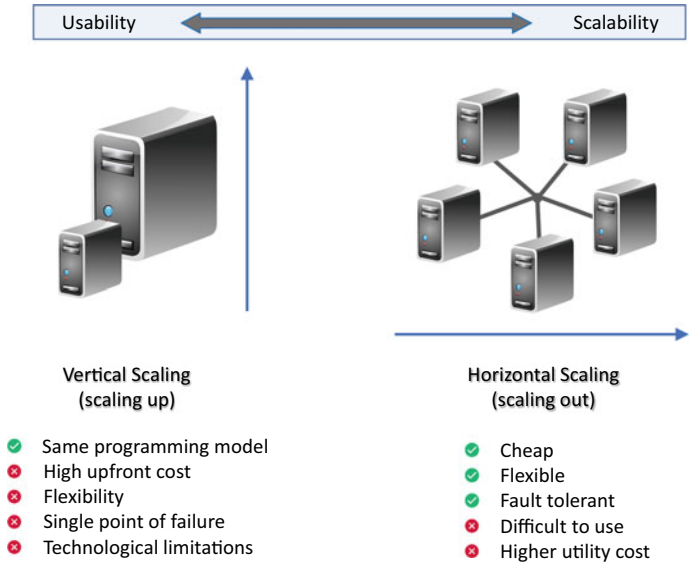


Fig. 5.2 Scaling Up versus Scaling Out

5.2.3 Generalisation

The generalizability of big data systems refers to operating with various data kinds, formats, and structures, which means that the system must be able to process multiple data types, including structured, semi-structured, and unstructured data, and get valuable insights from them. A big data system must be able to generalise to operate with various data kinds and formats and get valuable insights from them. In turn, this may assist businesses in making data-driven decisions and gaining a competitive edge.

A big data system needs the following qualities to accomplish generalisation:

- **Data integration:** Big data systems should have the capability of combining data from many sources, including databases, data warehouses, data lakes, and data streams, and make it possible to analyse that data.
- **Data processing:** A big data system should be able to process the structured, semi-structured, and unstructured data types that make up big data in different formats such as database, text, audio, video, and images.
- **Data analysis:** A big data system should perform several analyses, including statistical analysis, machine learning, NLP, and data mining.
- **Data visualisation:** The data must be represented in charts, graphs, or dashboards to help users understand the data more through big data systems.
- **Data management:** From data intake through data archiving, a big data system should be able to handle data and guarantee its correctness, consistency, and quality.

5.2.4 Extensibility

The extensibility of big data systems, which refers to the capacity to grow or include new features without present impairing operation, is a crucial and desired characteristic. As a result, the system must be adaptable enough to fit new data sources, processing techniques, storage innovations, and user interfaces as they become available. Big data systems must be extensible since they constantly change, and new data sources and technologies are continually being developed. Developers may guarantee that a system will remain valuable and efficient for years to come by building it with extensibility in mind.

The following are some essential factors that affect extensibility in a big data system:

- **Plug-in architecture:** Plug-in architecture is a modular design that is crucial for extensibility since it enables developers to add new components, features, or APIs to the system without disrupting existing ones; this means new components should be added or withdrawn quickly without modifying existing components in the system's architecture.
- **Standardised interfaces:** Standard interfaces provide for smooth communication between various components, allowing developers to add new features to the system without having to rewrite any current code.

5.2.5 Low Latency Reads and Updates

Low-latency reads and updates, which relate to the system's ability to read and update data quickly, are desired characteristics of a big data system. Reduced latency is crucial for big data systems because it allows for real time or almost real-time data processing, which is necessary for many applications, including online transactions, streaming analytics, and real-time monitoring. The ability to process data fast and effectively, which is crucial in many applications, may be ensured by developers by developing a system with low latency.

The following essential factors contribute to a big data system achieving low-latency reads and updates:

- **Distributed architecture:** Distributed architecture, which enables large-scale data processing in parallel across multiple nodes, provides low-latency reads and updates, resulting in the processing of large volumes of data quickly and efficiently.
- **Caching:** Caching is a technique for fast accessing frequently visited data by storing it in memory or on storage. Since it allows the system to rapidly retrieve often-used material without downloading it from the disk each time, caching is crucial for low-latency reads and updates.

- **Partitioning:** Data partitioning is a method that separates data into more manageable chunks and disperses it over several nodes. As a result, reading and updating data takes less time since the system can process data in parallel.
- **Optimization techniques:** Big data systems have optimization algorithms to provide high throughput and low latency, which allows data processing quickly and efficiently, such as processing data in memory rather than on disk.

5.2.6 Minimal Maintenance

The capacity of a big data system to require minimum continuous maintenance and upkeep is referred to as the system's desirable attribute of low maintenance. This implies that the system should function properly and efficiently without requiring ongoing oversight or assistance from IT employees. System developers can guarantee a system's dependability, effectiveness, and cost-effectiveness by building the system with low maintenance in mind.

These are a few key factors that help ensure low maintenance in a big data system:

- **Fault tolerance:** Failures should be handled graciously, and the system should automatically bounce back. This implies that the system should be able to recognise faults and automatically transition to a backup system or node.
- **Automatic backup service:** This service regularly backs up data without the need for manual intervention. This kind of service is frequently used to safeguard essential data and guarantee that it can be recovered in the case of a data loss or system malfunction. Cloud storage, external hard drives, and Network-attached Storage (NAS) devices are just a few locations where automatic backup services may be set to back up data. To ensure that data is safe and can be restored promptly, these services often employ several backup techniques, including complete backups and incremental backups.
- **Self-monitoring:** The monitoring itself provides to identify possible problems before they occur. Moreover, this helps determine the anomalies and notify IT personnel about them.

5.2.7 Debuggability

Big data systems' development, upkeep, and optimization are generally tricky, and they frequently need a high level of technical know-how to function well. Debuggability is key for a big data system to be dependable, effective, and simple to maintain over time. Debugging tools, robust logging and monitoring capabilities, and error-

handling methods are standard components of large data systems that help them be debuggable. These tools provide system activity monitoring, performance bottleneck detection, code execution tracking, code analysis, and code debugging for developers and IT employees.

In a big data system, the following important factors affect debuggability:

- **Monitoring:** A big data system should have a robust monitoring capability that users can use to track system activity and identify problems. All the actions and events, including the status of each task and infrastructure information of the big data cluster, should be centrally collected to track the system's health status and create notifications when issues arise.
- **Error handling:** Error handling or exception handling in big data systems helps ensure the system operates effectively and efficiently even when errors occur. Exceptions must be handled carefully since a plain run-time error from stale source data might terminate the entire procedure quickly. Data is altered before combining and matching with other data, often imported from various sources. Uncertainty regarding the nature of the data and the transformation algorithms—which the application coder frequently gives—causes the operation to end with an error while ingesting data from data sources.
- **Documentation:** The big data system must have comprehensive documentation that describes its architecture, design, and top development and maintenance methods and explains how the procedure operates and how to identify and resolve problems.

5.3 Big Data Processing Systems

Big data systems, designed to handle large volumes of data, enable organisations to extract insights from vast amounts of data that traditional methods cannot manage, which can be used to make informed decisions and support business growth [13]. Big data systems, typically involving distributed computing, where data is processed across multiple nodes in a cluster, enable the processing of data too large to fit in a single machine's memory in a reasonable amount of time. Big data systems, widely used in almost every field today, are generally used in data analytics, machine learning, and real-time processing applications.

Big data systems can be divided into four primary groups according to their intended use, as depicted in Fig. 5.3.

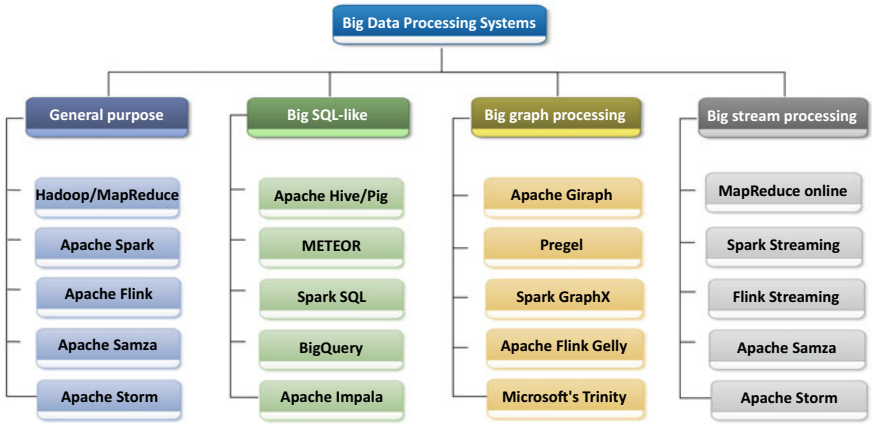


Fig. 5.3 Big data processing systems

5.4 Big Data Processing with Hadoop

Apache Hadoop was designed by Doug Cutting and Mike Cafarella in 2005 with inspiration from the Google File System and MapReduce programming model, to handle big data, typically characterised by its volume, value, variety, velocity, and veracity that conventional data processing tools and methods cannot handle. Hadoop, written in Java, is an open-source project under the Apache Hadoop license, which enables the processing of large-scale volumes of structured, semi-structured, and unstructured data in a parallel and distributed manner across clusters of thousands of computers [14]. Figure 5.4 depicts the Hadoop cluster architecture.

Hadoop has three crucial components, MapReduce, HDFS, and YARN. HDFS is the distributed data storage system where MapReduce processes the data on it. YARN is the resource management and job scheduling system that allocates resources for MapReduce to perform data processing. The details of these components are discussed in detail below:

5.4.1 MapReduce Paradigm

MapReduce [15] is a programming model developed by researchers from Google, written in Java to process large-scale data in parallel over a Hadoop cluster consisting of multiple machines.

Figure 5.5 depicts the working principle of the MapReduce paradigm. MapReduce has two essential functions; map and reduce. The map function creates mapper tasks that split big data into smaller chunks, each converted into key-value pairs. These mapper tasks are executed in worker nodes and produce key-value pairs to be stored in

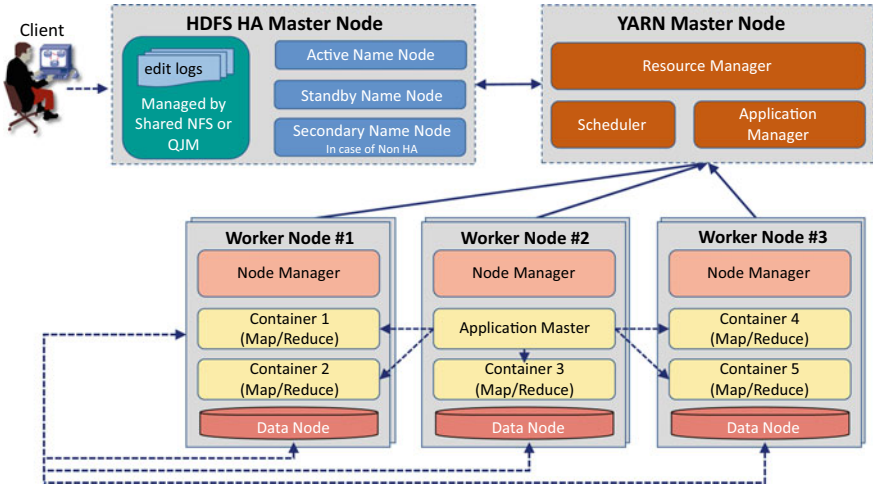


Fig. 5.4 The concept of Apache Hadoop architecture

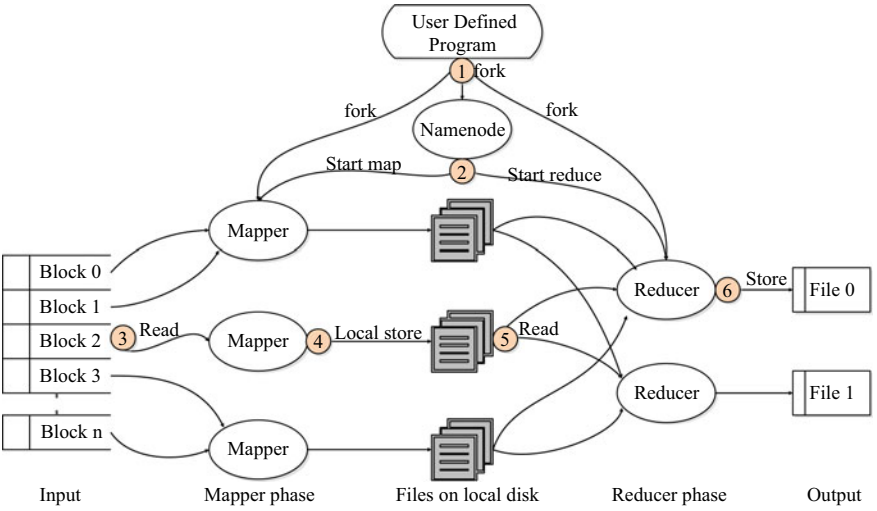


Fig. 5.5 MapReduce distributed programming model for big data

HDFS. After that, the reduce function is executed and creates reducer tasks consisting of three steps: shuffle, sort, and reduce. Shuffle groups the generated key-value pairs, which are then sorted in the sort phase. Shuffle and sort phases are executed simultaneously. Finally, the reducer gathers all the values belonging to the same key. Every reducer obtains all values associated with the same key and produces the final output.

Figure 5.6 simulates the MapReduce workflow of WordCount, which counts the number of occurrences of each word in a given text file. First, the dataset larger

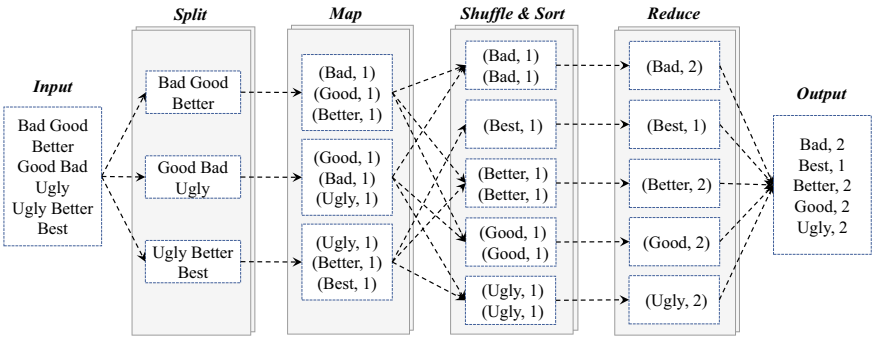


Fig. 5.6 MapReduce workflow of the WordCount application

than 128 MB is split into multiple blocks of 128 MB each by default Hadoop HDFS configuration. JobTracker service in the Hadoop framework creates mappers based on the number of data blocks. After processing the data blocks in *Map* phase, *Shuffle & Sort* receives the outputs to group and sort. Finally, *Reduce* combines all the results and writes the output into the HDFS.

5.4.2 Hadoop Distributed File System (HDFS)

HDFS is a distributed file system solution built to store and process large datasets on a cluster of commodity hardware [16]. It divides data into blocks and replicates them across multiple nodes in the cluster to ensure high throughput, reliability, and availability. HDFS provides a user-friendly interface which allows users to interact with it to monitor and track HDFS processes easily. The POSIX-compliant file system-based interface is accessible through the standard Linux commands or the APIs provided by Hadoop.

Figure 5.7 describes the architecture of HDFS. The namespace and metadata of all the files, including file permissions, ownership, and block locations, are maintained by the NameNode. It also selects the DataNodes that house the data blocks and responds to client requests for read-and-write processes on files. With the HDFS design, the NameNode is a single point of failure, but its metadata is frequently backed up to a backup NameNode to provide fault tolerance. The data blocks are stored in DataNodes to supply read-and-write requests from clients. DataNodes send heartbeats to the NameNode to inform the status of the blocks in a given period (three seconds by default). NameNode is responsible for sending instructions to DataNodes about managing the blocks, such as the location of blocks and replication factor.

5.4.3 Yet Another Resource Negotiator (YARN)

YARN, which comes with the Hadoop 2.0 distributions, is responsible for scheduling jobs and managing resources in the cluster to make it more versatile and efficient. It provides effective resource use and task scheduling across several worker nodes.

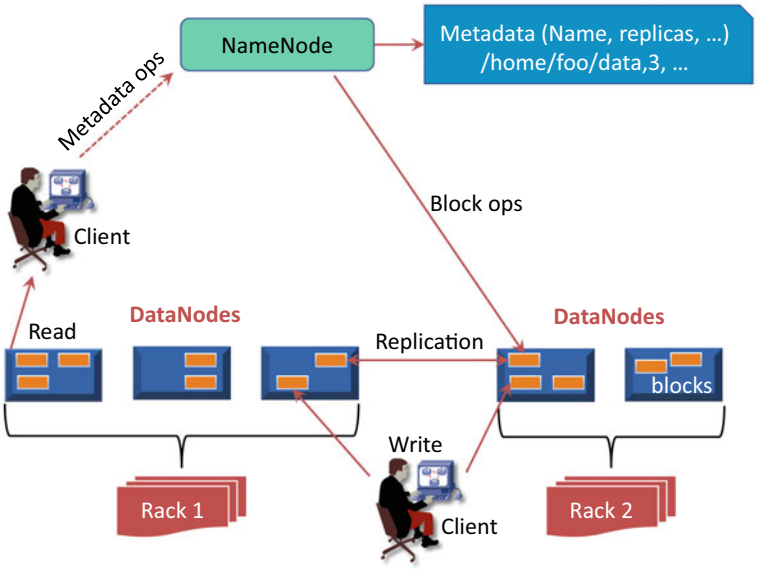


Fig. 5.7 HDFS architecture

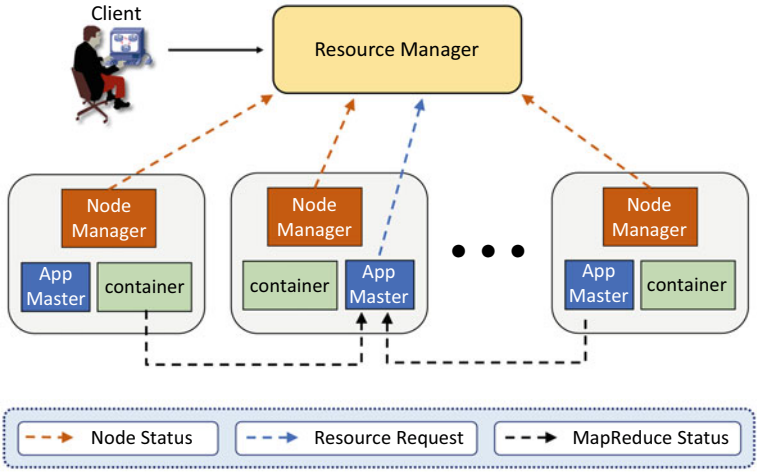


Fig. 5.8 YARN architecture and its components

The available resources in the Hadoop cluster are divided by YARN to allocate to the jobs if needed. YARN also provides a centralised management and monitoring system to track the progress of each specific task and health status. Figure 5.8 shows YARN's architecture and components.

YARN has three crucial components: ResourceManager, NodeManager, and ApplicationMaster. ResourceManager runs on the master node that receives the jobs submitted by clients and schedules them throughout the cluster. It is also responsible for allocating resources to the running jobs. NodeManager is executed in each worker node where MapReduce tasks, namely mappers, and reducers, run, which tracks resource utilisation of the node via ApplicationMaster and containers. The node manager reports the resource usage to the ResourceManager. The execution of jobs is managed by the application master that communicates with the ResourceManager using the YARN ApplicationMasterProtocol.

5.4.4 Installing Multi-node Hadoop Cluster

This section describes installing a multi-node Hadoop cluster of 1 master and two worker nodes on Ubuntu 20.04 on the local computer. To this end, we perform prerequisites before starting the installation.

5.4.4.1 Prerequisites

Creating a Network of Machines in VirtualBox with SSH Access: To create a Hadoop cluster, first, it is required to create a network of machines in VirtualBox with SSH access. After creating 3 Ubuntu-installed VM (the blue part needs to be changed for each VM);

- Open the hostname file and write *master* as a hostname:

```
$ sudo gedit /etc/hostname
master
```

- Change the hosts:

```
$ sudo gedit /etc/hosts
192.168.56.10 master
192.168.56.11 slave1
192.168.56.12 slave2
```

- Assign static IPs address for each VM:

```
$ sudo gedit /etc/network/interfaces
auto lo iface lo inet loopback
auto enp0s8
iface enp0s8 inet static
address 192.168.56.10
netmask 255.255.255.0
```

- Set up passwordless ssh access between nodes to create a Hadoop cluster (Steps to do on all nodes):

```
$ sudo apt-get update && sudo apt-get -y dist-upgrade
```

```
$ sudo apt-get install openjdk-8-jdk
$ sudo addgroup hadoop
$ sudo adduser --ingroup hadoop hduser
$ sudo adduser hduser sudo
$ sudo apt-get install ssh
$ sudo su - hduser
$ sudo ufw disable
$ ssh-keygen -t rsa -P "" -f ~/.ssh/id_rsa
$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorised_keys
$ chmod 700 ~/.ssh/id_rsa
$ ssh-copy-id -i /home/hduser/.ssh/id_rsa.pub hduser@192.168.56.10
$ ssh-copy-id -i /home/hduser/.ssh/id_rsa.pub hduser@192.168.56.11
$ ssh-copy-id -i /home/hduser/.ssh/id_rsa.pub hduser@192.168.56.12
```

After these steps, ping each other:

```
$ ping 192.168.56.11
```

and try if passwordless ssh is successful, such as:

```
$ ssh hduser@192.168.56.12 || $ ssh hduser@slave2
```

5.4.4.2 Downloading and Setting Values

- Download Hadoop 3.2.1 and extract the Hadoop binaries by the following commands.

```
$ wget https://archive.apache.org/dist/hadoop/core/hadoop-3.2.1/hadoop-3.2.1.tar.gz -P /Downloads
$ wget https://archive.apache.org/dist/hadoop/common/hadoop-3.2.1/hadoop-3.2.1.tar.gz -P /Downloads
$ sudo tar zxvf /Downloads/hadoop-* -C /usr/local
$ sudo mv /usr/local/hadoop-* /usr/local/hadoop
$ sudo chown -R hduser:hadoop /usr/local/hadoop
```

5.4.4.3 Setting Up a Multi-node Cluster

This section defines the steps for configuring a Multi-Node Cluster by updating the configuration files.

- Set static values:

```
$ sudo gedit /.bashrc
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export PATH=$PATH:$JAVA_HOME/bin
export HADOOP_HOME=/usr/local/hadoop
```

```
export PATH=$PATH:$HADOOP_HOME/bin
export HADOOP_CONF_DIR=/usr/local/hadoop/etc/hadoop
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
$ source  /.bashrc
```

- Update JAVA_HOME path:

```
$ cd $HADOOP_CONF_DIR
$ sudo gedit hadoop-env.sh
```

Replace this line:

```
export JAVA_HOME=$JAVA_HOME
```

with the following line:

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
```

- Update the core-site configuration file:

```
$sudo gedit core-site.xml
<configuration>
    <property>
        <name>fs.defaultFS</name>
        <value>hdfs://master:9000</value>
    </property>
</configuration>
```

- Create NameNode directory (perform these steps only in the master node):

```
$ sudo rm -rf /usr/local/hadoop/hdfs/namenodedata
$ sudo mkdir -p /usr/local/hadoop/hdfs/namenodedata
$ sudo chown -R hduser:hadoop /usr/local/hadoop/hdfs/namenodedata
$ sudo chmod 700 -R '/usr/local/hadoop/hdfs/namenodedata'
```

- Create DataNode directory (perform these steps only in the worker nodes):

```
$ sudo rm -rf /usr/local/hadoop/hdfs/datanodedata
$ sudo mkdir -p /usr/local/hadoop/hdfs/datanodedata
$ sudo chown -R hduser:hadoop /usr/local/hadoop/hdfs/datanodedata
$ sudo chmod 700 -R '/usr/local/hadoop/hdfs/datanodedata'
```

- Set up HDFS Properties (perform this only in master node):

```
$ sudo gedit hdfs-site.xml
<configuration>
    <property>
        <name>dfs.replication</name>
        <value>3</value>
    </property>
    <property>
```

```
        <name>dfs.namenode.name.dir</name>
        <value>file:///usr/local/hadoop/hdfs/namenodedata</value>
    </property>
    <property>
        <name>dfs.datanode.data.dir</name>
        <value>file:///usr/local/hadoop/hdfs/datanodedata</value>
    </property>
</configuration>
```

- Set up MapReduce Properties with JobHistoryServer (perform this only in master node):

```
$ sudo gedit mapred-site.xml
<configuration>
    <property>
        <name>mapreduce.jobtracker.address</name>
        <value>master:54311</value>
    </property>
    <property>
        <name>mapreduce.framework.name</name>
        <value>yarn</value>
    </property>
    <property>
        <name>mapreduce.jobhistory.address</name>
        <value>master:10020</value>
    </property>
    <property>
        <name>mapreduce.jobhistory.webapp.address</name>
        <value>master:19888</value>
    </property>
    <property>
        <name>yarn.app.mapreduce.am.env</name>
        <value>HADOOP_MAPRED_HOME=$HADOOP_HOME</value>
    </property>
    <property>
        <name>mapreduce.map.env</name>
        <value>HADOOP_MAPRED_HOME=$HADOOP_HOME</value>
    </property>
    <property>
        <name>mapreduce.map.speculative</name>
        <value>true</value>
    </property>
    <property>
        <name>mapreduce.reduce.speculative</name>
        <value>false</value>
    </property>
```

```
    <property>
        <name>mapreduce.reduce.env</name>
        <value>HADOOP_MAPRED_HOME=$HADOOP_HOME</value>
    </property>
    <property>
        <name>mapreduce.job.reduce.slowstart.completedmaps</name>
        <value>0.99</value>
    </property>
</configuration>
```

- Set up YARN Properties:

```
$ sudo gedit yarn-site.xml
<configuration>
    <property>
        <name>yarn.nodemanager.aux-services</name>
        <value>mapreduce_shuffle</value>
    </property>
    <property>
        <name>yarn.resourcemanager.hostname</name>
        <value>master</value>
    </property>
    <property>
        <name>yarn.resourcemanager.resource-tracker.address</name>
        <value>master:8025</value>
    </property>
    <property>
        <name>yarn.resourcemanager.scheduler.address</name>
        <value>master:8030</value>
    </property>
    <property>
        <name>yarn.resourcemanager.address</name>
        <value>master:8050</value>
    </property>
    <property>
        <name>yarn.log-aggregation-enable</name>
        <value>true</value>
    </property>
</configuration>
```

- Set up Master and Slave (worker) nodes (perform this only in master node):

```
$ cd $HADOOP_CONF_DIR
$ sudo cp workers masters
$ sudo gedit masters
master
```

```
$ sudo gedit workers
slave1
slave2
```

- Configure DataNodes' HDFS properties (perform this only in worker nodes):

```
sudo gedit hdfs-site.xml
<configuration>
    <property>
        <name>dfs.replication</name>
        <value>3</value>
    </property>
    <property>
        <name>dfs.datanode.data.dir</name>
         <value>file:///usr/local/hadoop/hdfs/datanodedata</value>
    </property>
</configuration>
```

5.4.4.4 Starting the Cluster

These codes are executed in the master node:

- First, we may disable the Uncomplicated Firewall (UFW) on a Linux system.

```
$ sudo ufw disable
```

- Start the cluster, YARN, and JobHistoryServer.

```
$HADOOP_HOME/bin/hdfs namenode -format
$HADOOP_HOME/sbin/start-dfs.sh
$HADOOP_HOME/sbin/start-yarn.sh
$HADOOP_HOME/sbin/mr-jobhistory-daemon.sh start historyserver
$ hdfs dfsadmin -safemode leave
```

- Create the user folder in HDFS:

```
$hadoop fs -mkdir /user
$hadoop fs -mkdir /user/hduser
```

- Copy a txt file from local to HDFS:

```
$ hadoop fs -mkdir small
$ hadoop fs -mkdir outputs
$ hadoop fs -copyFromLocal '/home/umit/Desktop/small.txt' outputs
```

- Check the status of the cluster:

```
$ hdfs dfsadmin -report
```

```
hduser@master:~$ hdfs dfsadmin -report
Configured Capacity: 63140864000 (58.80 GB)
Present Capacity: 38165925888 (35.54 GB)
DFS Remaining: 36685365248 (34.17 GB)
DFS Used: 1480560640 (1.38 GB)
DFS Used%: 3.88%
Replicated Blocks:
        Under replicated blocks: 60
        Blocks with corrupt replicas: 0
        Missing blocks: 0
        Missing blocks (with replication factor 1): 0
        Low redundancy blocks with highest priority to recover: 0
        Pending deletion blocks: 0
Erasure Coded Block Groups:
        Low redundancy block groups: 0
        Block groups with corrupt internal blocks: 0
        Missing block groups: 0
        Low redundancy blocks with highest priority to recover: 0
        Pending deletion blocks: 0

-------------------------------------------------
Live datanodes (2):

Name: 192.168.56.11:9866 (slave1)
Hostname: slave1
Decommission Status : Normal
Configured Capacity: 31570432000 (29.40 GB)
DFS Used: 740294656 (706 MB)
Non DFS Used: 10851708928 (10.11 GB)
DFS Remaining: 18351144960 (17.09 GB)
DFS Used%: 2.34%
DFS Remaining%: 58.13%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 1
Last contact: Sat Mar 11 06:34:17 GMT 2023
Last Block Report: Sat Mar 11 06:33:38 GMT 2023
Num of Blocks: 60


Name: 192.168.56.12:9866 (slave2)
Hostname: slave2
Decommission Status : Normal
Configured Capacity: 31570432000 (29.40 GB)
DFS Used: 740265984 (705.97 MB)
Non DFS Used: 10868662272 (10.12 GB)
DFS Remaining: 18334220288 (17.08 GB)
DFS Used%: 2.34%
DFS Remaining%: 58.07%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 1
Last contact: Sat Mar 11 06:34:16 GMT 2023
Last Block Report: Sat Mar 11 06:33:40 GMT 2023
Num of Blocks: 60
```

Fig. 5.9 The current state of the Hadoop cluster

Figure 5.9 shows the screenshot of the report of Hadoop cluster. Apache Hadoop provides APIs which enable access to the cluster via a web browser. Figure 5.10 shows the screenshot of the report of the Hadoop cluster. Moreover, Fig. 5.11 provides the information of data nodes via the 9870 port on the browser.

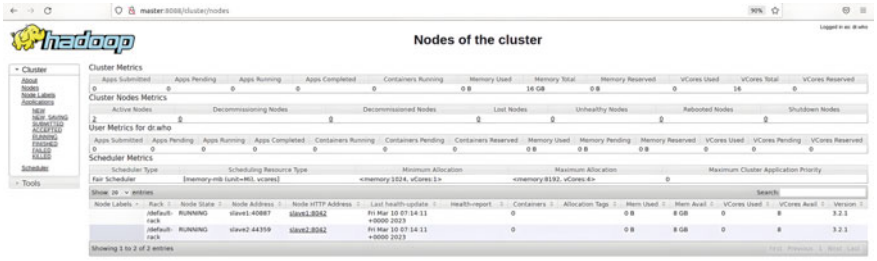


Fig. 5.10 The information of the nodes of the cluster

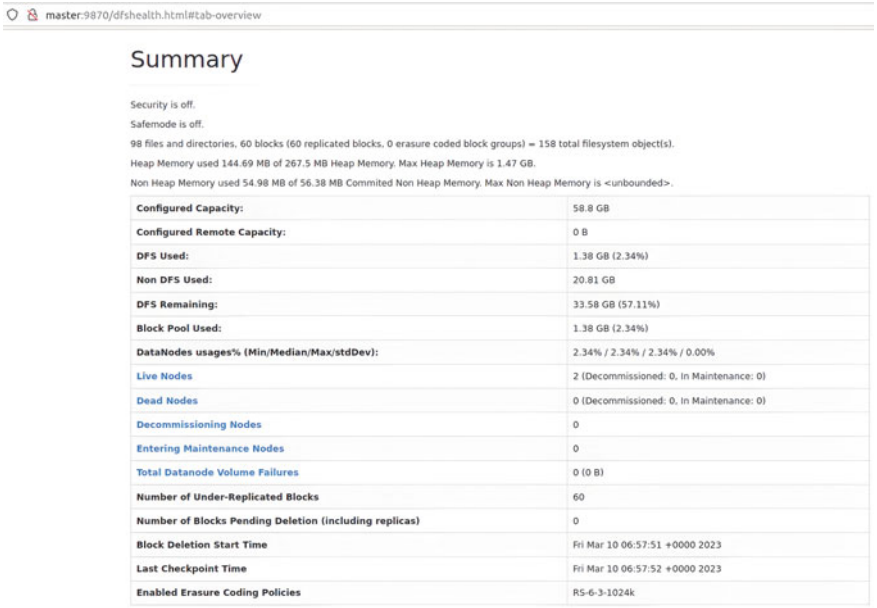


Fig. 5.11 The summary of data nodes

- Stop the cluster:
  `$HADOOP_HOME/sbin/stop-dfs.sh`
  `$HADOOP_HOME/sbin/stop-yarn.sh`
  `$HADOOP_HOME/sbin/mr-jobhistory-daemon.sh stop historyserver`

5.5 Apache Spark for Big Data Processing

Apache Spark[7] is an open-source unified distributed computing engine designed to handle massive amounts of data in parallel over a cluster of computers. It was developed by researchers from the University of California, Berkeley, in 2009 and was donated to the Apache Software Foundation in 2013.

Apache Spark is one of the top-level Apache projects that supports multi-language, including Java, Scala, Python, and R. It was developed to overcome the drawbacks of MapReduce by processing in memory, minimising the number of steps in a task, and reusing data across several concurrent processes. Unlike Hadoop, Spark does all data processing in memory, resulting in a much faster application. Moreover, Spark reuses data by employing an in-memory cache to significantly speed up ML algorithms that repeatedly run a function on the same dataset. The process of reusing data involves the construction of DataFrames, an abstraction over Resilient Distributed Datasets (RDD), a set of objects stored in memory and utilised in several Spark operations, which significantly reduces latency. The flexibility of Apache Spark allows it to handle a variety of workloads, including interactive searches, real-time analytics, ML, and graph processing. Several tasks may be smoothly combined into one application.

5.5.1 Apache Spark Core

Spark Core is responsible for managing memory, resolving faults, scheduling, distributing, monitoring activities, and communicating with storage systems. Different APIs are designed for Java, Scala, Python, and R and are used to access Spark Core, concealing the complexities of distributed processing for high-level operators. Figure 5.12 shows the core libraries and supported programming language for Apache Spark.

5.5.1.1 MLlib for Machine Learning

A library of techniques for performing machine learning at scale on data is included in Spark as MLlib. Data scientists may train Machine Learning models using R or Python on any Hadoop data source, save them using MLlib, and then import them into a Java- or Scala-based workflow. Machine learning can be completed rapidly because of Spark's fast, interactive computation in memory. The algorithms may do pattern mining, classification, regression, clustering, and collaborative filtering.

[7] https://spark.apache.org/.

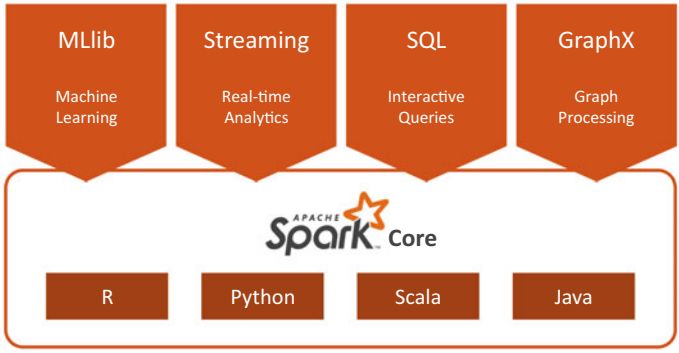


Fig. 5.12 The core libraries of Apache Spark

5.5.1.2 Spark Streaming for Real-Time Data Processing

Spark Streaming is used for real-time streaming analytics that uses Spark Core's quick scheduling ability. It uses the same application code created for batch analytics to analyse the data as it is ingested in mini-batches. Since they may utilise the same code for batch processing and real-time streaming apps, this increases developer productivity. Data from Twitter, Kafka, Flume, HDFS, ZeroMQ, and many more sources present in the Spark Packages ecosystem are supported by Spark Streaming.

5.5.1.3 Spark SQL for Interactive Queries

Spark SQL is a distributed query engine that offers interactive low-latency queries up to 100 times quicker than MapReduce. Business analysts can use Hive Query Language or conventional SQL to query data. Scala, Java, Python, and R all have APIs available to developers. Java Database connectivity (JDBC), Microsoft Open Database Connectivity (ODBC), JavaScript Object Notation (JSON), HDFS, Hive, Optimised Row Columnar (ORC), and Parquet are just a few data sources it supports.

5.5.1.4 GraphX for Graph Processing

Apache Spark has a distributed graph processing component called GraphX that allows users to interactively create and alter a graph data structure at scale using GraphX's ETL. It comes with various distributed Graph algorithms and a very versatile API.

5.5.2 Deploying Spark on YARN

This section explains how to install Apache Spark on a Hadoop YARN cluster.

5.5.2.1 Prerequisites

Before deploying Spark on the YARN cluster, it is required to install Scala on each worker node, including the master node.

```
$ sudo apt-get install scala
```

Check if the installation is successful.

```
$ scala-version
```

5.5.2.2 Installation

After installing Scala on each node, we only need to perform the installation of Spark on the master node only by executing the following command lines:

```
$ wget https://archive.apache.org/dist/spark/spark-3.2.1/spark-3.2.1-bin-hadoop3.2.tgz
$ tar -xzf spark-3.2.1-bin-hadoop3.2.tgz
$ sudo mv spark-3.2.1-bin-hadoop3.2 /usr/local/spark
$ sudo gedit  /.bashrc
export SPARK_HOME=/usr/local/spark/
export PATH=$PATH:$SPARK_HOME/bin
export LD_LIBRARY_PATH=$HADOOP_HOME/lib/native:$LD_LIBRARY_PATH
$ source  /.bashrc
```

5.5.2.3 Integrate Spark with YARN

Apache Spark should be configured with the YARN to ensure communication via the *HADOOP_CONF_DIR* environment variable.

```
$ mv $SPARK_HOME/conf/spark-defaults.conf.template $SPARK_HOME/conf/spark-defaults.conf
$ sudo gedit $SPARK_HOME/conf/spark-defaults.conf
spark.master yarn
spark.driver.memory 512m
spark.yarn.am.memory 512m
spark.executor.memory 512m
```

Set up the Spark history server:

```
$ sudo gedit $SPARK_HOME/conf/spark-defaults.conf
spark.eventLog.enabled true
spark.eventLog.dir hdfs://master:9000/spark-logs
spark.history.provider org.apache.spark.deploy.history.FsHistoryProvider
spark.history.fs.logDirectory hdfs://master:9000/spark-logs
spark.history.fs.update.interval 3s
spark.history.ui.port 18080
```

Create the log directory in HDFS:

```
$ hadoop fs -mkdir /spark-logs
```

Apache Spark is now ready to interact with the YARN cluster.

Execute the following code to start Spark with the Hadoop YARN cluster:

```
$ $HADOOP_HOME/sbin/start-dfs.sh
$ $HADOOP_HOME/sbin/start-yarn.sh
$ $HADOOP_HOME/sbin/mr-jobhistory-daemon.sh start historyserver
$ $ hdfs dfsadmin -safemode leave
```

The Spark History Server is now accessible by navigating to http://master:18080/ in a web browser.

5.5.3 Case Study

Let's execute the WordCount application on a txt file using PySpark, a Python API for Apache Spark, in Jupyter notebook.

```
from pyspark import SparkConf, SparkContext
conf = SparkConf().setAppName("Word Count")
# create a SparkContext object if it does not exist
try:
    sc = SparkContext(conf=conf)
except:
   pass
# Load the input text file from HDFS
input = sc.textFile("hdfs://master:9000/user/hduser/small/small
               .txt")
# Split the text into individual words and count their occurrences
wordCounts = input.flatMap(lambda line: line.split(" ")) \
               .map(lambda word: (word.lower(), 1)) \
               .reduceByKey(lambda a, b: a + b)
# Save the word count results to HDFS
wordCounts.saveAsTextFile("hdfs://master:9000/user/hduser/outputs
               /2")
# Stop the SparkContext
sc.stop()
```

Figure 5.13 is the screenshot from the UI showing the information regarding the executed job, such as event timeline, scheduler type, and executer processes.

Fig. 5.13 The history of the jobs in the user interface

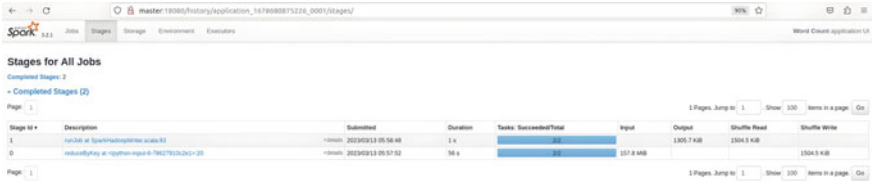


Fig. 5.14 The stages of the jobs

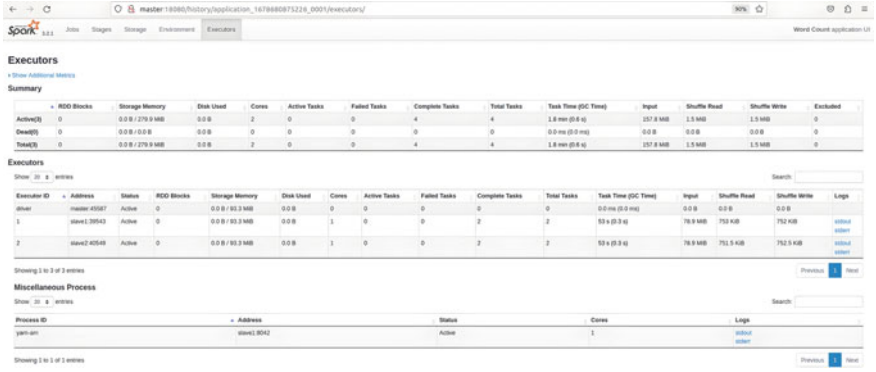


Fig. 5.15 The executors of the jobs

The job stages shown in Fig. 5.14 provide information about the jobs, including job submission date and time, process duration, whether the job is successful, and data size for input and output.

Figure 5.15 provides specific information about the executors. This information is crucial for debugging the Spark application.

5.6 Apache Hive for Data Engineering

Apache Hive is a fault-tolerant data warehouse system and SQL-like query language developed by Facebook. It is built on top of Apache Hadoop and enables users to query massive datasets stored in Hadoop using a syntax similar to SQL.

Hive essentially offers an interface similar to SQL for accessing data held in Hadoop's HDFS distributed file system or other compatible storage systems. Users can create queries and interact with data using a language called HiveQL, which is a variation of SQL. Hive converts these queries into MapReduce tasks carried out on the Hadoop cluster. Structured data in CSV or TSV files, Apache Avro data, JSON data, and binary formats like Parquet and ORC are all supported by Hive. Additionally, it has functions like bucketing and partitioning that make it possible to process and query massive datasets more effectively.

Hive allows users to work with big data using familiar SQL syntax, which lowers the barrier to entry for analysts and data scientists who may need to gain expertise in distributed systems or programming. Hive also integrates with other tools in the Hadoop ecosystem, such as Pig for data transformation and Spark for data processing.

5.6.1 Deploying Hive on YARN

The installation and integration of Hive on a Hadoop YARN cluster are performed in the following steps:

5.6.2 Installation

Like Spark, Hive installation must be performed on only the master node.

```
$ wget https://archive.apache.org/dist/hive/hive-3.1.1/apache-hive-3.1.1-bin.tar.gz -P  /Downloads
$ sudo tar zxvf  /Downloads/apache-hive-* -C /usr/local
$ sudo mv /usr/local/apache-hive-* /usr/local/hive
$ sudo chown -R hduser:hadoop /usr/local/hive
$ sudo gedit  /.bashrc
export HIVE_HOME=/usr/local/hive
export HIVE_CONF_DIR=/usr/local/hive/conf
export PATH=$HIVE_HOME/bin:$PATH
export CLASSPATH=$CLASSPATH:/usr/local/hadoop/lib/*:.
export CLASSPATH=$CLASSPATH:/usr/local/hive/lib/*:.
$ source  /.bashrc
```

5.6.3 Integration of Hive with Hadoop YARN

Apache Hive must be configured with the YARN to ensure communication via the HADOOP_CONF_DIR environment variable.

```
$ cd $HIVE_CONF_DIR
$ sudo cp hive-env.sh.template hive-env.sh
$ sudo gedit hive-env.sh
export HADOOP_HOME=/usr/local/hadoop
export HIVE_HOME=/usr/local/hive
```

Creating Hive warehouse directory:

```
$ hadoop fs -mkdir /user/hive/
$ hadoop fs -mkdir /user/hive/warehouse
$ hadoop fs -chmod g+w /tmp
$ hadoop fs -chmod g+w /user/hive/warehouse
```

Configuration:

```
$ cd $HIVE_CONF_DIR
$ sudo cp hive-default.xml.template hive-site.xml
$ sudo gedit hive-site.xml
<configuration>
    <property>
        <name>system:java.io.tmpdir</name>
        <value>/tmp/hive/java</value>
    </property>
    <property>
        <name>system:user.name</name>
        <value>hduser</value>
    </property>
</configuration>
$ sudo reboot
```

The installation of Hive has been successful, as seen in Fig. 5.16. For the configuration of Metastore, we now require an external database server. To this end, the Apache Derby database is employed.

```
$ wget http://archive.apache.org/dist/db/derby/db-derby-10.13.1.1/db-derby-10.13.1.1-bin.tar.gz
$ sudo tar xvzf db-derby-10.13.1.1-bin.tar.gz -C /usr/local
$ sudo gedit  /.bashrc
export DERBY_HOME=/usr/local/db-derby-10.13.1.1-bin
export PATH=$PATH:$DERBY_HOME/bin
export CLASSPATH=$CLASSPATH:$DERBY_HOME/lib/derby.jar:$DERBY_HOME/lib/derbytools.jar
```

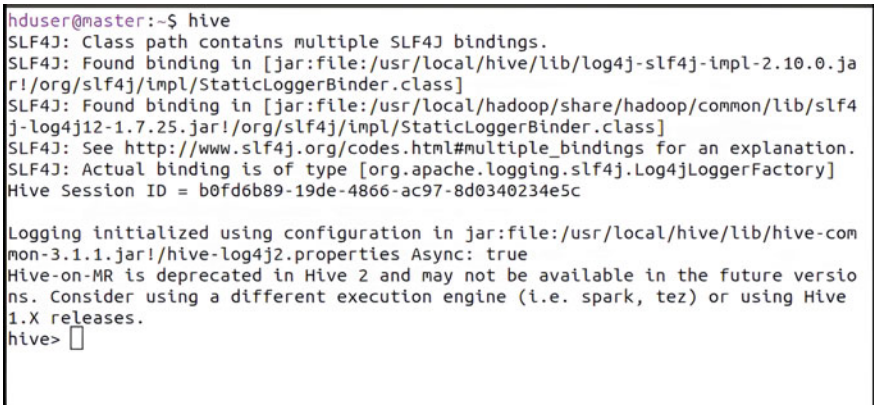


Fig. 5.16 Successfully installation of Apache Hive

```
source  /.bashrc
```

Now, we will create a directory called data to store metastore data in $DERBY_HOME directory.

```
$ sudo mkdir $DERBY_HOME/data
```

Create jpox.properties in /usr/local/hive/conf/ and update it using the codes below:

```
$ cat > jpox.properties
javax.jdo.PersistenceManagerFactoryClass =
org.jpox.PersistenceManagerFactoryImpl
org.jpox.autoCreateSchema = false
org.jpox.validateTables = false
org.jpox.validateColumns = false
org.jpox.validateConstraints = false
org.jpox.storeManagerType = rdbms
org.jpox.autoCreateSchema = true
org.jpox.autoStartMechanismMode = checked
org.jpox.transactionIsolation = read_committed
javax.jdo.option.DetachAllOnCommit = true
javax.jdo.option.NontransactionalRead = true
javax.jdo.option.ConnectionDriverName = org.apache.derby.jdbc.
ClientDriver
javax.jdo.option.ConnectionURL = jdbc:derby://master:1527/metastore
z_db;create = true
javax.jdo.option.ConnectionUserName = APP
javax.jdo.option.ConnectionPassword = mine
```

then Ctrl + D → to exit

Metastore schema initialisation:

As an initialisation stage, we must run the *schematool* command shown below when we start Hive. We utilise derby as a database type in our case:

Go to → bin directory of Hive and run the code below:

```
$ cd /usr/local/hive/bin
$ schematool -dbType derby -initSchema
```

Running Hive:

```
$ cd /usr/local/hive/bin
$ hive
```

Now, Hive is ready to perform queries.

5.6.4 Case Study

To perform a Hive application, we will use a small dataset, *employee.csv*,[8] from Kaggle. Let's say we have a dataset with details of employees, such as gender, salaries, bonus, and team. We would like to see the average salary for each team and store the results in a table.

To this end, first, we need to create a Hive table to store the employee dataset using the following HiveQL code. First, we need to copy the file into the directory called *data* in HDFS.

```
$ hadoop fs -copyFromLocal employee.csv data/
```

Now, we will create the table:

```
CREATE TABLE employees (
   name STRING,
   gender STRING,
   startDate TIMESTAMP,
   lastLoginTime TIMESTAMP,
   salary INT,
   bonus DOUBLE,
   seniorManagement BOOLEAN,
   team STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;
```

[8] https://www.kaggle.com/datasets/prakharjadaun/employee-data.

Next, we can load our data into the employees table:

```
LOAD DATA LOCAL INPATH 'data/employees.csv' INTO TABLE employees;
```

Now that we have our data in Hive, we can use the following query to calculate the average salary for each department:

```
CREATE TABLE team_avg_salary AS
SELECT team, AVG(salary) AS avg_salary
FROM employees
GROUP BY team;
```

This query creates a new table called team_avg_salary that contains the average salary for each team.

Finally, we can view the results by running the following query:

```
SELECT * FROM team_avg_salary;
```

5.7 Apache Sqoop for Data Ingestion

Apache Sqoop[9] is used to move data from relational systems or data centres to Hadoop, allowing users to move large-scale data from relational databases (i.e. MySQL, Oracle, and SQL Server) to data warehouses and data warehouses (i.e. HDFS). It was developed by the Apache Software Foundation and released as an open-source project.

The key benefits of Apache Sqoop are listed below:

- Simplifying the process of moving sizable information from relational systems to Hadoop.
- Allowing users to only transfer new or modified data by supporting incremental imports.
- Providing a GUI, a Java API, and a command-line interface that is simple to use.
- Enabling users to modify the transfer procedure to suit their unique needs.

Users can transfer or export data between Hadoop and relational databases using the instructions provided by the Sqoop Command-line Interface (CLI), a Java API. It also provides a Graphical User Interface (GUI) to simplify the transfer. The instructions are run in the Sqoop-installed machine's CLI or terminal.

The following subsections explain the installation and configuration of Apache Sqoop on Hadoop.

[9] https://sqoop.apache.org/.

5.7.1 Installation

```
$ wget https://archive.apache.org/dist/sqoop/1.4.7/sqoop-1.4.7.bin__hadoop-2.6.0.tar.gz -P  /Downloads
  $ sudo tar zxvf  /Downloads/sqoop-* -C /usr/local
  $ sudo mv /usr/local/sqoop-* /usr/local/sqoop
  $ sudo chown -R hduser:hadoop /usr/local/sqoop
  $ sudo gedit  /.bashrc
  export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
  export SQOOP_HOME=/usr/local/sqoop/
  export PATH=$PATH:$SQOOP_HOME/bin
  export CLASSPATH=$CLASSPATH:$SQOOP_HOME/lib/*:.

  $ source  /.bashrc
```

5.7.2 Configuration of Apache Sqoop

```
$ cd $SQOOP_HOME/conf
  $ mv sqoop-env-template.sh sqoop-env.sh
  $ sudo gedit sqoop-env.sh
  export HADOOP_COMMON_HOME=/usr/local/hadoop
  export HADOOP_MAPRED_HOME=/usr/local/hadoop
```

We need a driver/connector jar to enable the connection to the database. We will use the MySQL database in this context. To this end, we will download the MySQL connector jar, mysql-connector-java-5.1.36.tar.gz and extract it.

```
$ https://downloads.mysql.com/archives/c-j/
```

We need to copy this jar and put in /usr/local/sqoop/lib directory.

```
  $ cd /home/umit/Downloads/
  $ sudo cp '/home/umit/Downloads/mysql-connector-java-5.1.46-bin.jar' /usr/local/sqoop/lib
```

Check the Sqoop version.

```
  $ sqoop version
```

Before starting the case study, we will install MySQL on our Ubuntu-installed machine to transfer data from a relational database to HDFS.

To this end, first, we will perform the following codes in the master node under the user of *hduser*.

```
  $ sudo apt-get update
  $ sudo apt-get install mysql-server
  $ sudo ufw allow MySQL
  $ sudo systemctl start MySQL
```

```
$ sudo systemctl enable mysql
```

Create a new MySQL user:

```
$ CREATE USER 'hiveuser'@'localhost' IDENTIFIED BY 'hiveuser';
$ GRANT ALL PRIVILEGES ON * . * TO 'hiveuser'@'localhost';
$ FLUSH PRIVILEGES;
```

Starting mysql:

```
$ mysql -u hduser -p
```

MySQL is now ready to go.

Before starting to perform a case study, it is good to know the options for importing data into Hive from an external relational database. Table 5.1 presents the Sqoop commands with their description.

Table 5.1 Metrics of the job object

Sqoop command	Description
-hive-home <directory>	It overrides $HIVE_HOME
-hive-import	If no delimiters are specifically given, it imports tables into Hive using those set by default
-hive-overwrite	It overrides the Hive table's current data
-create-hive-table	During the process, a hive table is created. The operation will fail if this option is selected and the Hive table exists
-hive-table <table_name>	The table name to utilise when importing data into Hive is specified
-hive-drop-import-delims	When importing data into Hive, the delimiters \ n, \ r, and \ 01 are removed from string fields
-hive-delims-replacement	When importing data into Hive, the delimiters \ n, \ r, and \ 01 are replaced with a user-defined string from string fields
-hive-partition-key	It specifies the name of the Hive field used to split a sharded database
-hive-partition-value < value >	A string representing the partition key for Hive-imported data
-map-column-hive < map >	It overrides the defined columns' default SQL type to Hive type mapping

5.7.3 Case Study

Let's transfer the EMPLOYEES table consisting of four features, *"employeeId, firstName, lastName, jobTitle"* to Hive via Sqoop by performing the following Python code below:

```
import sqoop

def import_data(mysql_connection_string, mysql_username,\\
  mysql_password, hive_server2_uri, hive_database,\\
  hive_table, table_name, columns, delimiter,\\
   batch_size, num_threads):

  sqoop = sqoop.Sqoop()
  sqoop.set_connect_string(mysql_connection_string)
  sqoop.set_username(mysql_username)
  sqoop.set_password(mysql_password)

  sqoop.set_hive_server2_uri(hive_server2_uri)
  sqoop.set_hive_database(hive_database)
  sqoop.set_hive_table(hive_table)

  sqoop.set_table(table_name)
  sqoop.set_columns(columns)
  sqoop.set_delimiter(delimiter)
  sqoop.set_batch_size(batch_size)
  sqoop.set_num_threads(num_threads)

  sqoop.run()

if __name__ == "__main__":
  mysql_connection_string = "jdbc:mysql://localhost/mydatabase"
  mysql_username = "hduser"
  mysql_password = "123456"
  hive_server2_uri = "jdbc:hive2://localhost:10000/default"
  hive_database = "mySqlDatabase"
  hive_table = "EMPLOYEES"
  table_name = "EMPLOYEES"
  columns = ["employeeId", "firstName", "lastName", "jobTitle"]
  delimiter = ";"
  batch_size = 1000
  num_threads = 4

  import_data(mysql_connection_string, mysql_username, \\
     mysql_password, hive_server2_uri, hive_database, \\
     hive_table, table_name, columns, \\
     delimiter, batch_size, num_threads)
```

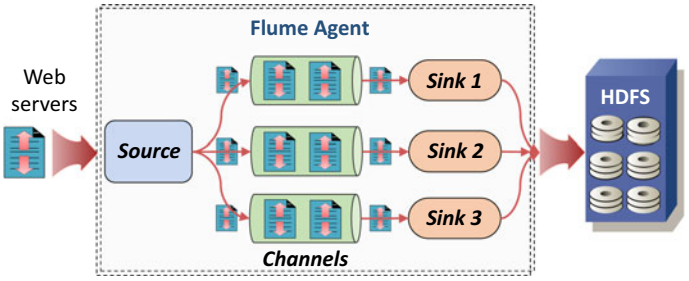


Fig. 5.17 The high-level architecture of Apache Flume

This code transfers the "EMPLOYEES" table in the MySQL database to the Hive table through Apache Sqoop. All the necessary information, such as database path, user name and password, the name of the database in Hive, the name of the SQL database, the Hive table and the columns, and the file's delimiter, is specified in the code.

5.8 Streaming Data Ingestion with Apache Flume

Apache Flume[10] is an open-source data ingestion technology that effectively gathers, transmits, and loads significant amounts of data from several sources to numerous destinations. Scalable, fault-tolerant, and highly available are all characteristics of Flume, which was created to manage large volumes of data that may be organised or unstructured. Figure 5.17 shows the architecture of Apache Flume and the logic behind the data transfer from a web server to HDFS.

Sources, channels, and sinks are the three fundamental building blocks of the Flume architecture. Several sources, including logs, social media feeds, and data sensors, are used to acquire the data [17]. Channels act as intermediate buffers between processing and sending the data to its destination. Sinks are in charge of transferring the data to the desired location, which may be Kafka, HDFS, or HBase in the case of Hadoop. The configuration of sources, channels, and sinks may be done according to the needs for data input. Other sources for Flume include file, syslog, exec, and HTTP. There are channels for memory, files, and JDBC. Hadoop HDFS, HBase, Solr, and other systems serve as sinks. Additionally, Flume has integrated interceptors that may alter or filter data as it travels through the Flume pipeline.

Due to Flume's high level of extensibility, it is possible to combine it with other big data technologies like Apache Kafka, Apache Spark, and Apache NiFi. Flume is also incredibly flexible, enabling users to increase its capabilities by adding unique sources, channels, and sinks.

[10] https://flume.apache.org/.

The next sections explain each step of the flume installation and configuration in detail for transferring data into the HDFS.

5.8.1 Installation

Like other big data frameworks, first, we will download Apache Flume software from the official website, https://flume.apache.org/.

```
$ wget https://dlcdn.apache.org/flume/1.11.0/apache-flume-1.11.0-bin.tar.gz
-P /Downloads
$ sudo tar zxvf /Downloads/apache-flume-* -C /usr/local
$ sudo mv /usr/local/apache-flume-* /usr/local/flume
$ sudo chown -R hduser:hadoop /usr/local/flume
$ sudo gedit /.bashrc
export FLUME_HOME=/usr/local/flume/
export PATH=$PATH:$FLUME_HOME/bin
$ source /.bashrc
```

5.8.2 Configuration of Apache Flume and Case Study

Unlike the other platforms, we will consider the configuration and case study parts together since the configuration in Apache Flume applications is done according to the case study. We will perform real-time Twitter data ingestion using Flume as a case study.

Now, we need to create a file called *flume.conf* and open it to add the parameters shown in Fig. 5.18.

```
$ touch flume.conf
$ sudo gedit flume.conf
```

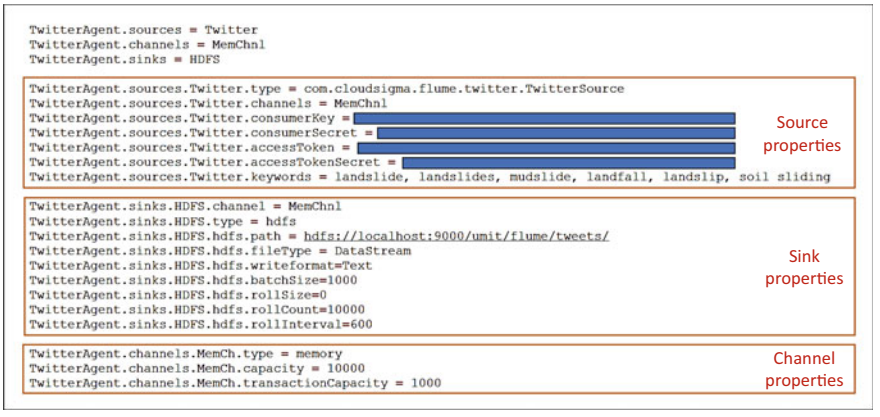


Fig. 5.18 Apache Flume configuration

After adding the parameters in the figure, we must execute the following command to start the streaming data ingestion.

```
./bin/flume-ng agent /
-f TwitterStream.properties /
-- name TwitterAgent /
-- conf $FLUME_HOME/conf /
-Dflume.root.logger=INFO, console
```

5.9 Apache Mahout: Distributed Machine Learning for Big Data Analytics

Apache Mahout[11] is an open-source project offering modules and algorithms for scalable machine learning and data mining. It is built to work on top of Apache Hadoop and Apache Spark, allowing it to process big datasets in a distributed and parallel manner.

Mahout provides a variety of machine learning methods, such as clustering, classification, and filtering. These algorithms employ common techniques such as k-means clustering, Naive Bayes classification, and Singular Value Decomposition (SVD) for collaborative filtering. The capacity to handle large-scale datasets is one of Mahout's key capabilities. Scalable approaches can manage datasets too large to fit into memory on a single system. Mahout achieves this scalability through the use of Hadoop's distributed computing capabilities. Data pretreatment methods like feature extraction, normalisation, and dimensionality reduction are also available from Mahout. These technologies aid in the preparation of data for machine learning algorithms (Fig. 5.19).

The key benefits of Apache Mahout are listed below in detail:

- **Scalability:** Mahout is designed to handle massive datasets in a distributed and parallel way, making it simple to scale up to manage big data.
- **Various algorithms:** Mahout offers a variety of machine learning techniques, such as clustering, classification, and collaborative filtering.
- **Customizability:** Mahout is highly adjustable, implying that customers may tweak and expand the algorithms to meet their needs.
- **Ease of use:** Mahout provides a straightforward and clear API for dealing with machine learning algorithms, making it simple to get started with the platform.
- **Community support:** Mahout has a vast and active community of developers and users who help and contribute to the project's growth.

[11] https://mahout.apache.org/.

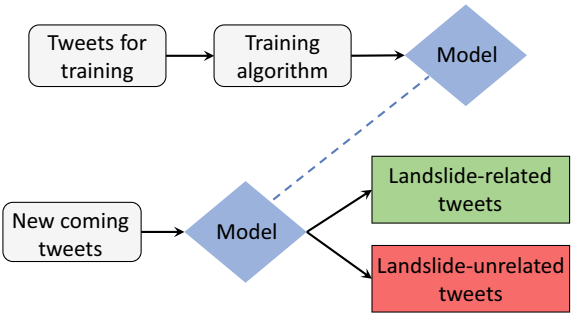


Fig. 5.19 Implementation an ML model from Twitter data using Apache Mahout

5.9.1 Installation and Configuration of Apache Mahout

Apache Mahout can be installed by downloading it from the website, or it can be installed using Maven. We will install it in both ways.

Like other big data tools installed on top of Apache Hadoop, we will install Apache Mahout in only the master node.

1. **Installation via tar file:**

- Download the latest version of Apache Mahout.

```
$ wget https://downloads.apache.org/mahout/14.1/apache-mahout-distribution-14.1.tar.gz-P /Downloads
$ sudo tar zxvf /Downloads/apache-mahout-* -C /usr/local
$ sudo mv /usr/local/apache-mahout-* /usr/local/mahout
$ sudo chown -R hduser:hadoop /usr/local/mahout
$ sudo gedit  /.bashrc
export MAHOUT_HOME=/usr/local/mahout
export PATH="$PATH":$HADOOP_HOME/bin:$MAHOUT_HOME/bin
export CLASSPATH="$CLASSPATH":$MAHOUT_HOME
$ source  /.bashrc
```

2. **Installation via Maven:**

- Install Maven.

```
$ sudo apt-get install maven
```

- Check Maven.

```
$ mvn -version
```

- Download and unzip the ZIP file.[12]

[12] http://ftp.wayne.edu/apache/mahout/14.1/mahout-14.1-source-release.zip.

- Move the folder to /usr/local/mahout.
- Go to the Mahout directory in the terminal.

```
$ cd /usr/local/mahout/
```

- Install Maven by following the command.

```
mvn install
```

5.9.2 Case Study

As a case study, we will perform data classification over the dataset[13] collected from Twitter using Naïve Bayes algorithm. With this application, tweets with many features, including location information, will be classified based on their posted location.

After downloading the dataset, we will execute the following codes to perform data classification using Apache Mahout:

- Creating folders in HDFS for source files

```
$ hadoop fs –mkdir DataSet
$ hadoop fs –mkdir DataSet/test
$ hadoop fs –mkdir DataSet/train
```

- Copying the source of the files into HDFS

```
$ hadoop fs –copyFromLocal DataSet/test/* DataSet/test
$ hadoop fs –copyFromLocal DataSet/train/* DataSet/train
```

- Converting the text data into the sequence file format

```
$ mahout seqdirectory –i DataSet –o DataSet/nbseqfiles
```

  -i → files directory (text)
  -o → output directory (sequence file)

- Converting sequence data into Term Frequency-inverse Document Frequency (TF-IDF) vect

```
$ mahout seq2sparse –i DataSet/nbseqfiles –o DataSet/nbsparse
```

- Splitting the dataset

```
$ mahout split –i DataSet/nbsparse/tfidf–vectors --trainingOutput
DataSet/nbTrain --testOutput DataSet/nbTest --randomSelectionPct 25
--overwrite --sequenceFiles –xm sequencial
```

[13] https://github.com/umitdemirbaga/TwitterData.

-i → files directory (tfidf-vectors sub directory)
–trainingOutput → directory for training data
–testOutput → directory for test data
–randomSelectionPort → percent of data to put in training
–overwrite → overwrite current data
–sequenceFiles → indicating that the files are of sequencial form
–xm → type of processing (sequencial/MapReduce)

- Building the Naïve Bayes model

```
$ mahout trainnb –i DataSet/nbTrain –el –li DataSet/nbLabels –o
DataSet/nbmodel –ow –c
```

-i → training files data
-el → extract labels from the files
-li → path to store the label index
-o → path to store the model
-ow → overwrite
-c → train complementary Naïve Bayes

- Testing the Naïve Bayes model

```
mahout testnb –i DataSet/nbTest –m DataSet/nbmodel –l DataSet/nbLa
bels –ow –o DataSet/nbpredictions –c
```

-i → test data directory
-m → model directory
-l → labels
-ow → overwrite
-o → predictions directory
-c → complimentary Naïve Bayes

Figure 5.20 shows the results of Naïve Bayes algorithm implemented in Apache Mahout that performs well with an 88.33% accuracy in tweet classification.

5.10 Learning Outcomes of the Chapter

- **Understanding the Main Characteristics of Big Data Analytics Platforms:** Analysing the fundamental characteristics, including distributed computing, data ingestion and integration, data storage and management, data processing and analysis, machine learning and advanced analytics, data visualisation and reporting.
- **Exploring Desired Properties of a Big Data System:** Evaluating properties such as robustness and fault tolerance, scalability, generalisation, extensibility, low-latency reads and updates, minimal maintenance, and debuggability in the context of big data systems.

```
=======================================================
Summary
-------------------------------------------------------
Correctly Classified Instances          :         53       88.3333%
Incorrectly Classified Instances        :          7       11.6667%
Total Classified Instances              :         60

=======================================================
Confusion Matrix
-------------------------------------------------------
a       b       <--Classified as
52      1        |  53          a     = test
6       1        |  7           b     = train

=======================================================
Statistics
-------------------------------------------------------
Kappa                                       0.3016
Accuracy                                   88.3333%
Reliability                                37.4663%
Reliability (standard deviation)            0.5301
Weighted precision                          0.8503
Weighted recall                             0.8833
Weighted F1 score                           0.8536

18/06/09 17:24:47 INFO driver.MahoutDriver: Program took 14387 ms (Minutes: 0.23978333333333332)
```

Fig. 5.20 Naïve Bayes implementation using Apache Mahout

- **Understanding Big Data Processing Systems:** Exploring different big data processing systems and their components, learning how to install them, and examining case studies demonstrating their role in big data analytics.
- **Exploring Big Data Processing with Hadoop:** Investigating the MapReduce paradigm, HDFS, YARN, and the installation of a multi-node Hadoop cluster.

References

1. L. Wang, J. Tao, R. Ranjan, H. Marten, A. Streit, J. Chen, D. Chen, G-Hadoop: MapReduce across distributed data centers for data-intensive computing. Future Gener. Comput. Syst. **29**(3), 739–750 (2013)
2. C. Ji, Q. Shao, J. Sun, S. Liu, L. Pan, L. Wu, C. Yang, Device data ingestion for industrial big data platforms with a case study. Sensors **16**(3), 279 (2016)
3. M.M. Najafabadi, F. Villanustre, T.M. Khoshgoftaar, N. Seliya, R. Wald, E. Muharemagic, Deep learning applications and challenges in big data analytics. J. Big Data **2**(1), 1–21 (2015)
4. L.T. Mohammed, A.A. AlHabshy, K.A. ElDahshan, Big data visualization: a survey, in *2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*. IEEE (2022), pp. 1–12
5. P. Jalote, *Fault Tolerance in Distributed Systems* (Prentice-Hall, Inc., 1994)
6. N. Khan, I. Yaqoob, I.A.T. Hashem, Z. Inayat, W.K. Mahmoud Ali, M. Alam, M. Shiraz, A. Gani, Big data: survey, technologies, opportunities, and challenges. Sci. World J. **2014** (2014)
7. J. Dean, Designs, lessons and advice from building large distributed systems. *Keynote from LADIS*, vol. 1 (2009)

8. M. Saadoon, S.H.A. Hamid, H. Sofian, H.H. Altarturi, Z.H. Azizul, N. Nasuha, Fault tolerance in big data storage and processing systems: a review on challenges and solutions. Ain Shams Eng. J. (2021)
9. N. Ayari, D. Barbaron, L. Lefevre, P. Primet, Fault tolerance for highly available internet services: concepts, approaches, and issues. IEEE Commun. Surv. Tutor. **10**(2), 34–46 (2008)
10. K. Kalia, N. Gupta, Analysis of Hadoop MapReduce scheduling in heterogeneous environment. Ain Shams Eng. J. **12**(1), 1101–1110 (2021)
11. Y. Cheng, X. Yu, W. Chen, R. Chang, Y. Xiang, A practical cross-datacenter fault-tolerance algorithm in the cloud storage system. Cluster Comput. **20**(2), 1801–1813 (2017)
12. A. Avizienis, Toward systematic design of fault-tolerant systems. Computer **30**(4), 51–58 (1997)
13. H.N. Rothberg, G.S. Erickson, Big data systems: knowledge transfer or intelligence insights? J. Knowl. Manag. **21**(1), 92–112 (2017)
14. U. Demirbaga, G.S. Aujla, Mapchain: a blockchain-based verifiable healthcare service management in IoT-based big data ecosystem. IEEE Trans. Netw. Serv. Manag. (2022)
15. J. Dean, S. Ghemawat, Mapreduce: simplified data processing on large clusters. Commun. ACM **51**(1), 107–113 (2008)
16. K. Shvachko, H. Kuang, S. Radia, R. Chansler, The hadoop distributed file system, in *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*. IEEE (2010), pp. 1–10
17. U. Demirbaga, HTwitt: a Hadoop-based platform for analysis and visualization of streaming twitter data. *Neural Computing and Applications* (2021), pp. 1–16

Further Reading

18. A.Y. Zomaya, S. Sakr (2017) Handbook of Big Data Technologies
19. J. Warren, N. Marz, Big Data: Principles and Best Practices of Scalable Realtime Data Systems. (Simon and Schuster, 2015)

Big Data Storage Solutions

6

Data storage is the backbone of the digital revolution. Without the ability to store and manage vast amounts of information, we cannot unleash the full potential of big data analytics.

—Virginia Rometty

Embarking on a journey through the landscape of big data storage solutions, this chapter unfolds the critical role these systems play in managing and extracting value from extensive datasets. It begins by highlighting the importance of storage systems, which explores traditional solutions like relational databases, data warehouses, NAS, and Storage Area Networks (SAN). The narrative then ventures into contemporary big data storage solutions, including the HDFS, diverse categories of NoSQL databases, and cloud storage solutions like Amazon S3 and Google Cloud Storage. Object storage systems, from distributed to hybrid configurations, are examined in detail. Further exploration encompasses in-memory databases, spanning both relational and non-relational types. The chapter concludes by guiding readers on selecting the most suitable big data storage solution based on factors such as scalability, performance requirements, emerging trends in storage technologies, edge computing, and the integration of artificial intelligence and machine learning.

6.1 Importance of Storage Systems for Big Data

Big data typically requires specialised data storage systems that manage data volume, velocity, and diversity. These systems must be scalable, adaptable, and efficient and must be able to manage and store significant amounts of data from several sources.

Ü. Demirbaga et al., *Big Data Analytics*,
https://doi.org/10.1007/978-3-031-55639-5_6

Big data storage systems are designed to store and manage enormous amounts of data from several sources, such as social media, sensors, and IoT devices.

The following explanations help to clarify the significance of various storage systems:

- **Effective handling of enormous data volumes**: Big data storage systems are built to handle massive data volumes, ranging from petabytes to exabytes. These systems are designed to handle data quickly and effectively by storing, retrieving, and managing it, which enables businesses to manage massive datasets without experiencing performance problems.
- **Real-time data processing**: Real-time data processing is adopted in many applications, including fraud detection, predictive maintenance, and financial analysis. Big data storage systems are made to handle and store data in a way that allows for real-time processing, enabling businesses to evaluate data as it is created.
- **Facilitating data analytics**: Access to enormous amounts of data from several sources is necessary for big data analytics. Analysts may access, analyse, and display data from many sources in a unified way thanks to storage solutions for big data, which act as a central repository for all data.
- **Facilitating cooperation and data sharing**: Big data storage solutions enable cooperation and knowledge sharing within the company by allowing numerous teams and people to access and share data. This aids in removing data silos and encouraging an organisation-wide data-driven culture.
- **Providing data privacy and security**: Data security and protection from illegal access are requirements for big data storage systems. To guarantee data security and privacy, these systems must meet data protection standards and include features like data encryption, access restrictions, and audits.
- **Enabling flexibility and scalability**: Big data storage solutions must be adaptable and scalable, enabling businesses to add or remove storage capacity as necessary. This is especially crucial with the rising data quantities and the need for companies to respond to shifting business requirements.

6.2 Traditional Storage Systems for Big Data

Various well-known technologies used for storing and handling massive amounts of data are included in traditional storage systems for big data. These technologies include SAN, NAS, relational databases, and data warehouses. While data warehouses are designed for analytical processing and historical analysis, relational databases offer a stable foundation for storing structured data with transactional support. NAS and SAN provide shared file-level and high-performance block-level access to meet distinct demands in cooperative and demanding data processing contexts. Effective large data storage and analysis depend on understanding the traits and trade-offs of these traditional storage technologies.

6.2.1 Relational Databases

Relational databases are established storage platforms often used to administer and store structured data. They work based on the relational data model, which arranges data into tables with predetermined schemas. Relational databases offer a reliable structure for data integrity, consistency, and dependability. They conduct data insertion, retrieval, update, and deletion activities using a SQL.

Relational databases have benefits for large data analytics, including transactional support, high consistency, and sophisticated query optimisation methods. However, when dealing with enormous amounts of data and fast-moving data streams, they could run into scalability problems. Relational databases' limited flexibility in processing unstructured or semi-structured data types frequently found in big data scenarios may also result from their strict schema.

Relational databases are widely used for data storage and management. Some examples of popular Relational Database Management Systems (RDBMS) include:

- **Oracle Database**: Oracle Database is a highly regarded and popular RDBMS that provides various tools and functions for effective data management, storage, and retrieval. With its strong transaction processing capabilities, Oracle Database excels at managing huge amounts of data while preserving the consistency and integrity of the data. Because of its superior scalability, businesses can easily increase their data storage as their needs change. Additionally, Oracle Database provides several optimisation methods, such as query and index optimisation, to boost query performance and increase system effectiveness.
- **MySQL**: MySQL is a well-liked open-source relational database management system known for its usability, performance, and dependability. It is extensively used in online applications, and many, particularly like its easy installation and use. MySQL has robust capabilities for maintaining structured data, supporting several data types, transactions, and concurrent access. It is appropriate for small to medium-sized applications because of its high performance and scalability. MySQL offers developers and organisations looking for a dependable and affordable RDBMS solution a strong foundation thanks to its vibrant community and thorough documentation.
- **Microsoft SQL Server**: SQL Server, created by Microsoft, is a robust and feature-rich relational database management system. It provides various features and functions for effective data management, such as cutting-edge security safeguards, high availability choices, and powerful reporting tools. When combined with other Microsoft products like the.NET framework and the Azure cloud platform, SQL Server creates a seamless environment for creating and deploying data-driven applications. Due to its user-friendly interface and thorough documentation, SQL Server is frequently used in enterprise contexts to handle crucial company data and support complicated analytical workloads.
- **PostgreSQL**: PostgreSQL is an open-source RDBMS recognised for its cutting-edge capabilities, scalability, and extensibility, which offers various data management functions by supporting complicated queries, data types, and transactions.

With the help of capabilities like table inheritance, views, and stored procedures, PostgreSQL enables programmers to create flexible and unique data structures. In addition, it supports several programming languages, allowing easy interaction with various application frameworks. The extensive and active community that supports PostgreSQL's development makes it a dependable option for businesses looking for an expandable and trustworthy RDBMS solution.

- **IBM Db2**: IBM Db2 is an enterprise-grade RDBMS developed to manage high-performance transactional and analytical workloads. It offers many features and functionalities, such as cutting-edge security measures, support for hybrid cloud settings, and effective data compression methods. Db2 is renowned for its scalability, which enables businesses to manage increasing data quantities and effectively interact with different data sources. Db2 offers organisations a strong and adaptable data management solution for their crucial business activities thanks to its multi-platform support and large toolkit.

6.2.2 Data Warehouses

Data warehouses are specialised repositories created to store and manage massive amounts of structured data from multiple sources [1]. They offer detailed searches and aggregations and are optimised for analytical processing. Data warehouses collect data from various operating systems, turn it into a common format, and then put it into the warehouse for analysis using an ETL procedure.

Data warehouses offer a unified perspective of an organisation's data to enable decision-making and historical analysis. They frequently use a multidimensional structure, such as a star or snowflake schema, to help effective data retrieval and analysis. Data warehouses, on the other hand, are best suited for structured data and may have trouble with the unstructured or semi-structured data types that are frequently present in big data.

This is a thorough list of well known and renowned data warehousing systems that have been well known in the market because of their strong capabilities and efficiency when managing enormous amounts of structured data:

- **Amazon Redshift**: Amazon Redshift is a fully managed data warehousing solution provided by AWS. It offers high-performance analytics and sophisticated queries and is built to handle enormous datasets. Redshift's columnar storage, parallel query processing, and autonomous scaling features enable businesses to quickly and affordably analyse massive amounts of data.
- **Snowflake**: Snowflake, a cloud-based data warehousing platform, is a scalable and extremely elastic data storage and analytics environment. Users may freely grow each component based on their needs since it separates storage and computation resources. Snowflake provides cutting-edge features, including automated query optimisation, built-in data sharing, and support for semi-structured data formats like JSON and Avro.

- **Google BigQuery**: Google BigQuery is a serverless data warehousing option offered by Google Cloud, which provides a fully managed, very scalable, and reasonably priced platform for storing and processing big data. BigQuery offers capabilities, including automated data segmentation, columnar storage, and interaction with other Google Cloud services for smooth data processing and analysis, and it supports SQL queries.
- **Microsoft Azure Synapse Analytics**: Microsoft Azure Synapse Analytics, formerly Azure SQL Data Warehouse, is a cloud-based data warehousing solution that interfaces with the larger Azure ecosystem. It offers scalable computing and storage capabilities for analytics, processing, and data storage. Azure Synapse Analytics includes capabilities including intelligent caching, workload separation, and close connection with other Azure services for comprehensive data management and analysis.
- **IBM Netezza**: IBM Netezza is an appliance-based data warehousing solution for high-performance analytics, which combines software and hardware to provide quick data loading, query execution, and parallel processing capabilities. Netezza provides in-database analytics, sophisticated analytics features, and interaction with IBM's more comprehensive data management and analytics offering.

6.2.3 Network Attached Storage (NAS)

NAS is a storage system allowing network-based file-level data access [2]. It functions as a standalone server devoted to file storage and retrieval and comprises one or more storage devices linked to a network. Clients may access and manage files stored on NAS devices using conventional network protocols like Network File System (NFS) or Server Message Block (SMB).

NAS has advantages, including simple management, scalability, and ease of implementation. It can be accessed from many platforms and operating systems and offers a shared storage solution for several customers. NAS systems are advantageous for cooperative big data analytics because they are appropriate when several users or applications require concurrent access to shared data. However, due to possible network congestion and storage device restrictions, NAS performance may suffer under intense workloads or while working with huge datasets.

- **Synology DiskStation**: Synology is well known for its broad selection of NAS systems, with the DiskStation series distinguishing out for its user-friendly interface and powerful feature set. The adaptability of Synology DiskStation models enables customers to increase storage capacity as their demands change. These NAS systems include cutting-edge data protection features, including Redundant Array of Independent Disks (RAID) setups, snapshot technology, and built-in encryption for increased security. Additionally, Synology's DiskStation Manager (DSM) operating system provides a wide range of applications, such as multi-

media streaming, backup options, and virtualization support, which makes it a flexible choice for home and small business customers.

- **QNAP Turbo NAS**: The Turbo NAS series from Quality Network Appliance Provider (QNAP) is renowned for its robust hardware and numerous capabilities that meet various storage requirements by offering high-performance file sharing and data management features. The QTS operating system from QNAP provides a simple user interface and a wide variety of productivity-enhancing tools, such as backup and disaster recovery features, virtualization compatibility, and multimedia streaming capabilities. The Turbo NAS versions also have extendable storage capabilities, enabling customers to easily meet their expanding data requirements.
- **Western Digital My Cloud**: Western Digital's My Cloud series targets home and small office environments focussing on use and simplicity. These NAS systems feature a simple setup procedure and easy management through a user-friendly interface. Users can access and share their data from anywhere with My Cloud devices' remote access features. They also link with well-known cloud services for smooth file synchronisation and backup options. The My Cloud series emphasises data privacy and security and gives consumers total control over their private information.
- **Netgear ReadyNAS**: Netgear's ReadyNAS series provides reliable and scalable NAS solutions for demanding home customers and commercial users. These NAS units include strong hardware that enables fast file sharing and data management. The ReadyNAS systems from Netgear allow a variety of RAID configurations for data security and advanced backup options for disaster recovery. Users may utilise their NAS systems to operate virtual machines thanks to virtualization features, and cloud backup integration adds another layer of data security. The ReadyNAS range is renowned for its dependability and adaptability, meeting the demands of both power users and small- to medium-sized organisations.
- **Buffalo TeraStation**: Buffalo TeraStation offers high-capacity NAS solutions emphasising file sharing and data security. Designed with redundancy, these devices allow RAID configurations to guarantee data availability and integrity. Buffalo TeraStation models give users various backup choices, including local and distant replication support and cloud storage service integration. These NAS systems include capabilities including Active Directory connection, snapshot technology for point-in-time recovery, and scalability to meet rising storage needs. They are intended for small to medium-sized organisations.

6.2.4 Storage Area Networks (SAN)

SANs are high-performance storage networks created for block-level data access [3]. By creating a specific network connection between servers and storage resources, they offer direct access to storage devices. Block-level data is transferred between servers and storage devices using SANs using protocols like Fibre Channel or Internet Small Computer System Interface (iSCSI).

SANs offer some benefits, including high throughput, minimal latency, and centralised storage management, which perform well in situations requiring quick and dependable access to significant data quantities, making them suitable for heavy data processing in big data analytics. SANs may extend vertically by expanding the capacity of already existing storage arrays or horizontally by adding more storage arrays. However, setting up and managing SANs may be challenging and require specialised training and experience.

Here are a few examples of popular SAN solutions:

- **Dell EMC PowerMax**: Dell EMC PowerMax is an enterprise-class SAN system noted for its superior performance, scalability, and cutting-edge data management capabilities, offering high-speed block-level access to centralised storage resources while being able to manage enormous workloads. With its wide range of scalability choices, businesses can quickly increase storage capabilities to meet rising demands. PowerMax uses cutting-edge technology like Non-Volatile Memory Express (NVMe) and machine learning techniques to enhance performance, better data location, and provide blazingly quick reaction times. To ensure stored data's security, effectiveness, and dependability, it also offers a full range of data management functions, such as data deduplication, compression, and encryption.
- **HPE StoreServ (3PAR)**: HPE StoreServ, commonly known as HPE 3PAR, is a reliable SAN system that combines storage virtualization, sophisticated data tiering, and powerful data protection features. StoreServ delivers smooth scalability and supports all-flash and hybrid storage arrays to accommodate a variety of workloads. Organisations can combine storage resources from several suppliers into a centrally managed single pool with the help of its storage virtualisation features, which facilitate administration and improve resource utilisation. StoreServ combines autonomous management, adaptive optimization, and policy-based automation to enhance performance, guarantee data availability, and promote effective storage provisioning. Additionally, it offers integrated data protection methods, including remote replication, data-at-rest encryption, and snapshots, ensuring complete data security and business continuity.
- **IBM SAN Volume Controller**: The IBM SAN Volume Controller (SVC) is a SAN system that offers centralised control and effective use of storage resources while bringing virtualization features to storage settings. SVC enables businesses to abstract physical storage arrays from several manufacturers and display them as a unified logical entity, combining them into a virtualized storage pool. This makes it possible to streamline storage management, move data seamlessly, and increase storage effectiveness. SVC uses cutting-edge technologies, including compression, automatic tiering, and thin provisioning, to maximise storage efficiency and cut costs. Organisations can effortlessly integrate with current infrastructure because of SVC's numerous connectivity choices, which include iSCSI and Fibre channels. To guarantee data accessibility and business continuity, it also offers data protection features, including point-in-time snapshots, remote replication, and disaster recovery capabilities.

- **NetApp SANtricity**: NetApp SANtricity is a robust SAN solution created for the high-performance storage needs of business environments. Many features are available, including dynamic disc pools, data protection measures, and sophisticated storage management skills. SANtricity allows businesses to customise storage array configurations to meet their unique requirements, facilitating the dynamic and effective deployment of storage resources. For speedy and dependable data recovery, it integrates snapshot-based data protection. Fibre Channel and iSCSI connectivity options are supported by SANtricity, enabling businesses to select the network architecture that best suits their needs. SANtricity's powerful storage management tools simplify administrative duties and offer an in-depth storage performance analysis, resulting in optimised storage operations and improved data dependability.
- **Pure Storage FlashArray**: Pure Storage FlashArray is a SAN system that focuses on providing high-performance all-flash storage arrays. It is perfect for demanding workloads that need quick data processing since it allows for extremely low-latency and high-speed access to data. FlashArray uses Pure Storage's patented hardware and software solutions to maximise storage economy and provide reliable performance. It includes features such as data reduction methods (compression and deduplication) to make the most of the available storage space. Enterprise-grade data services from FlashArray, such as data replication, data protection, and data encryption, guarantee the safety and accessibility of crucial data. Because of its scalable design, businesses can easily increase storage capabilities to keep up with growth and changing storage needs.

6.3 Big Data Storage Solutions

The selection of data storage for large-scale data relies on the application's particular needs, including the type and amount of data being kept, the rate at which data is created and processed, and the organisation's budget and infrastructure. Organisations can guarantee that their big data applications can efficiently and effectively store and handle enormous amounts of data by choosing the appropriate storage solution.

A taxonomy of large data storage and management systems is reproduced in Fig. 6.1.

6.3.1 Hadoop Distributed File System (HDFS)

HDFS is a distributed file system that stores and handles vast amounts of data across several commodity systems [4]. The Hadoop ecosystem, a well-liked framework for managing and analysing massive data, uses it as its primary storage system. Data blocks are stored in HDFS and dispersed among many cluster nodes. To ensure data durability and availability in the case of a node failure, each block is duplicated nu-

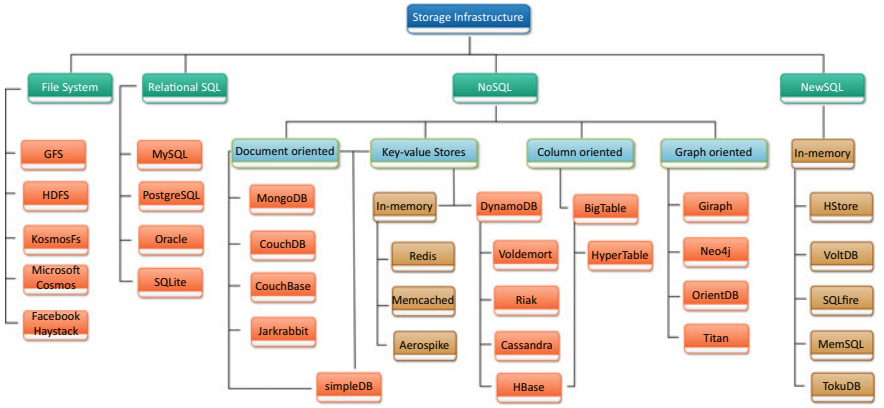


Fig. 6.1 Taxonomy of big data storage systems

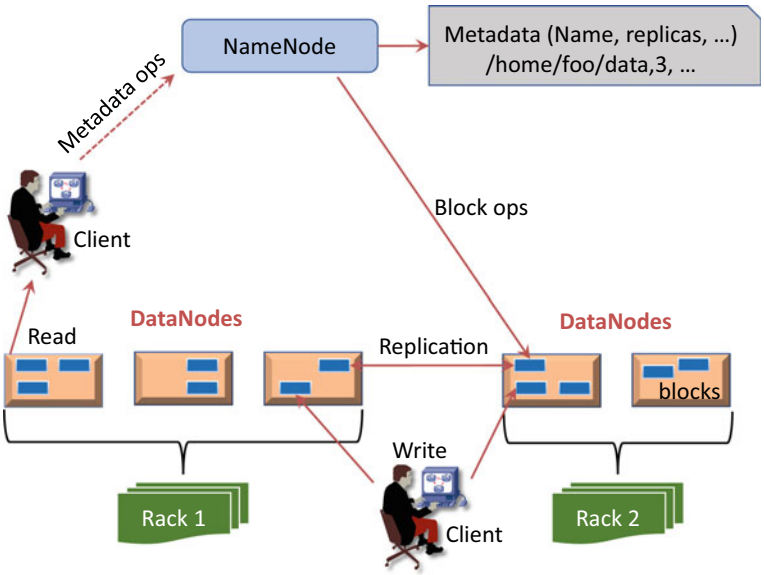


Fig. 6.2 HDFS architecture

merous times, often three times. Large files, including individual files in the petabyte range, are supported by HDFS. It is utilised in many different applications, such as log processing, data warehousing, and machine learning, and is suited for batch processing. YARN, Spark, Flink, and other frameworks and tools for distributed computing, all part of the Hadoop ecosystem, also utilise HDFS. Figure 6.2 shows the architecture of HDFS.

The NameNode, which serves as the single point of contact for clients and manages file system operations, is the heart of HDFS. It keeps track of the placement

and arrangement of data blocks and handles the file namespace, directory tree, and tracking system. The NameNode keeps the metadata in memory, making it possible to read and retrieve file contents quickly. The real data is saved in blocks dispersed throughout many cluster DataNodes. The read-and-write operations on the stored data are carried out by DataNodes, which also manages storing and serving the data blocks. To report their availability, block status, and general health, they interact with the NameNode. Communication between the NameNode and DataNodes is essential to keep the file system's integrity. When creating, deleting, and replicating blocks, the NameNode gives instructions to the DataNodes. To streamline data processing, it also manages block allocation and data localization.

6.3.2 NoSQL Databases

NoSQL databases are non-relational databases developed to manage unstructured and partially structured data. They can handle massive quantities of data quickly and are often very scalable. NoSQL databases do not employ the conventional tabular relational data model utilised by SQL-based (Structured Query Language) databases [5]. They are all made to handle big, high-velocity, and unstructured data because conventional SQL-based databases are unsuitable. NoSQL databases are well-liked for their scalability, flexibility, and performance. They are utilised in various contexts, including e-commerce, social networking, gaming, and financial services. However, they have significant drawbacks, including a lack of standardisation, restricted query capabilities, and eventual consistency.

There are four primary types of NoSQL databases used for big data most frequently:

6.3.2.1 Key-value Databases

In key-value databases, data is stored as a set of key-value pairs, where each key corresponds to a different value [6]. The keys and values may be strings, integers, or binary data. A hash table is frequently used to store the key-value pairs, making it possible to retrieve data using the key efficiently. The simplicity of key-value storage is one of its benefits. The key-value data architecture is simple to comprehend and use. Because there is no predefined schema, new data may be added or removed without changing the database's organisational structure. Key-value storage is, hence, highly adaptable and quick. The excellent performance of key-value storage is another benefit. Although a straightforward hash table lookup may be used to retrieve data based on a key, doing so is incredibly quick. In-memory databases, key-value stores, are frequently employed for real-time applications that need high throughput and low latency.

Nevertheless, a drawback of key-value stores is that they don't enable detailed searches or intricate data associations. There is no intrinsic mechanism to query data based on criteria other than the key because it is stored as key-value pairs. Some key-value stores may support range searches and secondary indexes, although these

capabilities are often less versatile and practical than those relational databases offer. Key-value stores might not be appropriate for applications that need extensive data searches or analysis. Redis, Riak, and Amazon DynamoDB are some examples of well-known key-value stores.

6.3.2.2 Columnar Databases

Columnar databases, which store data in columns rather than rows, are often called column-family stores or wide-column stores [7]. Data is divided into groups of similar columns called "columns" in columnar storage. The number of columns in each column family is unlimited, and each column's data type is complete. Analytical queries, particularly those involving vast volumes of data and requiring aggregations, filtering, or sorting based on several columns, are best served by columnar storage. Columnar stores may obtain only the data required for a particular query by storing data in columns rather than rows, leading to quicker query speeds and less storage space.

The ability of this type of database to handle analytical queries is one of its benefits. Columnar stores can do selective scans, which means they can read just the columns required by a particular query because they store data in columns. This may result in shorter query response times and less storage use. The flexibility of columnar storage to grow horizontally across several nodes or servers is another benefit. Columnar stores can manage massive volumes of data and offer high availability and fault tolerance since they can distribute data across numerous servers or nodes.

Columnar storage might not be appropriate for all user situations, though. They may not function as effectively for transactional queries or real-time data processing since they are optimised for analytical queries. Due to the possibility of changing or adding columns necessitating data rearrangement, columnar storage also demands careful schema design. Apache Cassandra, HBase, and Google Bigtable are examples of popular columnar databases.

6.3.2.3 Document Databases

Data is stored in document storage as documents, usually in JSON or Binary Encoded Javascript Object Notation (BSON) format. Each document comprises a set of key-value pairs that define the characteristics or attributes of a particular item or entity [8]. It's well-recognised that document storage is adaptable, scalable, and simple to use. They are frequently employed in use cases, including content management, online applications, and other dynamic and sophisticated data structures. The adaptability of document repositories is one benefit. As documents are stored in a schema-less manner, new data may be added or removed without having to alter the database structure. This makes document repositories particularly agile and adaptive to changing data requirements.

The scalability of document stores is an additional benefit. To offer high availability and fault tolerance, they may handle enormous volumes of data and distribute it over several servers or nodes. Moreover, document stores offer strong query capa-

bilities. A wide range of query operators, including range queries, full-text searches, and geographical queries, are supported by many document storage. Some document repositories include aggregation pipelines or supplementary indexes, which can facilitate sophisticated analytics or reporting.

Document-based databases might not, however, be appropriate in every situation. Some pages might be duplicated as the data is kept in a denormalized format. This may result in more significant storage needs and longer query times for some queries. Popular document storage includes Couchbase, MongoDB, and CouchDB.

6.3.2.4 Graph Databases

Data is stored in graph databases as nodes, edges, and attributes. Although edges describe connections or interactions between nodes, nodes represent entities like people, places, or objects [9]. Properties are key-value pairs connected to nodes or edges and provide extra information. Graph databases can manage huge, intricate information and are very effective at modelling and querying complicated relationships. They are frequently employed in fraud detection, recommendation engines, and social networking applications. Graph databases also offer a significant degree of flexibility in terms of data modelling. Without changing the database structure, nodes, edges, and attributes can be added or changed easily, which makes it possible for a data model to be flexible, adaptive, and change over time.

The key advantage of graph databases is that they can efficiently model and traverse complex relationships. Graph databases use a "traversal" approach to querying data, which means that queries start at a specific node and cross along edges to reach related nodes. This allows for highly targeted queries that can efficiently find patterns and relationships within the data.

Graph databases might not be appropriate for all use cases, though. Some queries, such as those that demand intricate aggregation or grouping, may be less effective. They may also require specific knowledge or skills to utilise efficiently and tend to have a higher learning curve than other databases. Neo4j, OrientDB, and JanusGraph are popular graph databases.

6.3.3 Cloud Storage Solutions

Big data applications can use scalable and adaptable cloud storage solutions that enable users to store and retrieve data online without using physical storage devices. Cloud service providers offer these solutions and frequently include pay-as-you-go pricing for scalable, on-demand storage. Many advantages come with cloud storage solutions, such as simple accessibility, scalability, data backup, and collaboration tools. They also have certain drawbacks, such as reliance on Internet access and security issues. Users should thoroughly assess their demands and select the best option depending on their needs.

The following are a few well-known cloud storage options.

6.3.3.1 Amazon S3

Amazon S3, provided by AWS, is a highly scalable and secure cloud-based object storage solution for big data. It provides companies with a solid storage infrastructure to handle and retrieve enormous volumes of data efficiently. Amazon S3, which uses a distributed storage architecture, automatically replicates data across several servers and availability zones to provide durability and high availability. The design of S3 is based on the idea of "buckets," which serve as logical containers for access control and data storage. Unrestricted scalability, industry-leading data availability and durability, thorough security measures including access control lists and encryption, and data lifecycle management capabilities are notable characteristics of S3. Amazon S3 offers enterprises a financially sensible alternative with a pay-as-you-go pricing model. With no upfront infrastructure investment required through this pricing structure, businesses can deploy resources more effectively and cut costs. Businesses can scale their storage requirements according to their unique demands by only paying for their storage resources, preventing overprovisioning, and cutting down on needless expenses.

Amazon S3 connects with the other AWS ecosystem services without a hitch. This connectivity creates a complete data processing, analytics, and visualisation ecosystem. Organisations may connect to services like AWS Lambda, Amazon Athena, and Amazon Redshift to optimise their big data operations. They use various tools and services to extract insights, conduct complicated analyses, and visualise data. By enabling organisations to build end-to-end solutions inside a single environment, the interoperability and synergy of AWS services improve productivity and encourage smooth data operations.

Enumerated below are several pivotal characteristics inherent to Amazon S3:

1. **Scalability**: Organisations can store and access practically any quantity of data using Amazon S3's virtually limitless scalability. S3 manages data expansion without any hiccups, obviating the need for upfront capacity planning and empowering organisations to increase their storage resources as required.
2. **Durability and Availability**: Amazon S3 automatically replicates data across numerous servers in various availability zones using its distributed architecture. Providing great durability through redundancy reduces the danger of data loss or unavailability. Amazon S3 offers an SLA of 99.99% for data availability and 99.999999999% (11 nines) for data durability.
3. **Security**: Data storage is protected by strong security features built into Amazon S3. Access control lists (ACLs), bucket policies, and AWS IAM roles are just a few of the many levels of security it offers. Additionally, AWS Key Management Service (KMS) or customer-provided keys can encrypt data at rest, guaranteeing data confidentiality.
4. **Data Lifecycle Management**: A wide range of data lifecycle management options is available with Amazon S3, enabling organisations to set up policies for automatically moving data across storage classes by consumption trends, financial constraints, or legal obligations. This functionality optimises cost by automatically shifting data inaccessible to lower-cost storage tiers.

Considering the aforementioned attributes, Amazon S3 offers substantial advantages in the context of big data storage.

- **Cost-Effectiveness**: Organisations can use Amazon S3's pay-as-you-go pricing model and avoid making upfront infrastructure investments. As Amazon S3 is scalable, organisations only pay for their storage capacity, reducing overprovisioning expenses.
- **Data Accessibility**: For organisations to swiftly recover data for processing and analysis, Amazon S3 offers easy and dependable data access. The simplicity with which Amazon S3 may be incorporated into current big data processes is improved by its interoperability with various data analytics tools and frameworks.
- **Integration with AWS Ecosystem**: Amazon S3 works easily with other cloud services, including AWS Lambda, Amazon Athena, and Amazon Redshift, as part of the AWS portfolio of services. Through this connectivity, organisations can use various tools and services for data processing, analytics, and visualisation.

6.3.3.2 Google Cloud Storage

Google Cloud Storage is a reliable and scalable cloud-based object storage solution offered by GCP, providing businesses with a reliable and adaptable data access and storage infrastructure. Google Cloud Storage, which runs on a distributed architecture across several data centres, uses geo-redundancy and replication technologies to provide excellent data durability and availability. Businesses can easily handle increasing data volumes because of its elastic scaling feature, which enables a pay-as-you-go pricing structure that minimises expenses. Google Cloud Storage prioritises data safety and compliance using cutting-edge security methods, including encryption and access control systems.

In addition, its seamless connection with the GCP ecosystem enables businesses to use a full range of data processing, analytics, and machine learning technologies for improved big data workflows. Google Cloud Storage enables enterprises to store, manage, and analyse their big data assets effectively, enabling quick decision-making and extracting insightful information. It does this with low-latency and efficient data accessibility.

Distinct features of Google Cloud Storage are indicated below:

1. **Elastic Scalability**: The elastic scalability feature of Google Cloud Storage enables businesses to manage increasing data volumes easily. Autonomously scaling storage resources to match changing needs removes the need for upfront capacity planning and provides the best performance for big data analytics applications.
2. **Geo-redundancy and High Durability**: Google Cloud Storage replicates data across geographically separated sites through a distributed infrastructure spread over numerous data centres. With an amazing SLA to reduce the risk of data loss or unavailability, this strategy offers great data durability and availability.

3. **Advanced Security Measures**: Google Cloud Storage prioritises data security by providing a wide range of security protections. With built-in compliance frameworks, encryption at rest and in transit, granular access control methods using Google Cloud IAM, and fine-grained access control mechanisms, it provides organisations with strong data security and regulatory compliance for their big data assets.

Distinctive advantages for big data storage are listed below:

- **Cost-Effective Pricing Model**: With a pay-as-you-go pricing approach, Google Cloud Storage enables companies to save expenses by only paying for the storage capacity they use. Organisations may make their big data storage operations more cost-effective without making a large upfront infrastructure expenditure.
- **Seamless Integration with GCP Ecosystem**: Within the GCP ecosystem, many services and tools are smoothly integrated with Google Cloud Storage. With the help of this connection, businesses can use all of GCP's capabilities for big data workflows, including its powerful data processing, analytics, and machine learning capabilities, simplifying and improving their entire data operations.
- **Efficient Data Accessibility and Performance**: Google Cloud Storage provides swift and reliable data access, enabling effective data processing and analysis. With its quick data retrieval capabilities, it supports big data applications that require real-time analytics and decision-making.

6.3.3.3 Microsoft Azure Blob Storage

Microsoft Azure Blob Storage offers a cloud storage solution for huge data storage needs. Its scalable design, tiered storage choices, strong data protection, and easy connectivity with the Azure ecosystem enable companies to effectively store, manage, and analyse their big data assets. Azure Blob Storage is desirable for enterprises looking for reliable storage due to its scalability, affordability, data analytic capabilities, and regulatory adherence advantages.

Furthermore, Microsoft Azure Blob Storage offers other benefits that increase its attractiveness as a reliable big data storage option. Its seamless interface with Azure Data Factory, a data integration service, is one of its primary advantages since it enables businesses to effectively orchestrate and automate data activities across numerous sources and destinations. With this connection, ingesting, converting, and loading data into Azure Blob Storage is simpler, allowing for more efficient data management and quick data-driven insights. Additionally, Azure Blob Storage provides built-in support for sophisticated analytics tools like Azure Machine Learning and Power BI, enabling businesses to take advantage of strong data analytics capabilities for sifting through their big data repositories to extract useful patterns, trends, and predictive insights. By utilising these sophisticated analytics tools, businesses can fully use their big data assets and get a competitive edge in today's data-driven environment.

The following provides an extensive exposition of the notable features exhibited by Azure Blob Storage, encapsulating the key characteristics and functionalities that underpin its efficacy and versatility as a cloud storage solution for accommodating big data storage requirements.

1. **Blob Storage**: Unstructured data can be stored as blobs using the specialised storage solution offered by Azure Blob Storage. This feature is appropriate for big data storage since it enables organisations to efficiently store and handle massive files and objects.
2. **Tiered Storage**: Azure Blob Storage offers multiple storage tiers, including hot, cool, and archive tiers. This tiered storage model allows organisations to optimise costs based on data access frequency and performance requirements. Frequently accessed data can be stored in the hot tier, while less frequently accessed data can be moved to the cool or archive tiers for cost savings.
3. **Data Protection and Security**: Data security and protection are top priorities for Microsoft Azure Blob Storage. Data confidentiality and integrity are ensured by encryption both at rest and while in transit. For example, shared Access Signatures and Azure Active Directory integration offer granular control over data access, reducing the risk of unauthorised access.

The ensuing section delineates the notable advantages conferred by Microsoft Azure Blob Storage for big data storage:

- **Enabling Unbounded Growth and Adaptability**: Azure Blob Storage offers scalable storage choices, enabling businesses to handle increasing amounts of big data. Data is highly available and durable by the distributed design and automated replication, and dynamic data expansion is supported by the on-demand scalability of storage resources.
- **Optimising Expenditure for Efficient Data Storage**: Organisations can save expenses by matching storage charges to data usage patterns through Azure Blob Storage's tiered storage concept, which enables businesses to manage resources efficiently, cutting storage costs for their large data assets.
- **Unveiling Actionable Knowledge through Robust Analytics**: Integration with Azure services, such as Azure Databricks and Azure Data Lake Analytics, offers efficient data processing and analysis. Organisations can use these technologies to extract useful insights from their big data to support data-driven decision-making and improve business outcomes.
- **Ensuring Regulatory Adherence and Data Governance**: The availability of compliance certifications from Azure Blob Storage, including International Organisation for Standardisation (ISO), SOC, and GDPR, guarantees conformity to industry-standard legal criteria. With its data privacy and control assurance, this feature is essential for organisations managing sensitive data.

6.3.3.4 IBM Cloud Object Storage

IBM Cloud Infrastructure provides a reliable and expandable cloud storage option, IBM Cloud Object Storage. It is made to manage enormous volumes of unstructured data and give businesses safe, reliable, and affordable storage alternatives. The distributed architecture that IBM Cloud Object Storage uses across various data centres makes high availability and data redundancy possible. This guarantees data protection against errors and enables easy access to saved items. The storage system is based on IBM's proprietary SecureSlice technology, which separates data into encrypted slices and distributes them among several storage nodes. This strategy improves data security and privacy because only one piece of the dataset is at risk from unauthorised access.

Considering various usage patterns and cost concerns, flexible storage classes are a standout feature of IBM Cloud Object Storage. Standard, Vault, and Cold storage classes are available. Regularly accessed data may be obtained quickly and directly using the Standard storage class. The Vault storage type is designed for less-often accessed material that needs quick recovery when needed. For long-term retention of data with low access needs, the cold storage class is appropriate. By offering several storage classes, IBM Cloud Object Storage enables businesses to optimise their storage expenses based on how frequently they access their data.

The key features of IBM Cloud Object Storage include:

1. **Scalability**: Organisations can store and manage enormous volumes of unstructured data thanks to IBM Cloud Object Storage's nearly infinite scalability. Without sacrificing speed or availability, it can easily manage data volumes ranging from petabytes to exabytes.
2. **Durability and Availability**: The storage solution is built with high availability and durability in mind. It uses a distributed architecture across several data centres to protect against data redundancy and failure. This architecture ensures uninterrupted access to stored items despite hardware or network failures.
3. **SecureSlice Technology**: Data is encrypted and divided into slices using the SecureSlice technology, which IBM Cloud Object Storage uses. The security and privacy of the data are then improved by distributing these slices among other storage nodes. Unauthorised access to one slice does not jeopardise the integrity of the entire data collection.
4. **Flexible Storage Classes**: Flexible storage classes are available from IBM Cloud Object Storage to accommodate various access patterns and financial constraints. Regularly used data can be obtained immediately by the Standard storage class. The Vault storage type is designed for data that needs quick retrieval yet has a low access frequency. Cold storage is a form of storage that is appropriate for long-term archiving of rarely accessed data.
5. **Integration and Data Management**: The storage solution enables enhanced data analytics and insights by easily integrating with Watson and other IBM Cloud services. Additionally, it provides data lifecycle management, allowing businesses to automate the transfer and destruction of data by predetermined rules.

6. **Data Encryption and Security**: The confidentiality and integrity of data are guaranteed by encryption supported by IBM Cloud Object Storage at rest and while in transit. Access controls, user authentication, and audit logs are sophisticated security measures it offers to safeguard data from unauthorised access and satisfy compliance standards.
7. **Cost-Effectiveness**: The pay-as-you-go price structure and variable storage classes in IBM Cloud Object Storage make it an affordable option. Organisations can reduce their storage expenses by choosing the best storage class depending on their data access patterns.
8. **API and Software Development Kit (SDK) Support**: Developers can easily connect and communicate with the storage solution with the help of IBM Cloud Object Storage's broad API and SDK support. It simplifies developing apps that take advantage of IBM Cloud Object Storage's features by providing SDKs for several programming languages.

6.3.3.5 Dropbox

Dropbox offers a simple and effective cloud storage option for those who need to store large amounts of data on the cloud. It is a dependable option for people and organisations due to its solid design, file synchronisation capabilities, collaboration features, and security precautions. Dropbox enables users to successfully store, retrieve, and collaborate on huge data assets by putting accessibility, scalability, and data safety at the centre of its design, boosting productivity and simplifying data management.

Dropbox's architecture comprises a globally dispersed network of computers housed in data centres. A distributed file system stores the data, making redundancy and efficient data access possible. The infrastructure guarantees the availability and integrity of stored data and has strong safeguards to guard against data loss or corruption.

Key features of Dropbox are listed below:

1. **File Synchronisation**: Dropbox lets users view and work on files smoothly by syncing data across various devices. Any modifications made to files are instantly synchronised with other connected devices, ensuring users always have access to the most recent version.
2. **Collaboration and Sharing**: Dropbox offers tools for teamwork that let users share files and folders with others. With the ability for numerous users to modify and comment on shared files in real time, it promotes productivity and effective teamwork.
3. **Version Control and Recovery**: Dropbox keeps track of file revisions and lets users return to earlier versions if necessary. This function guarantees data integrity and safeguards against unintentional changes or data loss.
4. **Security Measures**: Dropbox implements robust security measures to protect data during transmission and storage, including data encryption in transit and at

rest, access controls, two-factor authentication, and compliance with industry-standard security practices.

Along with such features, there are some important benefits provided by Dropbox:

- **Accessibility and Convenience**: Dropbox has a user-friendly design that gives easy access to saved files on various devices. Users dealing with huge data are more comfortable and productive via its synchronisation capability and straightforward procedure.
- **Collaboration and Teamwork**: With Dropbox's collaboration tools, teams can collaborate more effectively on large data projects by sharing huge files, real-time collaboration, and project management and communication.
- **Scalability and Reliability**: With Dropbox's architecture, organisations can manage growing amounts of big data, guaranteeing scalability and dependability. A dispersed network of computers makes it possible to store and retrieve data effectively, reducing downtime and providing high availability.
- **Data Backup and Recovery**: Dropbox's version control and recovery tools give an extra degree of data security, allowing users to restore lost files or roll back to earlier versions in the case of data loss or unintentional changes, protecting important large data assets.

6.3.4 Object Storage Systems

In contrast to conventional file hierarchies or block-level storage, object storage systems are a sort of data storage architecture that organises and maintains data as discrete objects. Each item in an object storage system is given a special identification number, commonly known as a Universal Unique Identifier (UUID), which makes it possible to retrieve and manipulate data quickly and easily. The ability of object storage systems to store substantial amounts of unstructured data, including multimedia files, documents, and sensor data, is one of its primary advantages. Object storage systems store things on a flat address space, making them extremely scalable and able to handle massive quantities of data, unlike standard file systems that organise data into a hierarchical directory structure.

Object storage systems provide several benefits compared to traditional storage designs. By spreading data over several storage nodes, they provide excellent durability and availability while providing redundancy and fault tolerance. Object storage solutions also offer seamless scalability, enabling businesses to increase their storage capacity as data quantities increase. Another important benefit is the interoperability of object storage systems with cloud computing environments. The underlying storage technology for many cloud service providers' platforms is object storage, allowing for easy interaction with cloud-based services and applications. In addition, object storage systems frequently include robust metadata capabilities, enabling effective data indexing, searching, and workflows driven by metadata. Faster data

retrieval is made possible by this feature, which also aids machine learning and sophisticated data analytics applications.

Various object storage systems are available, each with unique features and functionalities. Popular varieties are detailed below.

6.3.4.1 Distributed Object Storage

These systems offer great scalability and fault tolerance by distributing data over several storage nodes or servers. Data replication across several sites is possible via distributed object storage, allowing high availability and data durability.

6.3.4.2 Cloud-based Object Storage

Cloud service providers provide object storage, enabling businesses to store and access data online. As customers only pay for the storage resources they use, cloud-based object storage offers consumers flexibility, scalability, and cost-effectiveness.

6.3.4.3 Scale-out Object Storage

Massive scalability is built into this kind of object storage system. Adding more storage nodes or clusters enables businesses to extend their storage infrastructure and make sure that storage capacity can readily increase to handle the huge data volume that is always growing.

6.3.4.4 Object Storage with Versioning

Organisations can track various iterations of the same thing using versioning, which some object storage systems support. This functionality ensures that prior versions may be accessed and restored if necessary, which is especially helpful in situations where data changes often.

6.3.4.5 Hybrid Object Storage

The advantages of both on-premises and cloud-based storage are combined in hybrid object storage systems. They enable businesses to archive less-often accessed data in the cloud to reduce expenses while storing frequently accessed data on local storage devices for quick retrieval.

6.3.5 In-Memory Databases

Big data can be stored and managed in memory rather than on disc with the help of in-memory databases like Apache Ignite and SAP HANA. In-memory databases are the best choice for real-time analytics and other applications that need low-latency data access since this enables rapid access to data. Compared to conventional disk-based

databases, in-memory databases have several benefits, such as quicker data access, real-time analytics, and lower latency. They also perform better for high-speed data processing applications and are simpler to scale. Yet, they may cost more to deploy and need specialist technology to work at their best. Also, there is a chance of data loss if a system malfunctions since data is kept in memory. Some popular in-memory databases include SAP HANA, Oracle TimesTen, MemSQL, and Redis.

Different types of in-memory databases are available, each with its own characteristics and features. Some notable types are explained below.

6.3.5.1 Relational In-Memory Databases

Similar to conventional relational databases, these databases store data in an organised way utilising tables, rows, and columns. However, since all the data is kept in memory, queries can be executed, and transactions can be processed more quickly. Applications that need complicated queries, data consistency, and support for the Atomicity, Consistency, Isolation, Durability (ACID) qualities should use relational in-memory databases.

6.3.5.2 Non-Relational In-Memory Databases

These in-memory databases, sometimes called NoSQL databases, manage unstructured or partially structured data. High scalability, flexibility, and quick data access are all features they offer. Non-relational in-memory databases are frequently utilised to organise enormous amounts of varied data, such as social media data, sensor data, and log files.

6.3.5.3 In-Memory Data Grids

In-memory data grids (IMDGs) provide distributed in-memory processing and storage by distributing data over several nodes in a clustered environment. IMDGs offer high availability, fault tolerance, and scalability by duplicating data throughout the cluster. They are frequently employed in situations requiring distributed caching, real-time analytics, and high-speed data access.

6.3.5.4 In-Memory Analytics Databases

These databases are designed to perform complex analytical queries and data mining tasks. They provide sophisticated analytics features, including machine learning algorithms, statistical analysis, and predictive modelling, all while using the speed of in-memory computing. Applications that need instantaneous data insights and decision-making use in-memory analytics databases.

6.4 Choosing the Right Big Data Storage Solution

To ensure effective and dependable data processing, choosing a suitable big data storage solution necessitates a comprehensive study of different technological elements. Identifying prospective storage solutions that can handle a significant influx of data and future scaling requirements depends on thoroughly understanding the organisation's data requirements, including data volume, velocity, and diversity. It is crucial to carefully evaluate technical features, including data models, query capabilities, and fault tolerance methods, to determine whether they are compatible with the organisation's data processing processes and analytical needs.

Performance is crucial to decision-making; thus, throughput, latency, and data retrieval speed must be thoroughly examined. These indicators make it possible to evaluate a storage solution's capacity to manage workloads requiring a lot of data quickly while providing optimal data processing efficiency. Additionally, thorough cost analyses considering hardware needs, licencing costs, and ongoing maintenance costs enable organisations to match their storage solution preferences with their financial limitations, resulting in cost-effective implementation and long-term operational sustainability.

Data security and compliance issues are crucial in selecting big data storage. Storage systems must conform to strict security requirements to protect sensitive data, including strong encryption technologies, access restrictions, and authentication processes. Compliance with pertinent laws and sector-specific regulations is crucial to safeguard data privacy and fulfil legal duties. Examples include GDPR and HIPAA.

6.4.1 Factors to Consider

Numerous important considerations should be carefully considered while evaluating the critical variables in selecting the best big data storage solution. These elements make sure that the chosen storage solution meets the organisation's unique requirements, scalability requirements, and financial limitations. The considerations to take into account are thoroughly explained in the following:

1. **Scalability**: It is important that the storage system can expand without a hiccup to accommodate the expanding data volume. Ensuring the chosen solution can effectively manage growing data demands requires evaluating scalability features, including horizontal or vertical scaling, data partitioning capabilities, and support for distributed processing frameworks.
2. **Performance**: For efficient data processing, evaluating the storage solution's performance capabilities is essential. Data retrieval speed, throughput, latency, and query response time are important performance indicators. Understanding how the solution manages concurrent data access, indexing, and compression techniques is important to choose a solution that satisfies the organisation's performance needs.

3. **Data Model**: A crucial consideration is whether the data model of the storage solution is compatible with the organisation's data structure. It's critical to assess whether the solution delivers the necessary data manipulation and query capabilities for effective data analysis and whether it supports the essential data types (such as structured, unstructured, or semi-structured data).
4. **Data Security**: When working with sensitive or regulated information, ensuring data security is essential. Encryption, access restrictions, authentication procedures, and auditing tools should all be included in the storage solution's rigorous security features. Compliance with industry-specific rules and standards should also be considered to protect data privacy and fulfil legal obligations.
5. **Cost**: The financial viability of the storage solution must be assessed through a thorough cost study. This study should consider upfront implementation costs, continuing operational expenses, licencing fees, hardware requirements, and maintenance costs. It's critical to balance cost-effectiveness and satisfy the organisation's needs for performance and storage.
6. **Integration and Interoperability**: It is critical to evaluate how well the solution integrates with current data infrastructure and integration capabilities and assess its ability to symbiotically operate with the organisation's technological stack's data processing frameworks, analytics tools, and other systems. APIs, connectors, and data transfer mechanisms should be considered to provide seamless data transmission and interoperability.
7. **Vendor Support and Community**: Assessing the standing and dependability of the storage solution's provider is significant. The vendor's track record, customer service offerings, documentation, updates, and community support should all be considered. A solid vendor support system guarantees prompt issue resolution and assistance during technical difficulties or upgrades.

6.4.2 Scalability and Performance Requirements

Scalability and performance needs are essential when choosing a large data storage system. Scalability is the capacity of a plan to manage increased data volumes and smoothly meet expanding workloads. Performance, however, focuses on how well the solution processes and retrieves data within reasonable periods. Let's examine each of these elements in further detail:

1. **Scalability**: Massive data volumes that can increase quickly over time are common in big data contexts. The chosen storage system must have scalable features to satisfy these changing needs. There are two main scalability factors to think about:

    a. **Horizontal Scalability**: Distributing the data and processing over a bigger cluster refers to the capacity to grow by adding more resources, such as servers or nodes. Faster data processing is made possible by horizontal scalability,

which effectively manages increasing workloads and promotes parallel processing.

b. **Vertical Scalability**: Vertical scalability entails scaling up the current resources, often by increasing hardware components like CPU, memory, or storage capacity to accommodate bigger data volumes and more sophisticated processing tasks. In particular, vertical scaling benefits from tackling single, resource-intensive jobs that demand more processing capacity.

2. **Performance**: The performance of a large data storage solution greatly influences the efficiency and speed of data processing processes. It significantly impacts how well and quickly data can be managed and analysed. Several important criteria must be considered in assessing the performance of a storage solution as they directly affect how well the system handles large-scale data demands. These elements include but are not limited to the following:

   a. **Data Retrieval Speed**: Particularly in real-time analytics settings, the time it takes to access and retrieve data from the storage solution is critical. Faster data retrieval guarantees that information is available when it is needed for analysis and decision-making.
   b. **Throughput**: Throughput is the volume of data that can be processed in a predetermined length of time. Large data volumes can be processed effectively with high throughput, which shortens the time needed for analytics jobs and speeds up the creation of insights.
   c. **Latency**: The time elapsed between a request and the matching answer is called latency. Low-latency storage solutions are essential for instant data access applications, such as real-time analytics or interactive data exploration.
   d. **Query Response Time**: The effectiveness of data analysis is strongly influenced by the speed at which the storage solution can process and reply to queries. Faster exploration and retrieval of insights from the data are made possible by low query response times.

Organisations should consider factors including the anticipated pace of data expansion, the number of concurrent users, the complexity of analytical queries, and the required response time for various data processing jobs when evaluating scalability and performance needs. It is possible to determine how well a storage system satisfies these requirements under different workload conditions by doing performance benchmarks and stress tests.

6.5 Future Trends in Big Data Storage

Big data storage is an area that is continually changing due to technological breakthroughs and the rising demands of data-intensive applications. Several interesting future technologies are reshaping the big data storage market as businesses struggle

with the problems brought on by large data volumes. These themes include developments in storage technology, the rise of edge computing and distributed storage, and the incorporation of AI and machine learning methods. The ability to leverage the full potential of big data and make data-driven choices with increased agility and effectiveness depends on organisations keeping up with these changes. Doing so can open new options for efficient data storage, processing, and analysis.

6.5.1 Advances in Storage Technologies

Technological developments in storage have the potential to substantially influence big data storage solutions as the big data sector continues to develop. Both the hardware and software components of these breakthroughs have been improved significantly. SSDs, non-volatile memory express (NVMe) storage, and storage-class memory (SCM), among other hardware innovations, have boosted storage capacity, improved performance, and decreased latency. Faster data access and retrieval by these technologies increase the overall effectiveness of large data storage systems. On the software side, advancements in data management methods, compression formulas, and data deduplication technologies help to optimise data storage and the more effective use of storage resources. Organisations are now better equipped to manage the ever-increasing amounts of big data with these developments in storage technology.

6.5.2 Edge Computing and Distributed Storage

Edge computing has become more important in the big data environment due to the proliferation of IoT devices and the rising demand for real-time data processing. Edge computing reduces the latency and bandwidth needed for moving data to centralised storage or cloud settings by processing and analysing it closer to its source. Distributed storage systems that support edge computing spread data storage among several local nodes or edge devices. This method makes faster data processing, more scalability, and higher fault tolerance possible. Edge computing and distributed storage enable businesses to process data more quickly, do real-time analytics, and generate less network traffic, which makes them perfect for applications needing low-latency data processing and analysis.

6.5.3 AI and Machine Learning in Storage

Big data storage solutions that incorporate machine learning and AI techniques have the potential to completely change how data is managed, stored, and analysed. Storage systems can use AI and ML algorithms to improve data placement, organisation, and retrieval techniques. With these approaches, intelligent data tiering can be implemented, in which frequently accessed data is automatically transferred to

quicker storage tiers, and less-often accessed data is relocated to less expensive storage alternatives. Additionally, storage performance bottlenecks can be found and optimised using AI and machine learning, as can data replication and backup plans. Organisations can increase data management efficiency and performance, save expenses, and improve decision-making skills by utilising AI and machine learning in storage.

6.6 Learning Outcomes of the Chapter

- **Importance of Storage Systems for Big Data**: Discussing the significance of storage systems in big data.
- **Traditional Storage Systems for Big Data**: Exploring traditional storage systems, including relational databases, data warehouses, NAS, and SAN.
- **Big Data Storage Solutions**: Overviewing various big data storage solutions, such as HDFS, NoSQL databases, cloud storage solutions, object storage systems, and in-memory databases.
- **Choosing the Right Big Data Storage Solution**: Providing insights into factors to consider when choosing a big data storage solution, including scalability and performance requirements.
- **Future Trends in Big Data Storage**: Discussing emerging trends in big data storage, such as advances in storage technologies, edge computing, distributed storage, and integrating AI and machine learning.

References

1. M. Mohania, S. Samtani, J. Roddick, Y. Kambayashi et al., Advances and research directions in data-warehousing technology. Australasian J. Inf. Syst. **7**(1) (1999)
2. T. Andriani, M. Hidayatullah, D. Saputra, S. Esabella, and G. Gunawan, Building data centers using network attached storage (nas) and microprocessor operating systems, in *IOP Conference Series: Materials Science and Engineering*, vol. 1088, 1st edn. (IOP Publishing, 2021), p. 012076
3. V.V. Riabov, *Storage Area Networks (sans)* (Van Nostrand's Scientific Encyclopedia, 2005)
4. K. Shvachko, H. Kuang, S. Radia, R. Chansler, The hadoop distributed file system, in *IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*. (IEEE, 2010), pp. 1–10
5. R.P. Padhy, M.R. Patra, S.C. Satapathy, Rdbms to nosql: reviewing some next-generation non-relational database's. Int. J. Adv. Eng. Sci. Technol. **11**(1), 15–30 (2011)
6. A. El Alami, M. Bahaj, Y. Khourdifi, Supply of a key value database redis in-memory by data from a relational database, in *19th IEEE Mediterranean Electrotechnical Conference (MELECON)*. (IEEE, 2018), pp. 46–51

7. D.J. Abadi, S.R. Madden, N. Hachem, Column-stores vs. row-stores: how different are they really? in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data* (2008), pp. 967–980
8. G. Navarro, R. Baeza-Yates, Proximal nodes: A model to query document databases by content and structure. ACM Trans. Inf. Syst. (TOIS) **15**(4), 400–435 (1997)
9. I. Robinson, J. Webber, E. Eifrem, *Graph Databases: New Opportunities for Connected Data* (O'Reilly Media, Inc., 2015)

Further Reading

10. M. Strohbach, J. Daubert, H. Ravkin, M. Lischka, Big data storage, in *New Horizons for a Data-Driven Economy: A Roadmap for Usage and Exploitation of Big Data in Europe* (2016), pp. 119–141
11. M. Chen, S. Mao, Y. Zhang, V.C. Leung, M. Chen, S. Mao, Y. Zhang, V.C. Leung, Big data storage, in *Big Data: Related Technologies, Challenges and Future Prospects* (2014), pp. 33–49

Big Data Monitoring

7

If you cannot measure it, you cannot improve it.

—Lord Kelvin (1824–1907)

In this chapter, the reader is introduced to the pivotal domain of big data monitoring, delving into the fundamental concepts and tools essential for ensuring the seamless functioning of complex systems. The exploration begins by elucidating the different types of monitoring, namely proactive and reactive, and underscores the critical need for effective monitoring of big data systems. The chapter further dissects the components integral to monitoring, including alerts/notifications, events, logs, metrics, incidence tracking, and debugging capabilities. Providing a comprehensive overview, the narrative then outlines various available monitoring tools tailored for big data systems, ranging from DataDog and SequenceIQ to Sematext, Apache Chukwa, Nagios, Ganglia, DMon, and SmartMonit. By examining these tools, readers gain insights into the diverse functionalities and features contributing to efficient big data monitoring practices.

7.1 Understanding Monitoring

Monitoring is the process of collecting, transmitting, recording, and processing information to monitor the operation of any system to help system administrators make the right decision. Monitoring is a process of understanding the system from the moment it starts, during which it generates data and deciphers the internal behaviour of the processes connected to the problem points. Thus, it also undertakes the task of supervising the working process of a business plan. Keeping records of events helps

Ü. Demirbaga et al., *Big Data Analytics*,
https://doi.org/10.1007/978-3-031-55639-5_7

to gain insight and allows users to predict the course of the next action based on the insight obtained. One of the important purposes of monitoring is to understand how the system works and to look for ways to improve it.

Monitoring in computer science is defined as collecting, analysing, recording in databases and transferring all systems processes within the framework of determined rules. Monitoring can be at software and hardware levels, such as operating systems, database management systems, application software, and computer hardware.

It is almost impossible for systems to operate at 100% performance, and there are many times when they break down or do not operate at the desired optimum level. Monitoring the system is the only way to understand these performance drops and failures. Monitoring a system can help better understand its behaviour pattern and predict any failures before they occur. Monitoring maintenance processes are commonly used to optimise the system behaviour, ensure high utilisation and detect anomalies.

Figure 7.1 demonstrates a basic monitoring workflow for any system and application. Users can analyse the logs collected from multiple sources via either storage or a dashboard.

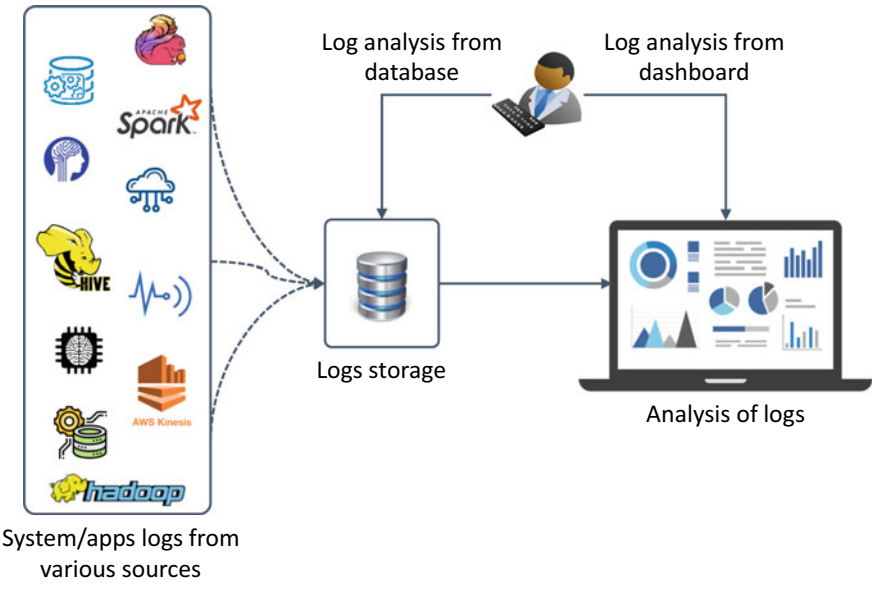


Fig. 7.1 A conceptual workflow of monitoring

7.2 Identifying the Types of Monitoring

In terms of monitoring techniques, there are considerations when designing monitoring solutions that play a crucial role in determining the type and function of the system. These techniques, each of which contains its own characteristics, are two types: Proactive and Reactive.

7.2.1 Proactive Monitoring

Proactive monitoring simply means constantly identifying potential problems before they create major system challenges. Thanks to proactive monitoring, you can predict these problems and develop solutions before a system's malfunction or performance degradation starts. Proactive monitoring helps identify potential issues within IT infrastructure or applications before system users encounter a problem and initiates actions to prevent the problem from affecting system operation. It is beneficial to prevent a system from failing or shutting down completely. Proactive monitoring actively uses system logs and metrics to understand the operating principle of the system it manages. The metrics collected simultaneously during a series of events play a crucial role in predicting the behaviour of this system in the future or giving a notification accordingly.

In this monitoring model, warning signs or signals given before something goes wrong are extremely helpful in preventing failure from the warning signal. For example, when the free disk space on a server falls below a certain threshold, the administrator in charge of the system offers an external disk, preventing the failure of the existing disk or the loss of streaming data.

7.2.2 Reactive Monitoring

Reaction monitoring, often known as aftermath monitoring, is a process of finding and assessing failures after they have occurred. When the warnings of proactive monitoring are ignored and no action is taken to address potential symptoms, reactive monitoring becomes active, signalling that a problem has occurred in the overall or part of the system. For example, in a big data system with heterogeneous hosts, when data is evenly distributed across all servers (high-powerful hosts and less-powerful hosts), proactive monitoring gets active and informs the users that less-powerful hosts will take longer to process data. When data processing begins, outliers will occur on low-powerful hosts. Other hosts, namely high-powerful ones, in the server that complete data processing will wait for these low-powerful hosts to complete their tasks, resulting in a waste of time and energy. At this point, reactive monitoring will detect these outliers and give a warning. In such cases, root cause analysis, a problem-solving method that analyses logs using proactive monitoring techniques, is required to understand why problems occur or the trigger of the problem.

In reactive monitoring, the opposite of proactive monitoring, instead of predicting potential problems and informing the user, it detects the error after a problem has occurred.

7.3 The Need for Monitoring

Even if the components that make up the computer system (i.e. software, hardware) have been tested beforehand, monitoring is still needed after the system is up and running. These reasons are listed below:

- The workload is different from that used in testing the system,
- User errors (i.e. executing a wrong command, configuration error),
- Hardware/network failures during execution,
- The need to detect possible errors before they occur,
- Error detection and troubleshooting,
- Evaluation of the efficiency of the existing system (i.e. effective power utilisation and resource planning),
- Verifying if SLAs[1] are satisfied.

7.4 The Components of Monitoring

Once we deeply understand the types of monitoring and in what situations it is necessary, let's look at the monitoring components.

7.4.1 Alerts/Notifications

It is a message system that informs the user or system administrator about the problem when any malfunction occurs. The specified conditions must be met for configuring a warning or notification. When these conditions occur, the relevant notice is triggered, and the user/system administrator is informed regarding this situation. Different media can be used to send notifications, such as simple message service, email, push notifications for mobile apps, and HTTP Push Events.

[1] SLA is an agreement between a service provider and a customer that defines the types and standards of services to be provided.

7.4.2 Events

In computer systems, an event is an activity recognised and handled by the system and continues in the background to keep the system running. These activities can be triggered by the system itself, by the user, or by other events, and are processed simultaneously with the flow of the programme. For example, the main memory has two main tasks: receiving the data to be processed from the hard disk and transmitting it directly to the processor. These data reading and transmitting operations are events of the main memory.

Each function and method that comprises a software application can be considered an event. It is important in software engineering to follow the output of each event in order to detect errors and eliminate problems in software applications.

7.4.3 Logs

The log is the file created by the system and keeps a record of all its events. It contains the details of each event and a timestamp that assigns a date and time to the event. Software developers use logs to check the system's flow and detect errors in the system. The logs form the basis of the monitoring, providing all the necessary information.

Figure 7.2 shows a screenshot of the logs presented in XML[2] format belonging to a MapReduce job.

The logs, which provide raw data that can be analysed to gain insights from the behaviour of the system, are kept by the organisations for a certain period in terms of security and compliance control of the system. Safe storage of logs containing sensitive information, which can be called the heart of the system, is important in terms of not creating security vulnerabilities.

7.4.4 Metrics

While logs are about a specific event, metrics are the smallest unit of insight from a log made for the system at a specific time. Logs that present data in different formats and structures gain meaning via metrics. Each metric that explains the relevant component has its measurement standard. This unit can be an estimated amount of memory remaining or a percentage of CPU usage. Metrics that present a single piece of information assist the system administrator in making decisions.

[2] Extensible Markup Language.

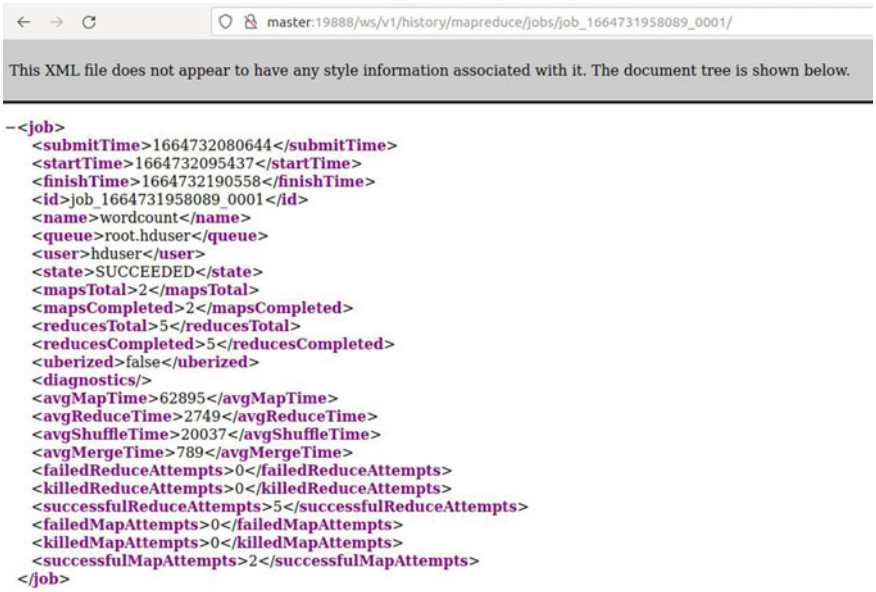


Fig. 7.2 Logs containing the events of a MapReduce job

Table 7.1 presents a part of metrics extracted from MapReduce job logs.

7.4.5 Incidence

An incident is a circumstance, state, or scenario that prevents a system from operating normally. Incidents which indicate an undesirable request condition slow down or even render the system inoperable. A series of incidents such as software/hardware failure, user error, and security vulnerability that occur in a system can trigger other things and cause other events.

Reactive monitoring tools capture incidents, and a warning message is sent to the system administrator to take action. The severity and priority of the incident are determined by levelling the incidents according to the severity level. Incident response, which is how to handle and manage the incident, is crucial in preventing the incident from more severe problems that can lead to significant expense or a system crash. Responding quickly to an incident helps minimise losses, end service disruptions, and reduce other risks in the future.

Table 7.1 Metrics of the job object

Item	Data type	Description
submitTime	Long	The time the job submitted (in ms since epoch)
startTime	Long	The time the job started (in ms since epoch)
finishTime	Long	The time the job finished (in ms since epoch)
id	String	The job id
name	String	The job name
queue	String	The queue the job was submitted to
user	String	The user name
state	String	The job state
mapsTotal	Int	The total number of maps
mapsCompleted	Int	The number of completed maps
reduceTotal	Int	The total number of reduces
reduceCompleted	Int	The number of completed reduces
uberized	Boolean	Indicates if the job was an uber job
diagnostics	String	A diagnostic message
avgMapTime	Long	The average time of a map task (in ms)
avgReduceTime	Long	The average time of the reduce (in ms)
avgShuffleTime	Long	The average time of the shuffle (in ms)
avgMergeTime	Long	The average time of the merge (in ms)
failedReduceAttempts	Int	The number of failed reduce attempts
killedReduceAttempts	Int	The number of killed reduce attempts
successfulReduceAttempts	Int	The number of successful reduce attempts
failedMapAttempts	Int	The number of failed map attempts
killedMapAttempts	Int	The number of killed map attempts
successfulMapAttempts	Int	The number of successful map attempts

7.4.6 Debugging Ability

Debugging is a multistep process that covers diagnosing and locating the root cause of the problem for fixing or eliminating it. To fully understand the bottleneck of the system, the system administrator needs a tool that makes it easy to access the system logs and monitor the flow of the application, as well as interpret them. Finding and resolving the error in the application is impossible without interpreting the system logs. To this end, it is crucial that the monitor tool has the debugging capability and presents it effectively.

7.5 Available Monitoring Tools for Big Data Systems

Monitoring tools are used for a number of reasons, such as noticing when things go wrong, debugging, gaining insights, sending data/notifications to other systems, and controlling capital expenditure to run cloud infrastructure.

There are some essential criteria listed below to keep in mind when choosing a system for viewing big data systems:

- How easy is the system to install and configure?
- Is it available as open source? If not, how much is the license fee?
- Can custom add-ons be made?
- How much is the resource load caused by system components?
- Is there a user interface?
- Does it provide data visualisation?

Both businesses and academia have developed various tools widely used for monitoring big data systems. Let's have a look at these tools in detail.

7.5.1 DataDog

Datadog[3] is a SaaS-based observability service for cloud-based big data systems, which allows users to monitor the information, such as servers, databases, and applications. It can also monitor cloud infrastructure, hosts (i.e. Windows, Linux), serverless stack, and cloud-based applications. Moreover, DataDog is used for various purposes, including managing logs, exploring metrics, and data visualisation.

Datadog collects logs from large-scale Hadoop clusters by deploying a DataDog agent on each running node in the cluster and presents these logs to the user in real time via the dashboard. It collects information on each process and system infrastructure information, such as CPU/memory utilisation and network status. It has an alert mechanism that gives a warning when the threshold the user sets is exceeded. Datadog has some technologies supporting various cloud platforms such as AWS,[4] Azure,[5] Google Cloud,[6] Kubernetes,[7] Red Hat OpenShift,[8] Pivotal Platform,[9] and provides network and security monitoring for these systems. Datadog shows outliers in big data systems but cannot perform root cause analysis for these cases.

Figure 7.3 shows the metrics from HDFS, such as HDFS disk capacity, total load by NameNode, and YARN, such as virtual core usage, pending/running tasks, while Fig. 7.4 demonstrates ZooKeeper metrics, such as requests, the status of connections, bytes sent/received, along with Java Virtual Machine (JVM) metrics, namely ParNew time and JVM heap usage [1].

[3] https://www.datadoghq.com/.
[4] https://aws.amazon.com/.
[5] https://azure.microsoft.com/.
[6] https://cloud.google.com/.
[7] https://kubernetes.io/.
[8] https://www.redhat.com/en/technologies/cloud-computing/openshift.
[9] https://docs.pivotal.io/.

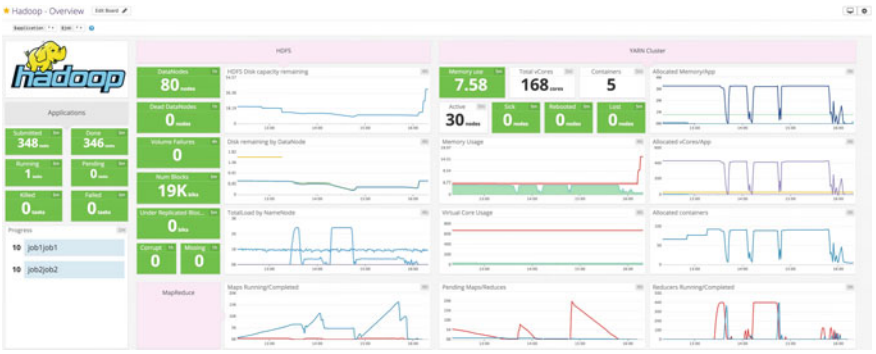


Fig. 7.3 Visualisation Hadoop metrics

Fig. 7.4 Visalization ZooKeper and JVM metrics

7.5.2 SequenceIQ

SequenceIQ,[10] licensed under Hortonworks/Cloudera [2], was developed by Hortonworks company to monitor Apache Hadoop distribution. This software, whose architecture is built on the Elasticsearch, Logstash, and Kibana (ELK) stack, uses Docker technology that provides isolation between monitoring tools and Hadoop components so that new components can be easily added to and removed from the system. The architecture consisting of different layers enables the collection of logs of each process of big data systems and cloud-based IT infrastructure. It also provides tracking based on the ELK stack. It monitors CPU and memory usage-dependent applications by applying auto-scaling to YARN, Hadoop's resource management and job scheduling technology. SequenceIQ assists in cost measurement by allowing users to examine Hadoop MapReduce applications in detail. Although this moni-

[10] https://github.com/sequenceiq.

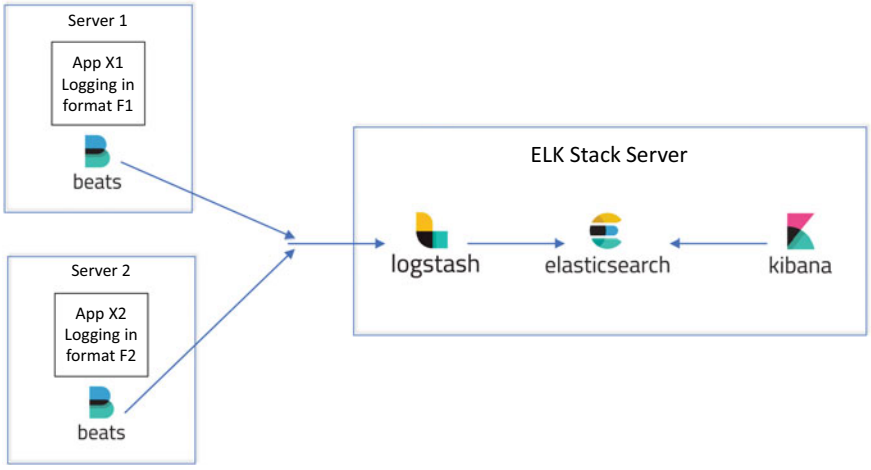


Fig. 7.5 ELK Stack architecture

toring system efficiently collects and stores logs from big data systems, it cannot perform root cause analysis and investigation of any failure.

Figure 7.5 depicts simply ELK stack architecture. Beats are a family consisting of lightweight data transporters, namely, filebeat, metricbeat, packetbeat, winlogbeat, auditbeat, and functionbeat, deployed on each host/server. Logstash is an open-source log aggregator that gathers data from a variety of sources and sends it to one or more locations for further steps, i.e. stashing or storage. Kibana, open-source analysis, and visualisation layer, runs on top of Elasticsearch and Logstash, generally used to visualise logs stored in Elasticsearch. It provides a dashboard with various interactive charts that enable the analysis and visualisation of large-scale data.

7.5.3 Sematext

Sematext [3] is a real-time monitoring and anomaly detection system for big data systems that can be easily integrated into systems such as Hadoop, Kafka, Cassandra, and Elasticsearch. While detecting slow tasks, nodes, and anomalies, Sematext has a mechanism that alerts users via notification when thresholds set by users are reached (for example, high CPU/memory consumption). It also has a graphical interface that provides visualisation of various tasks of big data systems. Although it offers excellent advantages in monitoring and anomaly detection, it cannot perform root cause analysis of anomalies and performance problems.

Figure 7.6 shows the screenshot of the user interface of Sematext monitoring tool, which provides an overview regarding JVM metrics, garbage collection details of the JVM, CPU, memory, disk, I/O, swap, and network traffic status.

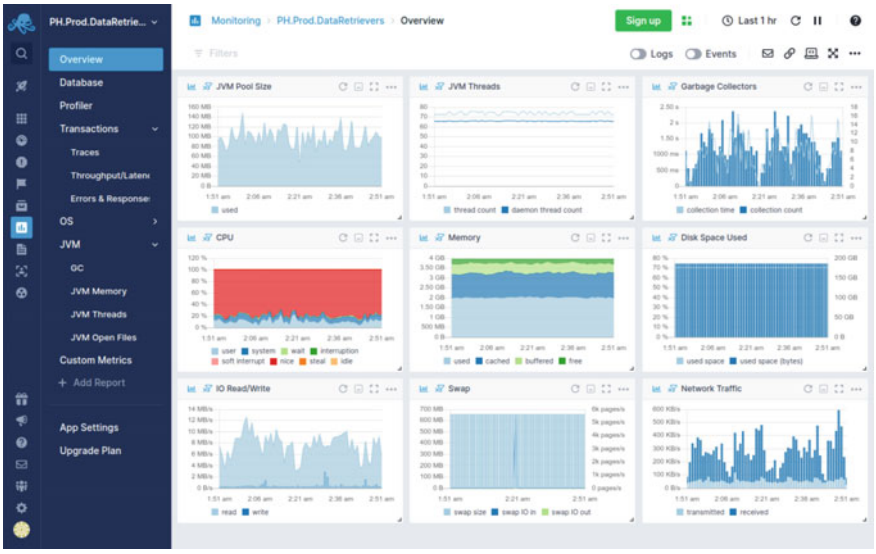


Fig. 7.6 Sematext visualisation interface

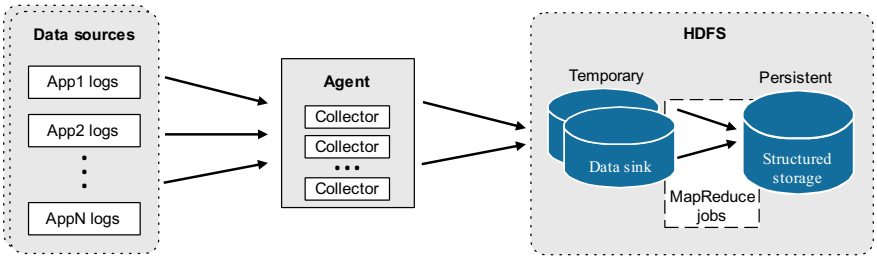


Fig. 7.7 Apache Chukwa architecture

7.5.4 Apache Chukwa

Apache Chukwa [4] is another log collection tool used to monitor large-scale distributed systems built on top of MapReduce and HDFS. Yahoo developed it and then donated it to the Apache Software Foundation under the Apache 2.0 license. Chukwa is a distributed system for gathering, combining, and analysing massive volumes of data generated by big data systems, which consists of multiple components, namely data collector, agents, and adaptor. The agent is responsible for collectors that are distributed to each worker node. Collectors collect the logs and send them to the adaptor. Once adaptors receive the collected logs, they sink them into HDFS.

Figure 7.7 demonstrates the architecture and design of Apache Chukwa. The collectors managed by the Agent collect log data from different data sources/applications. These logs are sunk into temporary storage via adaptors. Afterwards, MapReduce jobs are executed this data in a distributed manner to be stored in HDFS as persistent.

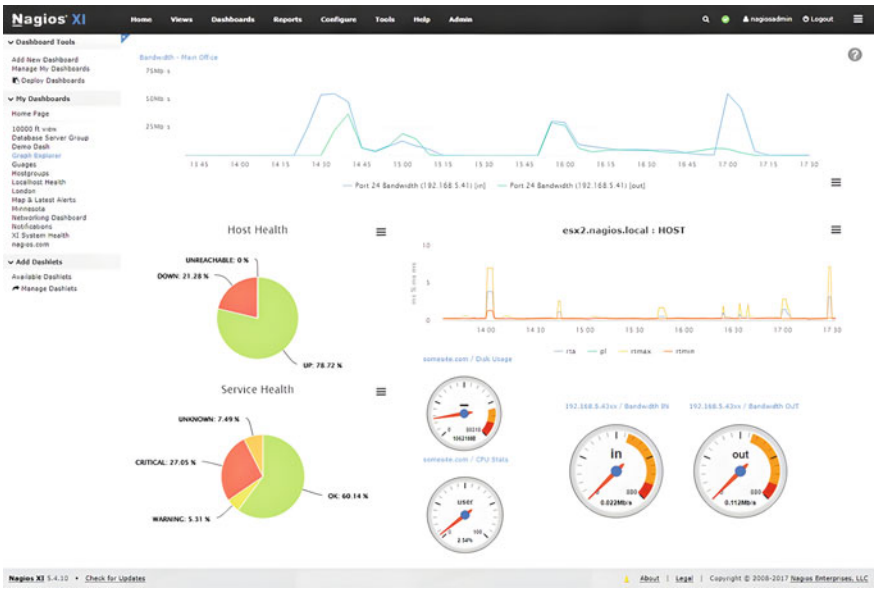


Fig. 7.8 Nagios user interface

7.5.5 Nagios

Nagios [5] is an open-source framework that aims for businesses to keep track of their network and IT infrastructure. System administrators and IT teams worldwide now often utilise it since Ethan Galstad first created it in 1999. Nagios employs a client-server design, with the server, Nagios Core, in charge of keeping track of hosts and services and notifying users of problems. Installed on the hosts under observation, the clients (also known as plug-ins) provide updates about their status to the server.

Servers, switches, routers, applications, and services can all be monitored by Nagios. It can check for CPU and memory use, storage space, network connectivity, and web server accessibility. Since Nagios is adaptable and adjustable, administrators may set their monitoring thresholds and rules, make custom notifications, and interface Nagios with other tools and systems. A sizable user and developer base has produced many plug-ins and extensions to expand its functionality. Figure 7.8 shows a part of the Nagios user interface demonstrating advanced graphs.

7.5.6 Ganglia

Ganglia [6] is an open-source distributed monitoring solution for big data and high-performance computing systems like clusters and grids. It was first created at the University of California, Berkeley, in 2000 and is currently utilised extensively in academic, industrial, and governmental research labs. Ganglia can monitor massive

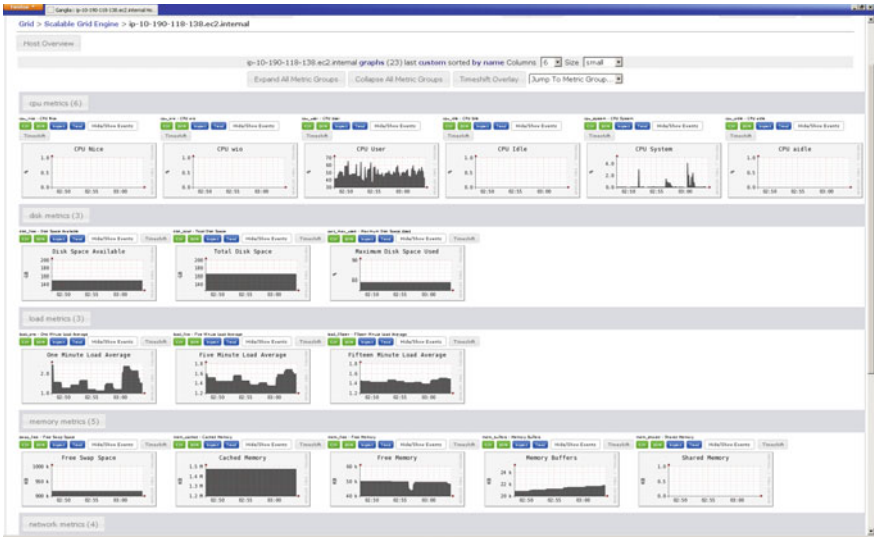


Fig. 7.9 User interface of Ganglia

clusters with thousands of nodes since it is built to be very scalable. Three primary parts make up its hierarchical architecture: the Ganglia web interface, the Ganglia monitoring daemon (gmetad), and the Ganglia metadata daemon (gmond).

On each node in the cluster, the gmond daemon runs and gathers system statistics, including CPU and memory consumption and network traffic. The gmetad daemon then aggregates these data, giving a uniform picture of the cluster's condition. With the Ganglia web interface, administrators may create reports, examine the cluster's status, and create individual alarms. Ganglia has a plug-ins feature that allows administrators to monitor metrics and services beyond those already included. Ganglia is a versatile and configurable monitoring solution because it offers various data visualisation tools and third-party interfaces. Data collection daemons and processes in Ganglia are demonstrated in Fig. 7.9.

7.5.7 DMon

DMon [7] is an open-source distributed monitoring platform developed for cloud-based big data systems as part of the Data-Intensive Cloud Applications with iterative quality enhancements (DICE) project. Large-scale distributed systems like Hadoop, Spark, Flink, and other big data platforms are the focus of DMon's monitoring capabilities. It uses a modular design and has several parts, including a user interface, a data processor, a data aggregator, and a data collector.

The data collector part of the distributed system gathers system metrics from every node and provides them to the data aggregator. Metrics are then combined by the data aggregator and sent to the data processor, who examines the data and issues warnings

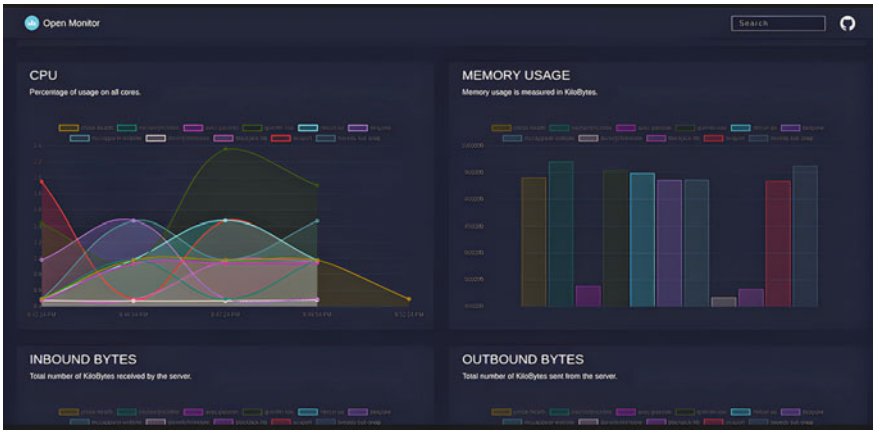


Fig. 7.10 DMon monitoring system user interface

if any abnormalities are found. The user interface offers administrators a dashboard to check the system's status, create reports, and create specific alarms. DMon's support for custom metrics, which enables administrators to track mainly distributed system components not covered by the built-in metrics, is one of its essential features. DMon is a very adaptable and configurable monitoring solution for big data platforms since it also supports several data visualisation tools and third-party interfaces. Figure 7.10 shows a part of the monitoring components of DMon monitoring system.

7.5.8 SmartMonit

SmartMonit [8] is a novel big data monitoring system developed for collecting comprehensive logs identifying big data tasks and infrastructure information from big data systems. It consists of three main components, Information Collection, Computation and Storing, and Visualisation [9].

A large-scale computer cluster's nodes' resource utilisation and job and task metrics are collected in real time using SmartAgent and Agent as part of information collection. The SmartAgent is installed on the master node and uses the TaskTrackers and Yarn APIs to gather data from the DataNodes about the tasks (mappers and reducers), applications (job details), and the cluster. Moreover, the System Information Gatherer and Reporter (SIGAR) library[11] is connected to the SmartAgent to track how well the master node's resources—including CPU, memory, and network bandwidth—are used. The Execution Graph may be constructed using the process data that the SmartCollector collects. The Computation and Storage module receives a stream of all monitoring data. In Computation and Storing, the monitoring data

[11] https://github.com/hyperic/sigar.

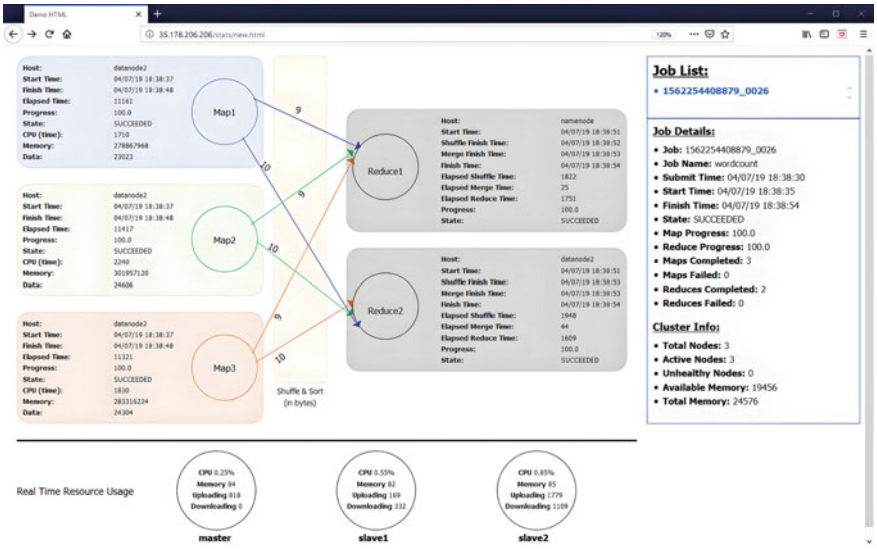


Fig. 7.11 SmartMonit execution graph

received from the cluster is gathered using a RabbitMQ Server[12] which is open-source message broker system offering high throughput, low latency, and reliable communication across applications. The SmartWriter then analyses the data gathered from the RabbitMQ Server and stores the results in the open-source time-series database InfluxDB. Time series data, including data from operations monitoring, sensor data, and application metrics, can be stored in this database and retrieved with high availability. The query engine and user interface are both included in the visualisation. To create the Execution Graph, the query engine runs queries against the database over a predetermined period. The Execution Graph and other monitored data aid human operators in quick comprehension and diagnosis.

SmartMonit gathers run-time system data, such as information on job execution and computer resource use. The ability to interpret the information collected and create a dynamic Execution Graph for each application while simultaneously displaying the graph in real time is crucial. The most important feature of SmartMonit is to provide an infrastructure for building a real-time debugging system for big data systems. Figure 7.11 shows the real-time execution graph of SmartMonit deployed in AWS.

[12] https://www.rabbitmq.com/.

7.6 Learning Outcomes of the Chapter

- **Understanding Monitoring**: Exploring the concept of monitoring in the context of big data.
- **Identifying the Types of Monitoring**: Distinguishing between proactive and reactive monitoring.
- **The Need for Monitoring**: Discussing the importance and necessity of implementing monitoring in big data systems.
- **The Components of Monitoring**: Describing key monitoring components, including alerts/notifications, events, logs, metrics, incidence, and debugging ability.
- **Available Monitoring Tools for Big Data Systems**: Introducing various monitoring tools for big data systems, such as DataDog, SequenceIQ, Sematext, Apache Chukwa, Nagios, Ganglia, DMon, and SmartMonit.

References

1. Datadog. Accessed: 2022-10-11. [Online]. Available: https://www.datadoghq.com/
2. S. Qanbari, A. Farivarmoheb, P. Fazlali, S. Mahdizadeh, S. Dustdar, Telemetry for elastic data (ted): Middleware for mapreduce job metering and rating, in *2015 IEEE Trustcom/BigDataSE/ISPA*, vol. 2. (IEEE, 2015), pp. 104–111
3. Sematext. Accessed: 2022-10-12. [Online]. Available: https://sematext.com/
4. Apache chukwa. Accessed: 2022-10-12. [Online]. Available: https://chukwa.apache.org/
5. Nagios. Accessed: 2022-10-15. [Online]. Available: https://www.nagios.org/
6. Ganglia. Accessed: 2022-10-17. [Online]. Available: http://ganglia.info/
7. Dmon. Accessed: 2022-10-17. [Online]. Available: https://github.com/Open-Monitor/dmon
8. U. Demirbaga, A. Noor, Z. Wen, P. James, K. Mitra, R. Ranjan, Smartmonit: Real-time big data monitoring system, in *2019 38th Symposium on Reliable Distributed Systems (SRDS)*. (IEEE, 2019), pp. 357–3572
9. W. Wen, U. Demirbaga, A. Singh, A. Jindal, R. S. Batth, P. Zhang, G. S. Aujla, Health monitoring and diagnosis for geo-distributed edge ecosystem in smart city. IEEE IoT J. **10**(21), 18 571–18 578 (2023)

Further Reading

10. G. Singh, *Monitoring Hadoop*. (Packt Publishing Ltd, 2015)
11. E. Diagboya, *Infrastructure Monitoring with Amazon CloudWatch: Effectively Monitor Your AWS Infrastructure to Optimize Resource Allocation, Detect Anomalies, and Set Automated Actions*. (Packt Publishing Ltd, 2021)
12. G. da Cunha Rodrigues, R.N. Calheiros, V.T. Guimaraes, G.L.d. Santos, M.B. De Carvalho, L.Z. Granville, L.M.R. Tarouco, R. Buyya, Monitoring of cloud computing environments: concepts, solutions, trends, and future directions, in *Proceedings of the 31st Annual ACM Symposium on Applied Computing* (2016), pp. 378–383

Debugging Big Data Systems for Big Data Analytics

8

Debugging is like being the detective in a crime movie where you are also the murderer.

—Filipe Fortes

This chapter unveils the intricate art of debugging big data systems for optimal analytics performance, providing a comprehensive guide to navigating real-world performance challenges. The exploration commences by delineating the critical debugging steps essential for identifying and resolving issues within big data systems. Focussing on the specific problems that can afflict these systems, such as data locality, resource heterogeneity, network issues, resource over-allocation, unnecessary speculation, and poor scheduling policies, the chapter dives into the intricacies of root cause analysis. Emphasising the importance of this analysis in the context of big data analytics, the narrative elucidates the systematic steps involved, accompanied by insightful details on tools and techniques, challenges, and considerations. The chapter explores available diagnosis tools tailored for big data systems, including Mantri, Texas Advanced Computing Centre (TACC) Stats, Data Centre Data Base (DCDB) Wintermute, and AutoDiagn, empowering practitioners to effectively diagnose and address complex issues in their analytics infrastructure.

8.1 Debugging for Real-World Performance Problems

Debugging is a multistep process including identifying, isolating, and fixing errors or defects within computer programmes, software, or systems. Big data processing systems consist of a wide range of software and hardware components and often run in large-scale, highly concurrent, and multi-tenant environments, causing frequent hardware and software failures, which makes it challenging to find the root cause

Ü. Demirbaga et al., *Big Data Analytics*,
https://doi.org/10.1007/978-3-031-55639-5_8

of the problem. Strong systems administration, data analysis, software development abilities, and a solid grasp of the system architecture are necessary. Moreover, automated tools and monitoring systems can facilitate debugging by sending real-time warnings and messages when bugs occur.

8.2 Debugging Steps

The following stages are commonly involved in the process of debugging big data systems depicted in Fig. 8.1:

- **Identify the error**: Defining the issue is the first step in debugging big data systems, which entails gathering data regarding the problem, including logs, performance metrics, and error messages. Some reasons make this step essential, such as incomplete or inaccurate data being processed, hardware or network issues, software bugs, and configuration errors.
- **Find the error location**: When the error is discovered correctly, it is necessary to review the system repeatedly to find the location of the fault. This step generally focuses on finding the error rather than detecting the error.

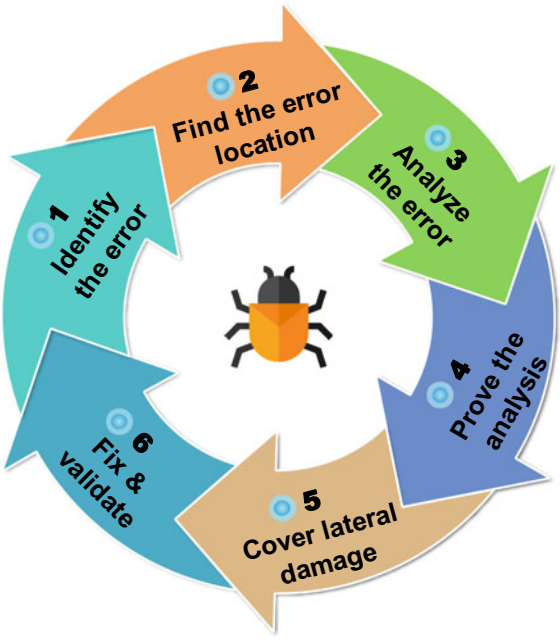


Fig. 8.1 Basic debugging steps in computer systems

- **Analyse the error**: This step covers analysing the error in detail, starting with the error itself and including its associated components. This step facilitates understanding errors and focuses on re-evaluating errors to find existing ones.
- **Prove the analysis**: After the error analysis, in this step, where the search for extra errors that may arise in the system is done, the system is put to the test and checked for errors.
- **Cover lateral damage**: This phase involves collecting tests for the updated part, including the system. All units must pass the tests undergone.
- **Fix & validate**: This is the final stage, where the fix and validation are done. Units that pass the tests are checked for accuracy and then approved.

8.3 Problems in Big Data Systems

8.3.1 Data Locality

Data locality refers to relocating computation near the actual data location rather than relocating massive data to the computation [1]. This reduces traffic on the entire network. Moreover, the system's total throughput is raised by doing this. In distributed systems, data is stored in blocks. It is copied to more than one node or cluster. However, since every data block is not in every node, the data block to be processed is transferred to the node where the processing will be executed since computation power is available in the cluster, resulting in slow query times, increased network traffic, and poor system performance.

Big data systems, such as Hadoop, employ several methods to maximise data locality to solve data locality problems. Data partitioning is a popular strategy that includes breaking up enormous datasets into manageable units called partitions. Processing resources are allocated to the node or cluster where the data is stored, and each partition is kept on a particular node or cluster. As a result, less data must be sent across the network, which enhances system performance and lessens network congestion. Data replication is a different strategy adopted by Hadoop that produces numerous copies of the data across several nodes or clusters. As a result, even if one node or cluster fails, data is always available and accessible. By enabling processing resources to retrieve data from the closest accessible node or cluster, replication can also enhance data locality.

Apache Spark deploys Resilient Distributed Datasets (RDDs) to ensure data locality. RDDs are divided data structures that may be cached across a group of nodes in memory. Spark's processing engine is created by allocating jobs to nodes with the required data stored in memory to maximise data locality.

8.3.2 Resource Heterogeneity

Resource heterogeneity has many advantages in large-scale data systems but must be carefully planned, managed, and monitored to prevent drawbacks [2]. Organisations should consider their unique demands and limits when implementing a heterogeneous system. They must also invest appropriately in resources, expertise, and technologies to operate the system successfully. The disadvantages of resource heterogeneity for big data systems are listed below:

- **Complexity**: Managing a diverse group of resources can be tricky since they may have varied configurations, capacities, and capabilities. Significantly if the system scales up or down over time, it can be challenging to administer, monitor, and debug.
- **Increasing the cost**: A wide variety of resources can be expensive to acquire and keep up with, mainly if people with varying experience levels manage them or originate from separate suppliers. The specialised software, middleware, or hardware requirement to support the various resources may also result in higher operating costs for heterogeneous systems.
- **Compatibility**: Integrating resources from several suppliers or sources might be complex because they could utilise various protocols, formats, or interfaces, resulting in compatibility problems, inconsistent data, or performance issues.
- **Resource allocation**: It can be challenging to allocate resources in a diverse system because various resources may have varying consumption rates, priorities, or limits. Inadequate performance, inefficient use of resources, or an unbalanced task allocation may result from this.

8.3.3 Network Issues

Networks that enable data transfer, scalability, speed, fault tolerance, and cooperation are essential for the success of big data systems. Companies may guarantee that their big data systems perform successfully and efficiently and maximise their data value by investing in network optimization, administration, and monitoring. However, big data systems rely on network-based distributed architectures and frequently experience network issues. Common network problems in big data systems are listed below:

- **Network congestion**: In a distributed system, network congestion that results in longer query times, higher latency, and worsened system performance can happen when data is moved between nodes or clusters. Systems with extensive data or complex processing pipelines may find this especially difficult.
- **Network failures**: Failures or outages in networks can prevent node or cluster connectivity, processing, or data transmission. Depending on the degree and length

of the failure, this might result in data loss, system downtime, or performance deterioration.
- **Cyber security vulnerabilities**: Security risks, including virus attacks, illegal access, and data breaches, can put networks at risk. These dangers can seriously harm or lose the system and endanger data availability, confidentiality, or integrity.

8.3.4 Resource Over-Allocation

A significant issue in big data systems is resource over-allocation, which can happen when more resources (such as CPU, memory, or storage) are assigned to a task or job than are needed [3]. It stems from various factors, such as inaccurate resource requirements, static resource allocations, inefficient resource utilisation, and lack of monitoring and optimization.

Resource over-allocation poses some problems listed below for efficient big data processing:

- **Waste of resources**: Over-allocation of resources can result in squandered or inefficiently used resources, increasing expenses, decreased effectiveness, and longer processing times.
- **Performance reduction**: Over-allocating resources can also result in performance degradation because they may be distributed too thinly over several activities or processes, which causes processing times to take longer or cause more delay.
- **Scalability limitations**: Over-allocation may be constrained by big data systems' capacity to scale since adding more nodes or clusters could be more challenging if existing resources are not being utilised effectively.

Some strategies can be applied to overcome resource over-allocation problems in big data systems, such as enhanced resource allocation management, capacity planning, monitoring and tracking resource utilisation, and resource optimization to ensure optimal system performance and efficiency.

8.3.5 Unnecessary Speculation

During the execution, tasks may get slow for various reasons, including hardware corruption or software misconfiguration. When tasks are still not completed successfully after more than expected, big data systems like Hadoop create and run the same tasks on different nodes instead of detecting the reasons. This is called speculative execution in Hadoop [4]. However, while speculative execution can increase efficiency in big data systems, it can also lead to additional resource consumption and overhead, resulting in excessive resource consumption and increased waiting time for subsequent tasks. Figure 8.2 depicts speculative execution workflow in the Hadoop ecosystem.

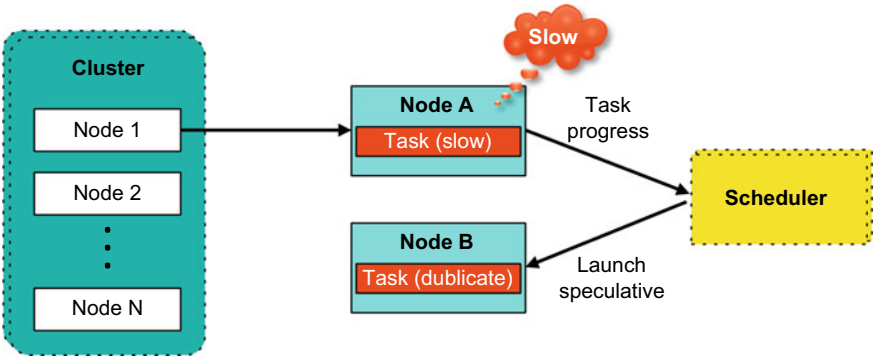


Fig. 8.2 Speculative execution workflow in Hadoop

Speculative execution in Hadoop is enabled by default. To disable this options, the configuration file, *mapred-site.xml*, can be edited:

```
<property>
    <name>mapred.map.tasks.speculative.execution</name>
    <value>false</value>
</property>
<property>
    <name>mapred.reduce.tasks.speculative.execution</name>
    <value>false</value>
</property>
```

8.3.6 Poor Scheduling Policy

Scheduling, which includes allocating and managing computing resources like CPU, RAM, and storage to various activities and processes operating in the system, is a crucial part of big data systems. Resource optimization and timely, effective completion of tasks and projects are the primary objectives of scheduling. A centralised scheduler that controls resource allocation throughout the cluster commonly handles scheduling in big data systems. The scheduler employs algorithms and policies to decide how resources should be distributed and divided among various tasks and jobs, considering task and job priority, resource availability, and workload characteristics. A centralised scheduler that controls resource allocation throughout the cluster commonly handles scheduling in big data systems. The scheduler employs algorithms and policies to decide how resources should be distributed and divided among various tasks and jobs, considering task and job priority, resource availability, and workload characteristics. First In First Out (FIFO) scheduler, capacity scheduler, and fair scheduler are the standard scheduling policies used in big data systems. Figure 8.3 depicts the working principle of the default scheduling policies in the Hadoop ecosystem.

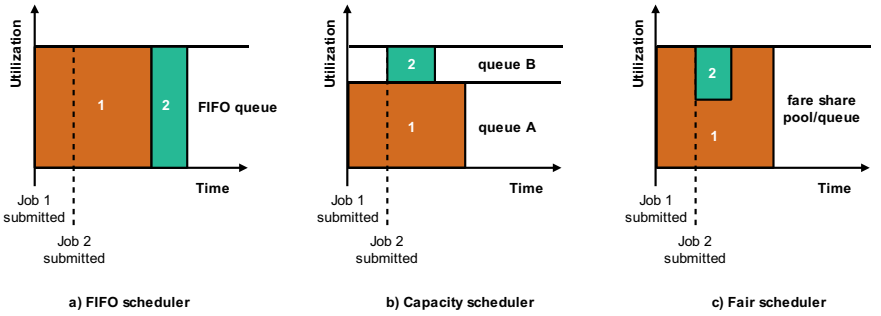


Fig. 8.3 Schedulers in Hadoop

Proper scheduling is essential for the efficiency and scalability of big data systems since it may significantly influence resource consumption, reaction time, and system throughput. Because of this, it's critical to utilise scheduling policies and algorithms that are appropriate for the system's workload and unique requirements. Scheduling policies control the distribution and sharing of computing resources (such as CPU, memory, and disk) across various tasks and jobs in the system. Poor scheduling practices can lead to resource conflicts, bottlenecks, and lengthy user wait times, which reduces system throughput and efficiency. Here are some scheduling policies causing performance reduction at some point:

- **First-come, first-served (FCFS) scheduling**: With this policy, resources are distributed according to this guideline to tasks in the order they are received. The system may perform poorly overall due to the lengthy wait times for more extensive or challenging activities.
- **Greedy scheduling**: This approach devotes as many resources as possible to a single task or job to maximise performance. Nevertheless, this may lead to resource overuse and conflict, lowering system throughput and efficiency.
- **Capacity scheduling**: This scheduler needs some overhead to control resource distribution and enforce scheduling rules. This penalty can lower the total system throughput and raise delays in circumstances with many loads. Moreover, it is not well-suited for real time or interactive workloads that require more dynamic resource allocation.

8.4 Root Cause Analysis in Big Data Systems

Root Cause Analysis (RCA) is a systematic approach used in many fields to pinpoint the root causes of issues or failures. RCA is essential for debugging and troubleshooting sophisticated analytics procedures in the context of big data platforms. Large-scale data processing and analytics frequently span distributed infrastructures

and use various technologies. The size and complexity of the underlying infrastructure of big data systems make it difficult to identify the source of problems or breakdowns. In these complex situations, RCA for big data systems is a specialised field that seeks to pinpoint the key causes of failures, performance bottlenecks, or data discrepancies. Organisations can improve the reliability and accuracy of their data analytics processes by implementing effective remedial methods, optimising system performance, and identifying the underlying issues.

8.4.1 Importance of Root Cause Analysis in Big Data Analytics

RCA holds significant importance in big data analytics due to data-driven systems' inherent complexity and scale. Data must be processed and analysed in big data analytics to draw useful conclusions and guide well-informed choices. Nevertheless, when problems, errors, or performance snags appear, the dependability and accuracy of these analytics operations may be jeopardised. For businesses to properly address these issues and guarantee the accuracy and reliability of their data analytics findings, RCA is crucial in determining the root causes of such problems. Organisations may reduce risks, boost performance, and increase the overall efficacy of their analytical processes by implementing RCA approaches to big data analytics platforms.

8.4.1.1 Ensuring Accuracy and Reliability

In big data analytics, accurate and reliable insights are essential for making well-informed decisions. By locating and resolving the root causes of problems that might jeopardise the integrity of the data, RCA contributes to the accuracy and dependability of analytics outputs. Organisations may fix underlying issues and avoid recurrence by comprehending the core causes. As a result, there is a decrease in the likelihood of erroneous or deceptive analytics results and improved data quality. By fostering confidence in the insights produced and encouraging data-driven decision-making across various disciplines, including business, health care, finance, and more, RCA helps improve the trustworthiness of big data analytics systems.

8.4.1.2 Optimising Performance and Efficiency

Big data systems are renowned for being dispersed and complicated, with many interrelated parts and technologies. Due to the complex interdependencies within these systems, it can be difficult to pinpoint the underlying causes of problems. RCA allows businesses to identify the causes of performance bottlenecks or breakdowns, improving system effectiveness. By addressing the underlying issues, companies may remove bottlenecks, speed up data processing, and increase system productivity. As resources are used effectively and the performance of the big data analytics infrastructure is maximised, this results in greater operational efficiency and cost-effectiveness.

8.4.1.3 Driving Continuous Improvement

RCA is crucial in promoting the ongoing development of big data analytics processes. Organisations can learn from past occurrences and put preventative measures in place to lessen the chances of similar difficulties in the future by methodically identifying and resolving the core causes of crises. Organisations may use RCA to learn more about their big data systems' dependencies, vulnerabilities, and failure patterns. The capacity to improve system robustness, stability, and resilience is given to them by this understanding. As organisations document and share their results, RCA promotes a culture of learning and information sharing, promoting a general grasp of the problems and potential solutions in big data analytics.

8.4.2 Root Cause Analysis Steps

The foundations of RCA serve as the foundation for a systematic and thorough investigation process that aims to pinpoint and address the root causes of issues or failures in big data systems. RCA is a thorough investigation of all the variables and interconnections contributing to the manifestation of problems to identify the primary root cause or reasons behind the observed effects. RCA helps researchers and practitioners gain a profound knowledge of the underlying causes of issues, resulting in efficient remedial actions and preventative measures. This insight is made possible by adhering to essential principles, adopting an organised strategy, and using suitable methodologies. However, it is essential to recognise and address the common challenges and limitations encountered during the analysis process to ensure its accuracy and reliability.

To systematically identify the root causes of issues or failures in big data systems, the RCA approach entails several clearly defined processes. Each stage adds to the entire process by giving it organisation, direction, and clarity. Conducting a complete and efficient RCA requires understanding and adherence to these procedures. The following are the steps in RCA [5,6]:

8.4.2.1 Problem Identification

The first phase in RCA, sometimes called problem identification, entails fully grasping the problem or failure that calls for study within the context of big data systems. It includes clearly describing and articulating the issue by looking at the symptoms, comprehending how they affect the system's operation, and establishing definite analysis goals. To complete this phase, you must gather pertinent data and information about the issue, consult user comments and feedback, examine system logs and analytics, and take into account any occurrences that have been recorded. Problem identification aims to keep the inquiry narrowly focussed on the current issue by laying a strong basis for RCA in later stages. Stakeholders can effectively allocate resources, choose the investigation parameters, and create a focussed strategy for identifying the root causes by describing the problem explicitly.

8.4.2.2 Data Collection

One of the most important steps in RCA is data collection, which entails acquiring a wide variety of pertinent data and information from numerous sources within big data systems. This procedure methodically locates and obtains data from user reports, incident records, monitoring tools, system logs, and other relevant sources. The information gathered should include various topics, such as system performance measurements, error logs, user interactions, network traffic, and additional information to shed light on the current issue. As the basis for further analysis and research, ensuring the data gathered is reliable and representative. Determining the best sample strategies, developing data collection processes, and using cutting-edge tools and technologies to extract, convert, and load the data for analysis are all necessary for effective data collection. Analysts can gain a holistic understanding of the system's behaviour and performance by methodically collecting complete and trustworthy data, enabling them to precisely identify the problems' underlying causes.

8.4.2.3 Data Analysis

The data analysis stage of RCA comprises a thorough and systematic investigation of the gathered data to draw out significant patterns and insights in the context of big data systems. Numerous analytical approaches, statistical methodologies, and data visualisation tools must be used to examine, modify, and analyse the data. Data analysis is to find patterns, correlations, anomalies, and possible causal links that might reveal the root causes of the problem being studied. Descriptive statistics and data profiling are two examples of exploratory data analysis approaches that analysts may use to get a basic grasp of the properties and distributions of the data. Regression analysis, correlation analysis, and machine learning algorithms are examples of sophisticated analytical techniques that may be used to find hidden patterns and correlations in data. Data visualisation is essential in this stage because it makes it easier to analyse large, complex datasets by presenting them in an understandable visual format. Analyzers can get important insights and create hypotheses about probable underlying causes contributing to the observed problem by digging deep into the data, opening the door for more research and analysis.

8.4.2.4 Root Cause Identification

The identification is a major step in RCA that entails a thorough and systematic study of the data, evidence, and information gathered to identify the underlying causes of the issue in big data systems. This multidimensional approach involves a thorough review of the data that is available, the use of investigation tools, and the use of logical reasoning to identify the main root cause or factors that are accountable for the observed problem. To dive deeply into the system's complexity and trace the sequence of events leading to the problems, researchers use a variety of approaches, such as the "5 Whys" technique, fishbone diagrams, fault tree analysis, or causal loop diagrams. Analysts work to pinpoint the fundamental causes of an issue, whether technical, operational, or organisational, by thoroughly investigating causal links and

dependencies. The goal is to address the root causes of the problem rather than just its symptoms, which will make it possible to create effective corrective and preventative activities. The capacity to improve big data systems' stability, dependability, and general performance is given to organisations by practitioners who identify the core causes and obtain insightful knowledge about the systemic flaws or shortcomings that must be addressed.

8.4.2.5 Root Cause Validation

Root cause validation requires a thorough and stringent assessment to evaluate the correctness and validity of the discovered root cause in the context of big data systems. During this stage, it will be confirmed whether the root cause found through investigative methods, logical thinking, and data analysis is the main source of the issue that has been noticed. Validation frequently calls for additional testing, simulations, or analysis to demonstrate a cause-and-effect link between the determined root cause and the problem's manifestation. By subjecting the root cause to rigorous scrutiny and validation, analysts verify that it aligns with the available evidence, adheres to logical reasoning, and withstands critical analysis. This validation process is crucial to mitigate the risk of prematurely attributing the problem to an incorrect or incomplete root cause, which may lead to ineffective solutions or recurring issues. By confirming the accuracy of the root cause, stakeholders can confidently proceed with the formulation and implementation of appropriate corrective actions and preventive measures, addressing the underlying issue and reducing the likelihood of future occurrences of the problem.

8.4.2.6 Corrective Actions

To address the identified root cause and remedy the issue of big data systems, corrective actions entail creating and using focussed measures. Practitioners then formulate specific activities to address the underlying problem once the root cause has been correctly identified and validated. These steps may include a variety of interventions, such as adjusting system settings, changing software code, improving data quality checks, or putting new procedures in place. The remedial measures treat the fundamental cause and lessen its effects on the system's functionality, stability, or dependability. Additionally, when developing corrective actions, consideration is given to their viability, efficiency, and potential hazards. It is important to think about the possible effects of the remedial measures on other system components and to confirm that they are compatible with the system's current infrastructure and operational procedures. Following their formulation, corrective actions are routinely carried out, frequently adhering to a well-defined change management procedure while considering the required resources, the timetable, and collaboration with relevant parties. Corrective steps implemented correctly may considerably increase system performance, reduce risks, and stop similar issues from happening again, improving large-scale systems' overall dependability and functioning.

8.4.2.7 Preventive Measures

Preventive measures entail using proactive activities and techniques to reduce the chance of upcoming issues or failures. Preventive measures are created to address systemic vulnerabilities, reduce possible hazards, and improve the system's resilience based on the knowledge gathered from the root cause investigation. These actions include establishing reliable monitoring and alerting systems, putting preventative controls and safety precautions in place, enhancing system redundancies, and implementing best practices and industry standards. Preventive measures include training initiatives and programmes to ensure staff members have the knowledge and abilities to proactively recognise and solve possible concerns. Preventive actions can include ongoing examination and enhancement of system design, infrastructure, and procedures to accommodate changing threats and technological advancements. By putting preventative measures in place, businesses can successfully lower the likelihood of future issues occurring, improve system performance and dependability, and promote a mindset of proactive problem-solving and risk management in the context of big data platforms.

8.4.2.8 Implementation and Monitoring

The execution and supervision of corrective actions and preventative measures in big data systems are included in implementation and monitoring. The implementation phase entails implementing the intended changes, adjustments, or interventions after the selected actions have been developed and approved. Coordinating appropriate parties, distributing required resources, adhering to defined change management procedures, and adhering to project deadlines are all critical at this phase. It is crucial to maintain efficient communication, sufficient testing, and appropriate documentation to monitor the development and results of the executed measures. Monitoring is essential for determining the success of adopted policies and confirming their influence on the system. Monitoring is the gathering and analysis of pertinent data and performance indicators to assess how well the actions addressed the underlying issue and enhanced the system's overall performance. Any deviations, new problems, or improvement areas can be found through routine and regular monitoring. Organisations can make informed adjustments, spot potential gaps, and proactively handle emerging issues by closely monitoring the implemented corrective actions and preventive measures. This will help ensure the RCA process's success and effectiveness within big data systems.

8.4.2.9 Documentation and Reporting

Documentation and reporting refer to the thorough and systematic recording of all pertinent information, analysis findings, actions performed, and results. They are essential parts of the RCA process. This phase emphasises the need to document the RCA process to maintain traceability, knowledge retention, and efficient stakeholder communication. Data collection, analytical methods, research strategies, and validation procedures must be organised and preserved as part of the

documentation process. Additionally, it entails summing up the determined underlying cause, outlining the solutions created and put into practice in clear, specific detail, and recording the outcomes. By allowing practitioners to build on prior knowledge and reduce duplication of effort, this documentation is an invaluable resource for future studies. Additionally, reporting is crucial in conveying the conclusions and suggestions reached via the RCA process to the appropriate stakeholders. Reports often include a problem description, methodology used, RCA findings, remedial actions, preventative measures, and an outcome expectation. The reporting and documenting phase guarantees openness, responsibility, and the sharing of knowledge and lessons learnt to support organisational learning and ongoing development.

8.4.3 Tools and Techniques for RCA in Big Data Systems

A range of specialised software tools and analytical methods aim at identifying and resolving the underlying causes of issues in big data systems, which provide effective troubleshooting problems in different scenarios. They make it possible to gather and analyse system logs, track performance indicators, visualise and analyse complex data, do statistical analyses to look for trends and anomalies, and run simulations and controlled experiments. Users can learn more about the inner workings of big data systems, identify the sources of issues, and implement effective remedies by utilising these tools and methodologies.

Let's conduct an in-depth exploration and comprehensive analysis of these techniques and tools.

8.4.3.1 Log Analysis and Monitoring Tools

Log analysis and monitoring tools are crucial for RCA of big data systems, which gather and examine the logs produced by the system's many servers, applications, and frameworks. They assist in locating abnormalities, error messages, and behavioural patterns that might point to a problem's underlying cause. Tools for log analysis frequently include functions like search, filtering, and visualisation of log data. These functions allow for pinpointing issues and investigating their underlying causes. Here are some log analysis and monitoring tools:

- **ELK Stack**: ELK Stack[1] is a widely used open-source toolset for log analysis. Logstash collects, processes, and centralises logs from various sources, while Elasticsearch indexes and stores the log data for fast search and analysis. Moreover, Kibana provides a user-friendly interface for visualising and exploring log data.
- **Splunk**: Splunk[2] is a popular commercial tool for log analysis and monitoring that collects and indexes log data from different sources. Splunk provides powerful

[1] www.elastic.co/.
[2] https://www.splunk.com/.

querying capabilities, dashboards, and alerting features, allowing users to search, correlate, and analyse logs in real time.

8.4.3.2 Performance Monitoring and Profiling Tools

Performance profiling and monitoring technologies are utilised to measure and evaluate the performance of big data systems. These instruments gather performance indicators, including CPU and memory consumption, network throughput, and query response times. By keeping an eye on these measures, abnormalities, and bottlenecks may be found, which aids in determining the likely reasons for performance problems. Developers can spot resource-intensive processes, poor code, or configuration issues influencing system performance using profiling tools, which offer insights into the execution behaviour of applications. Some performance monitoring and profiling tools are described below:

- **Apache Kafka**: Apache Kafka[3] is a distributed streaming platform that can also monitor big data systems' performance. It provides message throughput, latency, and broker performance metrics, enabling users to monitor and optimise Kafka cluster performance.
- **Apache Spark Monitoring and Instrumentation**: Apache Spark, a widely used big data processing framework, offers built-in monitoring and instrumentation capabilities. It provides metrics related to job execution, task performance, memory usage, and data shuffle operations that some monitoring tools like Ganglia [7] or Prometheus[4] can be used to collect and visualise Spark performance metrics.

8.4.3.3 Data Visualisation and Analysis Tools

Data visualisation and analysis tools are used to understand patterns, correlations, and anomalies in big data systems, which assist in transforming complex data into visual representations. These tools allow researchers to visually browse big datasets and discover connections between system variables or components. The discovery of root causes is facilitated by interactive dashboards, charts, graphs, and heatmaps to graphically portray system behaviour and data patterns in data visualisation and analysis tools. The following describes some tools used for data visualisation and analysis:

- **Tableau**: Tableau[5] is a popular data visualisation tool that allows users to create interactive dashboards and visualisations. It supports various data sources, including big data platforms, and provides a wide range of chart types, graphs, and filters for exploring and analysing data visually.

[3] https://kafka.apache.org/.
[4] https://prometheus.io/.
[5] https://www.tableau.com/.

- **Apache Superset**: Apache Superset,[6] an open-source data exploration and visualisation platform, supports querying and visualising different data types and offers features like interactive dashboards, charting capabilities, and the ability to create and share data exploration workflows.

8.4.3.4 Statistical Analysis Techniques

Statistical analysis techniques are applied to analyse data quantitatively and spot statistically significant patterns or correlations that assist in discovering correlations, trends, and anomalies that might point to a problem's underlying cause. Regression analysis, hypothesis testing, correlation analysis, time-series analysis, and algorithms for anomaly identification are some examples of statistical analysis techniques. Using statistical analysis methodologies, users can obtain deeper insights into system behaviour and pinpoint probable causes based on data patterns and statistical significance. Some favourite programming languages are commonly used to perform statistical analysis, as indicated below:

- **R**: R[7] is a programming language and environment used for statistical computation and graphics, which offers a wide variety of statistical analysis methods, including time-series analysis, hypothesis testing, regression analysis, and hypothesis testing. For the analysis of logs collected from big data systems, R packages like "dplyr," "ggplot2," and "forecast" can be utilised.
- **Python libraries**: Python has several libraries for statistical analysis, such as NumPy,[8] SciPy,[9] and Pandas,[10] which offer functions and tools for performing statistical operations, hypothesis testing, and data manipulation. They can analyse large and streaming log data from big data systems to define the RCA of performance-related problems.

8.4.3.5 Experimentation and Simulation Tools

Experimentation and simulation techniques are utilised to perform tests in controlled situations, confirm theories, and pinpoint the root causes of the issues. These technologies enable organisations to simulate various scenarios, setups, or workload patterns in big data systems to comprehend the influence on system behaviour. By conducting experiments, users identify factors, test proposed fixes, and track the effects on system performance or data consistency. Tools for simulation can also assist in simulating specific problems or failures so that their causes can be investigated and potential remedies explored.

[6] https://superset.apache.org/.
[7] https://www.r-project.org/.
[8] https://numpy.org/.
[9] https://scipy.org/.
[10] https://pandas.pydata.org/.

- **Apache JMeter**: Apache JMeter[11] is an open-source tool for load testing and performance measurement. It allows users to simulate various scenarios, create test plans, and generate load on big data systems to assess their performance under conditions like fault injection scenarios.
- **SimGrid**: A simulation framework called SimGrid [8] makes creating and simulating distributed systems possible. It can be utilised to model the behaviour and functionality of big data systems, assessing the effects of various variables, settings, and workload patterns.
- **IoTSim-Osmosis**: IoTSim-Osmosis [9] It is an open-source simulation framework which enables the collection of the performance metrics from the MapReduce framework as well as testing and validation of osmotic computing applications.

8.4.4 Challenges and Considerations in RCA for Big Data Systems

The enormous scale, dispersed structure, and need for real-time processing make big data systems difficult to use. Due to the interdependencies between different components, the vast number of data, and the dynamic nature of the systems, identifying and addressing root causes can be difficult. The complexity of the infrastructure and software also presents additional challenges for RCA attempts. Proactive monitoring, scalability concerns, resource allocation, and keeping up with emerging technology are essential to manage these issues successfully. A thorough awareness of the unique difficulties and factors that must be considered to successfully diagnose problems, improve system dependability, and optimise performance is necessary for conducting RCA in big data systems.

8.4.4.1 Handling Huge Amount of Logs

Performing RCA for big data systems presents challenges related to the overwhelming amount of logs generated by these systems. Big data systems generate enormous logs from various components, such as servers, applications, and frameworks. These logs provide vital data that can help discover the underlying causes of performance-related problems. However, the sheer amount of logs can make collecting, storing, and analysing difficult. Effective log management procedures must be implemented to handle the enormous amount and velocity of log data, ensuring that pertinent logs are gathered and kept for RCA reasons.

Moreover, the dispersed nature of large data systems further complicates RCA attempts. It is challenging to get a complete picture of the system's behaviour since the logs produced by various components may be scattered over several nodes or clusters. Coordinating and correlating logs from diverse sources is crucial to precisely identify the events that led to a problem.

[11] https://jmeter.apache.org/.

8.4.4.2 Distributed Nature of Big Data Systems

In big data systems, the data processing tasks are divided and distributed across multiple nodes or machines, often forming a cluster or a network of interconnected nodes. This distributed nature of big data systems poses specific challenges and considerations in performing RCA for troubleshooting the problems of big data systems.

One of the key challenges is the complexity of identifying and diagnosing issues that span multiple nodes or components within the distributed system. It is challenging to identify the underlying source of a problem when it appears in several ways across various nodes. The interdependencies and interactions between dispersed components can make the RCA process more challenging since issues in one piece impact other parts. Furthermore, the dispersed nature of large data systems creates difficulties for coordination and communication among many nodes or clusters. Data consistency, synchronisation, and network latency problems can affect how well and reliably the system functions. The identification and eradication of root causes must take into account these difficulties as well as the possible effects of dispersed communication and coordination problems.

8.4.4.3 Complexity of Software and Infrastructure

It is challenging to track the flow of data and control since the software components of big data systems frequently include various technologies, frameworks, and libraries. Understanding the relationships and dependencies between different software components might make pinpointing an issue's core cause difficult. Additionally, problems in one component have ripple effects on others, making RCA more difficult. Additionally, the infrastructure for big data systems consists of a distributed network of nodes, clusters, and storage devices. In terms of hardware setups, operating systems, and network setups, these components might differ and are geographically separated. The RCA process is made more difficult by how diverse and heterogeneous the infrastructure is. Extra factors must be considered when doing RCA due to resource allocation problems, network connectivity, or device issues.

Furthermore, big data systems often depend on specialised tools and frameworks for handling massive amounts of data, such as Hadoop, Spark, or other distributed databases. These technologies have unique nuances and complexity, such as particular setup options, tuning parameters, and performance improvement strategies. Effective RCA requires a thorough understanding of the complexities of these technologies and how they relate to the rest of the system.

8.4.4.4 Real-Time and Streaming Data Considerations

Log data generated by real time and streaming components within big data systems are challenging due to many factors. One significant challenge is the continuous and high-velocity nature of real-time and streaming log data. Big data systems frequently handle enormous volumes of data in real time, producing a constant stream of log

events. It isn't very easy to gather, store, and analyse the data for RCA purposes in a timely and effective manner because of the huge volume and quick input of log data.

Another challenge is the need for near real-time analysis of streaming log data. Due to their offline processing architecture and potential for introducing delays when identifying and resolving root causes, traditional batch-oriented log analysis systems might not be appropriate for real-time RCA. RCA systems should deploy specialised methods and tools to instantly process and examine streaming log data to identify and diagnose problems quickly. Furthermore, other factors must be considered while assuring the dependability and integrity of streaming log data. Data loss or consistency issues can arise in real time and streaming applications due to network delay, system malfunctions, or high data ingestion rates. These issues must be resolved to ensure the gathered log data correctly reflects the system behaviour and can be trusted for RCA efforts.

8.4.4.5 Cross-Component Dependencies and Interactions

Big data systems consist of several components: storage systems, processing frameworks, databases, and networking infrastructure. The functionality, dependability, and performance of these components are interdependent, where problems with one component may affect the operation and performance of others. To conduct RCA in big data systems successfully, it is essential to comprehend and examine these relationships and interactions.

A cross-component environment demands extensive visibility and monitoring capabilities for resolving problems. Understanding system behaviour comprehensively—including data flow, control, and communication—becomes crucial for RCA. Understanding cross-component dependencies and locating the source of issues requires using monitoring tools and approaches that offer insights into interactions between various components and record the pertinent metrics and events.

8.5 Available Diagnosis Tools for Big Data Systems

Although diagnosing big data systems is difficult and complex, several tools can make the process easier. The most commonly used tools for debugging performance issues in big data systems are listed below.

8.5.1 Mantri

Mantri [10] is a big data debugging system for detecting and handling outliers in MapReduce clusters. Outliers, which are the tasks that significantly affect MapReduce clusters' performance, are detected using a statistical method. Using statistical analysis and ML algorithms, Mantri locates and manages outliers in MapReduce clusters. Outlier handling and outlier detection are the two stages of the Mantri

operation. Mantri employs a series of statistical measurements in the outlier identification stage to find nodes that are acting erratically. Mantri uses machine learning methods to forecast the effect of outlier nodes on the cluster and to take remedial action during the outlier handling phase.

Different workloads are used on an extensive testbed to gauge Mantri's performance. As a result, task execution times, cluster throughput, and other performance metrics significantly improve when Mantri is used to identify and manage outliers in MapReduce clusters.

8.5.2 TACC Stats

TACC Stats [11] is a real-time monitoring tool that gathers data from High Performance Computing (HPC) systems on the infrastructure and the status of each job. It has been used in production on multiple different methods for roughly five years, and many HPC systems are actively using it. It enables analysis and reporting using the data gathered, such as resource use, energy consumption, network, and I/O activities. It can provide information on performance degradation due to task mistakes, system flaws, and resource requirements depending on resource consumption. TACC Stats, however, cannot visualise the data gathered and does not offer root cause analysis. It was also designed for HPC clusters and is inappropriate for huge data systems.

8.5.3 DCDB Wintermute

DCDB Wintermute [12] offers real-time monitoring, which can spot performance problems and determine their root causes. It provides a broad selection of setup choices to fulfil the requirements of Operational Data Analytics (ODA) applications on HPC systems. DCDB Wintermute monitors and analyses big data tasks as it is designed for HPC systems. Its auto-scaling capabilities enable the automated reallocation or allotment of computer resources in response to variations in demand.

8.5.4 AutoDiagn

AutoDiagn [13] is a real-time performance diagnosis framework for big data systems. Its flexible architecture makes it adaptable for big data systems with the help of plug APIs. It consists of two main components; monitoring and diagnosing. AutoDiagn is written in Java, and all source code is published on GitHub.[12] Figure 8.4 shows the architecture of the AutoDiagn system.

The monitoring component is responsible for collecting the status of each specific big data task and infrastructure information of the big data cluster, which deploys

[12] https://github.com/umitdemirbaga/AutoDiagn.

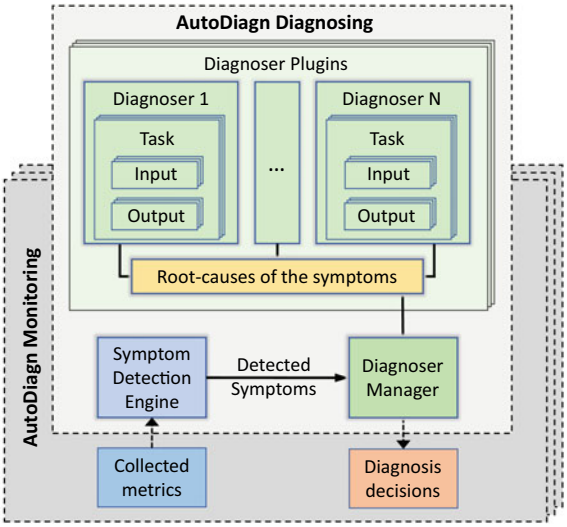


Fig. 8.4 AutoDiagn architecture

YARN APIs[13] to gather task information and deploys Sigar API[14] for infrastructure, such as CPU, memory, network, and I/O. Afterwards, it stores these logs in a time-series database, InfluxDB,[15] via a message broker system, RabbitMQ.[16] This robust data transfer pipeline allows monitoring the cluster of thousands of nodes. The diagnosing component, built on top of the monitoring system, analyses the collected logs and performs root cause analysis to identify the main problem causing performance reduction in the big data system.

As shown in Fig. 8.5, AutoDiagn diagnosing comprises three core components: Symptom Detection, Diagnosis Management, and Decision-Making. Symptom Detection continuously queries the streaming data from the RabbitMQ server to find outliers. Once outliers are detected, Diagnosis Management creates a diagnoser for each symptom to evaluate all the possible reasons for the outliers. Finally, Decision-Making defines the root cause of performance degradation.

[13] https://hadoop.apache.org/docs/r3.2.1/hadoop-yarn.
[14] https://github.com/hyperic/sigar.
[15] https://www.influxdata.com/.
[16] https://www.rabbitmq.com/.

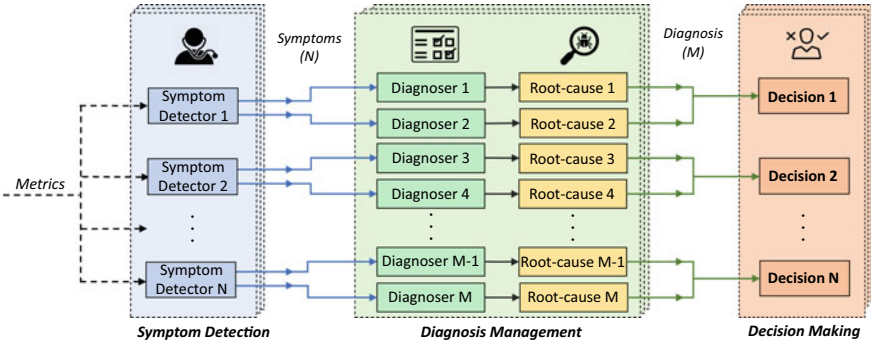


Fig. 8.5 AutoDiagn diagnosis workflow

8.6 Learning Outcomes of the Chapter

- **Debugging Big Data Systems for Big Data Analytics**: Exploring the debugging process for real-world performance problems in big data systems.
- **Debugging Steps**: Detailing the steps involved in debugging big data systems.
- **Problems in Big Data Systems**: Identifying common problems in big data systems, including data locality, resource heterogeneity, network issues, resource over-allocation, unnecessary speculation, and poor scheduling policy.
- **Root Cause Analysis in Big Data Systems**: Discussing the importance of root cause analysis in big data analytics, outlining the steps involved, and exploring tools and techniques for root cause analysis in big data systems. Also, addressing challenges and considerations in root cause analysis for big data systems.
- **Available Diagnosis Tools for Big Data Systems**: Introducing various diagnosis tools for big data systems, such as Mantri, TACC Stats, DCDB Wintermute, and AutoDiagn.

References

1. U. Demirbaga, G.S. Aujla, Rootpath: Root cause and critical path analysis to ensure sustainable and resilient consumer-centric big data processing under fault scenarios, in *IEEE Transactions on Consumer Electronics* (2023)
2. Z. Zong, R. Ge, Q. Gu, Marcher: A heterogeneous system supporting energy-aware high performance computing and big data analytics. Big Data Res. **8**, 27–38 (2017)
3. A. Pimpley, S. Li, R. Sen, S. Srinivasan, A. Jindal, Towards optimal resource allocation for big data analytics, in *EDBT* (2022), pp. 2–338

4. T.-D. Phan, S. Ibrahim, G. Antoniu, L. Bougé, On understanding the energy impact of speculative execution in hadoop, in *2015 IEEE International Conference on Data Science and Data Intensive Systems*. (IEEE, 2015), pp. 396–403
5. M.A. Barsalou, *Root Cause Analysis: A Step-by-step Guide to Using the Right Tool at the Right Time*. (CRC Press, 2014)
6. B. Andersen, T. Fagerhaug, *Root Cause Analysis*. (Quality Press, 2006)
7. Ganglia. Accessed: 2022-10-17. [Online]. Available: http://ganglia.info/http://ganglia.info/
8. H. Casanova, Simgrid: A toolkit for the simulation of application scheduling, in *Proceedings First IEEE/ACM International Symposium on Cluster Computing and the Grid*. (IEEE, 2001), pp. 430–437
9. K. Alwasel, D.N. Jha, F. Habeeb, U. Demirbaga, O. Rana, T. Baker, S. Dustdar, M. Villari, P. James, E. Solaiman et al., Iotsim-osmosis: A framework for modeling and simulating iot applications over an edge-cloud continuum. J. Syst. Architect. **116**, 101956 (2021)
10. G. Ananthanarayanan, S. Kandula, A.G. Greenberg, I. Stoica, Y. Lu, B. Saha, E. Harris, Reining in the outliers in map-reduce clusters using mantri. Osdi **10**(1), 24 (2010)
11. R.T. Evans, J.C. Browne, W.L. Barth, Understanding application and system performance through system-wide monitoring, in *IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. (IEEE, 2016), pp. 1702–1710
12. A. Netti, M. Müller, C. Guillen, M. Ott, D. Tafani, G. Ozer, M. Schulz, Dcdb wintermute: Enabling online and holistic operational data analytics on hpc systems, in *Proceedings of the 29th International Symposium on High-Performance Parallel and Distributed Computing* (2020), pp. 101–112
13. U. Demirbaga, Z. Wen, A. Noor, K. Mitra, K. Alwasel, S. Garg, A. Zomaya, R. Ranjan, Autodiagn: An automated real-time diagnosis framework for big data systems, in *IEEE Transactions on Computers* (2021)

Further Reading

14. U. Demirbaga, G.S. Aujla, RootPath: Root cause and critical path analysis to ensure sustainable and resilient consumer-centric big data processing under fault scenarios" in *IEEE Transactions on Consumer Electronics* https://doi.org/10.1109/TCE.2023.3329545
15. M. Marra, G. Polito, E.G. Boix, A debugging approach for live big data applications, in *Science of Computer Programming*, vol. 194 (2020), p. 102460
16. M.A. Gulzar, M. Interlandi, S. Yoo, S.D. Tetali, T. Condie, T. Millstein, M. Kim, Bigdebug: Debugging primitives for interactive big data processing in spark, in *Proceedings of the 38th International Conference on Software Engineering* (2016), pp. 784–795

Machine Learning for Big Data Analytics

9

Machine learning will automate jobs that most people thought could only be done by people.

—Dave Waters

This insightful chapter delves deeply into the enormous possibilities of using machine learning to extract meaningful insights from large amounts of data, which meticulously dissects the realm of supervised machine learning for big data analytics, unravelling the challenges inherent in its application and elucidating pre-processing methodologies essential for optimal outcomes. A comprehensive array of popular supervised machine learning algorithms is scrutinised, including Linear Regression, Logistic Regression, Decision Tree, Random Forest, Support Vector Machines, Naïve Bayes Classifier, and K-Nearest Neighbour. Transitioning seamlessly, the chapter navigates the landscape of unsupervised machine learning, shedding light on diverse techniques such as K-means Clustering, Hierarchical Clustering, Density-Based Spatial Clustering of Applications with Noise (DBSCAN), Gaussian Mixture Models, Principal Component Analysis, t-distributed Stochastic Neighbour Embedding (t-SNE), Apriori Algorithm, Isolation Forest, and Expectation-Maximisation. The chapter culminates by venturing into neural network algorithms, probabilistic learning fundamentals, and performance evaluation and optimisation techniques, providing a holistic panorama of machine learning paradigms tailored to the challenges of big data analytics.

9.1 Harnessing Machine Learning for Big Data Insights

Machine learning for big data involves using sophisticated computing algorithms and statistical methods to extract information, patterns, and insights from vast and complicated datasets [1]. Due to the exponential expansion of data across many disciplines,

Ü. Demirbaga et al., *Big Data Analytics*,
https://doi.org/10.1007/978-3-031-55639-5_9

traditional data processing systems frequently find it difficult to handle the sheer amount, diversity, and velocity of big data. Machine learning approaches provide a workable answer using computer power and algorithmic effectiveness to automatically identify patterns, forecast outcomes, and provide useful intelligence [2].

Using distributed computing frameworks and parallel processing strategies, machine learning algorithms are created for the big data environment to manage enormous datasets. Scalable and high-performance analytics is made possible by these algorithms' effective parallel processing and data analysis across various computer resources. Additionally, machine learning models may modify and learn from big datasets, revealing hidden patterns and correlations that would not be seen using conventional data analysis techniques.

Scalable machine learning refers to a machine learning system's ability to handle vast volumes of data and complicated models efficiently and effectively [3]. Scalable machine learning seeks to create algorithms and approaches that may be utilised to analyse enormous datasets while retaining high accuracy and efficiency [3]. One of the most challenging aspects of scaling up machine learning is dealing with the massive amounts of data that must be handled. Researchers and practitioners have developed various ways to meet this difficulty, including distributed computing, parallel processing, and efficient data storage and retrieval. These techniques enable machine learning models to be trained on big datasets in a fraction of the time standard methods require. Another challenge in scaling up machine learning is maintaining model accuracy and reliability as the dataset expands. Regarding this issue, academics have devised strategies for decreasing overfitting, such as regularisation and early halting, and increasing model generalisation, such as transfer learning and ensemble approaches.

9.2 Supervised Machine Learning for Big Data Analytics

Supervised machine learning is a subfield of machine learning in which a model is trained on a labelled dataset to predict the target variable for new, unseen data. Collecting insights and information from vast and complex datasets that cannot be handled or analysed using typical data processing techniques is called big data analytics. Combining supervised machine learning with big data analytics presents an intriguing possibility to tackle various real-world challenges such as customer churn prediction, fraud detection, and financial forecasting.

9.2.1 Challenges of Applying Supervised Machine Learning to Big Data Analytics

There are various challenges to applying supervised machine learning to large-scale data analytics [4]. First, because huge datasets may be massive and complicated, data pretreatment is an important stage in the machine learning pipeline. This includes

data cleaning and transformation, feature selection, and handling missing values, outliers, and noisy data. The second challenge is that selecting a suitable machine learning method for large datasets is difficult due to the large number of accessible possibilities, each with its own set of strengths and weaknesses. Moreover, assessing a model's performance on large datasets necessitates using specialised approaches such as cross-validation and hyperparameter optimisation to guarantee that the model is not overfitting the training data. Furthermore, the scalability of machine learning algorithms to enormous datasets is a significant difficulty since many classical methods may be incapable of dealing with such massive data. Finally, ethical concerns about using big data and machine learning must be addressed, such as privacy, bias, and fairness. Overcoming these obstacles necessitates a thorough grasp of supervised machine learning and big data analytics and the creation of new algorithms, tools, and approaches for effective and efficient data analysis.

9.2.2 Pre-processing Big Data for Supervised Machine Learning

Data cleaning and transformation are two critical processes in pre-processing big data for supervised machine learning.

Data cleaning entails discovering and repairing or deleting the data's flaws, inconsistencies, and inaccuracies. This might involve dealing with missing numbers, fixing typos or spelling errors, and deleting outliers or noisy data points. Data cleaning guarantees that the data is correct, comprehensive, and consistent. Data wrangling is another major phase in the machine learning process as it increases machine learning models' accuracy and performance. By cleaning, integrating, and converting, users can verify that the data is correct, consistent, and in a format suited for analysis and machine learning.

The process of data wrangling, including turning raw data into a more appropriate format for analysis and machine learning, includes a variety of steps:

- **Data integration:** This entails merging data from several sources into a single dataset.
- **Data transformation:** This involves transforming the data into a more acceptable format for analysis and machine learning. Data transformation requires the following steps:
  - **Feature engineering:** This step allows the creation of new features from existing ones.
  - **Feature selection:** It is the process of selecting the most relevant characteristics from a dataset to improve the performance of the ML model.
  - **Data discretisation:** This step includes converting continuous data into discrete categories.

In the context of supervised machine learning, the advantages of data transformation are listed below:

- **Improving prediction accuracy:** We can eliminate flaws and inconsistencies contributing to erroneous predictions by cleaning and converting data.
- **Reducing the training time:** We can speed up the training of machine learning models by normalising the data.
- **Improving model interpretability:** We can make comprehending how machine learning models generate predictions easier by translating the data into a more intelligible format.

Let's examine these two indispensable concepts, feature engineering and feature selection, in detail.

9.2.2.1 Feature Engineering

The technique of producing new features or variables from existing ones in a dataset to improve the performance of machine learning algorithms is known as feature engineering. It entails choosing, manipulating, and merging vital information to capture the underlying patterns and connections in the data. Feature engineering is an important phase in supervised machine learning for big data analytics since it substantially influences the prediction models' accuracy and efficiency.

Feature engineering aims to develop new features that will aid machine learning algorithms in better capturing the complicated interactions between input data and the target variable [5]. This might include choosing the most relevant characteristics, converting or normalising the features, or developing new features that capture higher-order interactions between existing features. For example, relevant information can include customer demographics, use trends, and purchase history in a customer churn prediction scenario. Creating new features such as the average time between purchases, the total amount spent on the product, or the number of customer support calls the consumer makes might all be feature engineering.

Several techniques, such as domain knowledge, statistical methods, and automated feature selection algorithms, can be used for feature engineering in supervised machine learning for big data analytics. Defining domain knowledge entails using expert knowledge to determine the most relevant aspects of a particular problem. Statistical approaches like correlation analysis and principal component analysis can be used to discover the essential features in the data. Automated feature selection methods, such as decision trees or evolutionary algorithms, can automatically choose the most relevant characteristics based on their predictive value.

9.2.2.2 Feature Selection

Feature selection is a method in supervised machine learning for big data analytics that includes selecting a subset of the most relevant characteristics from a more comprehensive set of possible features to improve the performance of machine learning

algorithms [6]. Feature selection is critical for various reasons. First, using a smaller set of features reduces the dimensionality of the data, making the ML algorithm more efficient and less prone to overfitting. Second, utilising a smaller number of features can make the model more interpretable since it is simpler to comprehend the influence of individual variables on the target variable. Third, employing fewer features can enhance model accuracy by reducing unnecessary or redundant information that might bring noise into the model.

Numerous feature selection strategies exist in supervised machine learning for big data analytics. Using domain expertise to determine the most relevant characteristics for a particular problem is one technique. This method relies on expert knowledge to identify the most essential characteristics based on their relevance to the issue area. Another strategy is determining the most significant traits based on their association with the target variable using statistical approaches such as correlation analysis, mutual information, or hypothesis testing. Another option is to utilise machine learning techniques like decision trees, linear models, or neural networks to identify the most relevant characteristics based on their prediction potential.

9.2.3 Popular Supervised Machine Learning Algorithms for Big Data Analytics

Linear regression, logistic regression, decision trees, random forests, Support Vector Machines (SVMs), Naïve Bayes classifiers, and k-Nearest Neighbours (K-NN) are popular supervised machine learning approaches for big data analytics. These algorithms create predictive models that generate accurate predictions or classifications for fresh, unknown data based on prior data with known outcomes. They can handle various data formats and may be utilised for regression and classification tasks. These supervised approaches are widely used in multiple industries, including banking, health care, and e-commerce, to generate data-driven choices and automate fraud detection, consumer segmentation, and personalised recommendations.

9.2.3.1 Linear Regression

Linear regression is a popular supervised machine learning approach for big data analytics that predicts a continuous target variable using one or more input characteristics, which finds the best-fit line describing the linear connection between the input characteristics and the target variable [7].

In big data analytics, linear regression can analyse enormous datasets to uncover trends, patterns, and correlations between variables. Linear regression is most beneficial when the connection between the input characteristics and the target variable is linear, which indicates that the target variable can be written as a linear combination of the input features.

There are some challenges in implementing linear regression in big data. The dataset may contain many characteristics, making it challenging to determine the most relevant aspects contributing to the goal variable when employing linear

regression for big data analytics. Feature selection approaches such as correlation analysis or regularisation methods can be utilised to pick the most relevant characteristics for the model. Another challenge is that the dataset may contain noisy or missing data, which might affect the linear regression model's accuracy. Data preparation techniques such as data cleaning, feature scaling, and outlier identification can be applied to solve these challenges and increase the model's performance.

9.2.3.2 Logistic Regression

Logistic regression is a sophisticated and adaptable technique for various classification problems in big data analytics, which is often used to estimate the likelihood of an event occurring based on one or more input characteristics [8]. Logistic Regression simulates the link between the input properties and the target variable, a binary variable that indicates the presence or absence of the event. It is very popular among data analysts and business stakeholders as it is simple to execute, analyse, and convey. Moreover, it can handle large-scale and complex datasets, making it a commonly used algorithm in big data applications.

Logistic regression can be complex in big data analytics for various reasons. For starters, massive datasets with many features might result in overfitting and poor generalisation performance. Regularisation techniques or feature selection approaches can be used to remedy this. Second, unbalanced classes can lead to biased models that predict negative cases more accurately than favourable situations. Techniques like oversampling or undersampling and employing alternative cost functions might be applied to overcome this issue. Lastly, Logistic regression presumes a linear connection between the input characteristics and the target variable's log odds, which may not always be accurate in complicated datasets.

9.2.3.3 Decision Tree

Decision trees are a strong and adaptable method for big data analytics problems, including regression and classification. It is a well-known and widely applied supervised machine learning technique that labels or assigns a value to each subset based on the target variable after recursively partitioning the dataset into subgroups based on the input feature values [9].

Large and complex datasets with many features can be automatically reduced to the essential components of the model using the Decision Tree technique. Decision trees are perfect for complex data because they can manage nonlinear correlations between input attributes and goal variables. They are well-liked by data analysts and other business stakeholders since they are also easy to understand and visualise. Decision trees are important in identifying patterns and correlations within large datasets when used with appropriate techniques to reduce overfitting and improve model performance.

The decision tree has the benefit of being able to manage nonlinear connections between input data and the target variable. They are also simple to analyse and present, making them popular among data analysts and corporate stakeholders. One

disadvantage of employing decision trees for large data analytics is that they are susceptible to overfitting, mainly when the tree is deep and complicated. Techniques such as pruning, regularisation, or ensemble approaches like random forest or gradient boosting can solve this issue and increase the model's generalisation performance.

9.2.3.4 Random Forest

Random forest, another popular and powerful machine learning technique for solving classification and regression problems in big data analytics, is an ensemble learning method that integrates numerous decision trees to improve model performance and prevent overfitting [10]. Random forest can analyse massive, complicated datasets consisting of many attributes. It works by building numerous decision trees on various random subsets of the data and characteristics, then aggregating each tree's predictions to get a final forecast, which reduces variation and improves model stability.

Random forest can handle categorical and streaming but missing and noisy data. It can automatically identify the most significant attributes of the model, making it suitable for large datasets. Simple logic and structure make it popular among data analysts and corporate stakeholders.

One disadvantage of utilising Random forest for big data analytics is that it can be computationally costly, especially when the dataset is large or the forest has many trees. Techniques such as parallel processing and distributed computing can be implemented to boost performance and scalability.

9.2.3.5 Support Vector Machines (SVMs)

Support vector machines are another sophisticated and frequently used machine learning method for big data analytics that is especially well-suited for binary classification problems [11]. They determine the hyperplane that best divides the classes in the input feature space and then apply it to generate predictions on new data. SVMs help analyse massive, complicated datasets with multiple characteristics. They are especially beneficial when the data cannot be separated linearly because they may employ kernel functions to translate the input characteristics into a higher dimensional space where they can be separated. Furthermore, SVMs can deal with categorical and continuous data and are relatively robust to noisy data.

SVMs are a margin-based strategy that minimises the distance between the decision border and the nearest data points in each class. When paired with techniques like regularisation or cross-validation, this can increase generalisation performance and prevent overfitting. SVMs may also be utilised for classification and regression problems, making them useful algorithms for big data analytics. One disadvantage of employing SVMs for big data analytics is that they can be computationally expensive, mainly when the dataset is enormous, or the kernel function is complicated. Techniques such as kernel approximation and stochastic gradient descent can be applied to increase performance and scalability.

9.2.3.6 Naïve Bayes Classifier

Naive Bayes (NB) classifier is a probabilistic algorithm extensively used for classification jobs in big data analytics, which operates by computing the likelihood of each class given the input features and then choosing the class with the highest probability as the predicted class [12]. In practice, NB presumes that characteristics are independent, which is not always true. Despite this oversimplified assumption, NB classifiers frequently outperform expectations in practice and are widely used in applications such as spam filtering, sentiment analysis, and document classification.

There are various advantages to using NB classifiers for big data analytics. To begin with, they are computationally efficient, making them appropriate for massive and complicated datasets. NB classifiers use a modest amount of memory to store the model parameters and can rapidly generate predictions for new data points without requiring considerable processing. Moreover, NB classifiers are simple to implement and comprehend, needing a fundamental understanding of probability theory. Furthermore, they can handle categorical and continuous data and resist noisy data. Finally, because NB classifiers can be trained on small datasets, they are a good choice for applications with limited labelled data.

One weakness of NB classifiers is that they might suffer from the zero-frequency problem, which occurs when the chance of a feature appearing in a particular class in the training data is zero, resulting in inaccurate classification. Smoothing techniques such as Laplace smoothing and Bayesian smoothing can be used to solve this problem.

9.2.3.7 K-Nearest Neighbour (K-NN) Algorithm

The K-NN method is a non-parametric, lazy learning technique extensively used in big data analytics for classification and regression applications [13]. When a new data point is received, the algorithm selects the k-nearest neighbours to that data point from the training dataset based on a similarity metric such as Euclidean distance or Manhattan distance. The new data point's class or value is then calculated based on the class or average value of its k-nearest neighbours.

K-NN is a straightforward and intuitive technique requiring no training or parameter adjustment, making it suitable for large data settings. It is widely used to process categorical and numerical data for image recognition, natural language processing, and recommendation systems. However, when working on large-scale datasets, K-NN can be expensive and time-consuming as it calculates the distance between the new data point and all other points in the dataset.

The simplicity and flexibility of the K-NN algorithm provide excellent convenience for users in the analysis of big data. K-NN shows high performance in categories, numerical data, and complex data. Big data analytics offers significant savings in time and resources since it does not require a separate training process. K-NN can also manage noisy or missing data. K-NN is widely used for image detection, natural language processing, and recommendation systems. Despite its simplicity, k-NN has proven effective in many real-world applications, making it a vital weapon in data scientists' arsenal.

9.3 Unsupervised Machine Learning for Big Data Analytics

9.3.1 K-means Clustering

K-means clustering is an unsupervised machine learning approach that attempts to divide a given dataset into a defined number of groups. It works by assigning data points to the nearest centroid, which represents the centre of a cluster, iteratively and updating the centroids depending on the newly assigned points [14]. The algorithm's goal is to minimise within-cluster variation, which means keeping the data points in each cluster as close as feasible. This iterative process is repeated until the centroids stabilise and no additional reassignments occur. K-means clustering is commonly used in data mining, pattern recognition, and picture segmentation. Its appeal stems from its simplicity, efficiency, and scalability. However, it is sensitive to initial centroid placement and may converge to local optima. Understanding the fundamental concepts and concerns of K-means clustering is critical for using this approach effectively in exploratory data analysis and clustering assignments.

9.3.2 Hierarchical Clustering

Hierarchical clustering is an unsupervised machine learning approach to group data based on similarity or dissimilarity, which creates a hierarchical structure of clusters in the form of a dendrogram, a tree-like figure [15]. The technique allows for either a bottom-up (agglomerative) or a top-down (divisive) approach to clustering.

In the agglomerative technique, each data point represents an independent cluster at first. The method iteratively merges the most comparable clusters based on a distance metric of choice, resulting in a cluster hierarchy. The algorithm computes the pairwise distance or dissimilarity between clusters at each iteration and merges the two most similar clusters into a new cluster. This method is repeated until all data points have been integrated into a single cluster or a stopping requirement has been satisfied.

The divisive method begins with a single cluster containing all data points and recursively divides it into smaller clusters depending on dissimilarity measurements. The technique detects the cluster with the largest dissimilarity and separates it into two or more subclusters at each iteration. This method is repeated until each data point forms its distinct collection or until a stopping requirement is met.

Hierarchical clustering represents data hierarchically, allowing for flexible investigation of clustering solutions at various degrees of granularity. It does not need the number of clusters to be specified in advance, making it suitable for exploratory research. However, hierarchical clustering may be computationally costly for big datasets, and the choice of distance metric and linking mechanism has a major influence on the outcomes. Understanding the concepts and implications of hierarchical clustering is critical for using this approach effectively in various clustering and pattern recognition applications.

9.3.3 DBSCAN

DBSCAN is an unsupervised machine learning approach for the density-based grouping of spatial data. DBSCAN, in contrast to typical clustering methods, does not require previous knowledge of the number of clusters. It finds groups based on the density of data points in their surroundings. The algorithm defines two parameters: epsilon (), which indicates the radius of neighbouring points to be considered, and MinPts, which gives the minimum number of points necessary to make a dense zone [16].

DBSCAN begins by randomly choosing an unvisited data point and identifying its neighbourhood. If there are more than MinPts points in this neighbourhood, a new cluster is constructed, and all access points inside the area are allocated to this cluster. The process is repeated for these newly assigned points recursively until no more points can be added to the collection. The programme then moves on to the next unvisited location. It continues the process, either forming new clusters or categorising the point as noise if the density conditions are unmet.

DBSCAN's merits include its ability to detect clusters of any form and deal with noise. It resists parameter changes and does not rely on distance measures, making it appropriate for datasets with varying densities. However, selecting optimal numbers for MinPts can be difficult, and the algorithm's performance degrades as dimensionality increases. Understanding DBSCAN's complexities is critical for efficient use in spatial data grouping and analysis jobs.

9.3.4 Gaussian Mixture Models (GMM)

Gaussian Mixture Models (GMMs) are statistical models used in machine learning for clustering and density estimation tasks. They are especially helpful in unsupervised machine learning applications where the objective is to identify structures and patterns in data without the aid of labelled samples. A GMM displays the data as a mixture of Gaussian distributions, where each part of the mixture represents a cluster or a subpopulation [17]. The name "Gaussian" describes the shape of the bell-shaped distribution, which is characterised by their mean (average) and variance (spread).

The GMM assumes that the data points are produced by combining K Gaussian distributions, where K is the number of clusters that need estimation. Each data point has a latent variable identifying the group it originated from. Because the latent variables are not immediately seen, they are sometimes called "hidden variables". Gaussian distribution parameters and the probability of data points belonging to each cluster must be estimated using GMM. The mean and covariance matrices of each Gaussian distribution and the mixing coefficients, which reflect the percentages of data points assigned to each cluster, are all parameters of a GMM. The mean and covariance matrix control each Gaussian distribution's position and form, and the mixing coefficients govern each cluster's relative importance.

The Expectation-Maximisation (EM) technique is frequently used to estimate GMM parameters. The EM algorithm iteratively changes the parameters to maximise

the likelihood of the observed data given the present estimations. Based on the most recent parameter estimations, the E-step calculates the chance that each data point belongs to each cluster. It updates the parameters by averaging the data points in the M-step, where the weights are the probability calculated in the E-step. The model can be utilised for several tasks once the GMM parameters have been determined. For clustering, the highest probability cluster is given to each data point. The probability density function of the data may be estimated using GMM, which is utilised for density estimation. The likelihood of new data points can be calculated, or new samples can be generated from the learnt distribution.

9.3.5 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a dimensionality reduction technique used in unsupervised machine learning for analysing and visualising high-dimensional data, which aims to convert the original characteristics into a new collection of variables known as principal components [18]. The initial elements making up these main components are linear combinations, and they were selected to capture the most variance in the data possible. Moreover, PCA enables a more succinct and comprehensible description of the dataset by reducing the number of main components used to describe the data.

There are several uses for PCA in big data analytics. Limiting the number of dimensions in the data to two or three primary components may be utilised for visualisation, making it simple to see and explore the data. PCA is frequently used as a pre-processing step before other machine learning techniques to decrease computing complexity and eliminate duplicate or unnecessary information. Additionally, PCA can help determine the dataset's most significant characteristics and offer perceptions of the underlying structure or connections between the variables.

The following stages are involved in performing PCA [19]:

1. **Standardisation:** PCA often begins with standardising the data by subtracting and dividing the mean by the standard deviation for each feature. Standardisation guarantees that each character has a comparable scale and prevents the analysis from being dominated by elements with bigger variations.
2. **Covariance Matrix Calculation:** The next step is constructing the covariance matrix once the data has been standardised. The covariance matrix calculates the correlations between feature pairs and reveals how the characteristics vary collectively.
3. **Eigenvalue Decomposition:** The covariance matrix is split into eigenvectors and eigenvalues. The eigenvalues show how much variance is collected along each eigenvector, and the eigenvectors reflect the axes or directions in the original feature space.
4. **Selection of Principal Components:** The eigenvectors are arranged in decreasing order according to the respective eigenvalues. The eigenvector with the greatest eigenvalue represents the first principal component; the second main component

is represented by the eigenvector with the next highest eigenvalue, and so on. Selecting a criterion, such as keeping components that account for no less than a predetermined proportion of the total variance (for example, 95%), helps to establish how many primary components to keep.

5. **Projection onto Principal Components:** The original data is projected onto the chosen primary components. The coordinates along the major components represented by each data point may be considered new features that capture the most significant trends in the data.

9.3.6 t-SNE

t-SNE is a powerful dimensionality reduction technique to visualise high-dimensional data. Unlike other methods, t-SNE retains local interactions and captures intricate nonlinear patterns. It can do this by producing probability distributions that evaluate the degree of similarity between data points in both high-dimensional and low-dimensional regions [20]. By reducing the divergence between these distributions, t-SNE iteratively adjusts the placements of the data points in the low-dimensional space, producing a scatter plot that reveals clusters, neighbourhoods, and overarching patterns in the data.

Big data analytics benefit from the visualisation offered by t-SNE since it enables analysts to examine and decipher complicated relationships in high-dimensional datasets. Finding clusters, subgroups, and outliers is where this method shines the most. It has been effectively applied in several fields, including studying social networks, natural language processing, genomics, and image analysis. Since t-SNE might not scale well to large datasets, the technique's computing cost must also be considered. Additionally, a thorough examination and comprehension of the underlying data are necessary to interpret t-SNE visualisations.

9.3.7 Apriori Algorithm

The Apriori method is popular in unsupervised machine learning for finding frequent item sets in a transactional database. It is very helpful in big data analytics when datasets have a lot of transactions and objects. The principal objective of the Apriori algorithm is to find groups of objects that commonly appear together in transactions [21]. These collections, known as frequent itemsets, are essential to association rule mining. In market basket analysis or recommendation systems, association rules are frequently used to explain relationships between items.

The level-wise search method employed by the Apriori algorithm produces frequent item sets. All individual items are first identified to determine the initial collection of frequently occurring itemsets. Then merging the popular items from the previous level to create new candidate items gradually lengthens the items. The algorithm prunes or removes candidate itemsets that do not reach a certain

minimum support level. The percentage of transactions in which an item set appears is its support.

The Apriori algorithm has several uses in large data analytics. It is used to analyse business transactional data, including retail, e-commerce, and consumer behaviour analysis. Insights regarding product co-occurrences, consumer buying habits, and cross-selling opportunities can be gained from the association rules that have been found. Additionally, the algorithm can support decision-making by optimising product placement, creating marketing plans, and providing tailored recommendations.

9.3.8 Isolation Forest

The isolation forest algorithm is a well-liked unsupervised machine learning method for anomaly identification in big data analytics, which is intended to spot data points that are odd or aberrant and depart from the rest of the data [22]. The method successfully locates outliers or abnormal patterns in massive datasets by separating anomalies. The isolation forest technique uses the idea that anomalies are often more isolated and rarer than regular data points. It generates a random forest-like structure where each tree is developed by picking a feature and a random split value within the range of that feature. The procedure is carried out again until all of the data points are isolated or separated.

Anomaly data points require fewer splits to isolate than normal ones while building the isolation forest [23]. This is because anomalies stand out from the remainder of the data due to unique attribute values. The method provides a score for anomalies in each data point based on the average number of splits needed to isolate each data point across several trees. It is more likely to be an abnormality if the anomaly score is lower. A threshold value is established to identify a data point as abnormal or typical. Anomalies are defined as data points with anomaly scores over this level, and normal data points are those with scores below them. The threshold may be established by examining the distribution of anomaly scores or using domain expertise.

The isolation forest technique has several benefits for big data analytics anomaly detection. It is appropriate for huge datasets because of its linear time complexity concerning the amount of data points. The approach is resilient and adaptable to different data types since it does not rely on any assumptions regarding the distribution of the underlying data. Additionally, it can effectively manage high-dimensional data unaffected by excessive characteristics. Isolation forest has been used in many fields, including fraud detection, network intrusion detection, system monitoring, and outlier identification in sensor data. Analysts can quickly find and look at unusual data items, revealing important insights or possible dangers.

9.3.9 Expectation-Maximisation Algorithm

The EM algorithm is a popular iterative optimisation technique used in unsupervised machine learning for estimating the parameters of statistical models, which

is especially useful in big data analytics, where datasets are frequently complicated and contain hidden or latent factors [24]. The primary aim of the EM algorithm is to locate the highest likelihood or maximum a posteriori estimations of the model parameters when there are gaps in the data or incomplete datasets. In a two-step iterative process, the expectation step (E-step) and the maximisation step (M-step) alternate in this algorithm.

In the E-step, the method computes the expected value of the log-likelihood function concerning the most recent estimations of the model parameters. In this stage, latent or missing variable values are estimated using observed data and existing parameter estimations. The E-step uses the existing model parameters to determine the likelihood or probability of each latent variable falling into a certain state or category. The method updates the estimates of the model parameters in the M-step by maximising the anticipated log-likelihood achieved in the E-step. The next step is to find the parameter values that maximise the expected log-likelihood function. Finding the ideal parameter values often entails solving optimisation equations or using numerical techniques. The E-step and M-step are iteratively repeated by the EM algorithm until convergence. At this point, there is no longer a substantial change in the parameter estimations between iterations. The technique looks for a set of parameter estimates that, while accounting for the existence of hidden or latent variables, maximise the likelihood of the observed data.

There are several uses for the EM algorithm in big data analytics. Cluster techniques like the GMM are frequently utilised to estimate the parameters of the underlying probability distributions. It is also used in many other unsupervised learning tasks, such as imputation for missing data, mixture models, and latent variable models.

9.3.10 Spectral Clustering

Spectral clustering is a powerful unsupervised machine learning technique used for clustering analysis in large-scale datasets. It is particularly effective in locating clusters or groups in large datasets when the underlying structure may be difficult for conventional distance-based clustering approaches to capture fully [25].

The spectrum clustering method performs clustering by using the spectrum characteristics of the data. It displays the data as an affinity or similarity matrix, which calculates the pairwise connections between the data points. Different similarity metrics, such as Gaussian similarity, cosine similarity, or k-nearest neighbours, can be used to build the affinity matrix. After obtaining the affinity matrix, spectral clustering attempts to divide the data into clusters by converting the issue into a task of graph partitioning. It uses a dimensionality reduction approach to the affinity matrix, such as PCA or Laplacian Eigenmaps. The dimensionality reduction stage minimises noise or unimportant data while capturing the data's most useful features. After dimensionality reduction, spectral clustering uses a graph partitioning technique to locate the clusters by treating the modified data as nodes in a graph. A common strategy is using the K-means algorithm on the data's reduced-dimensional form. Based on how close the data points are to the cluster centroids, the K-means

algorithm distributes data points to clusters. An additional choice is to utilise spectral clustering techniques that work directly with the eigenvectors of the Laplacian matrix created from the affinity matrix.

The advantage of spectral clustering is that it can handle data not well segregated in the original feature space and find clusters with complicated forms. It can detect clusters with overlapping or asymmetrical areas and capture nonlinear interactions. Spectral clustering is used in several fields, including social network analysis, document clustering, picture segmentation, and gene expression analysis. However, there are other things to think about with spectral clustering. It could need parameter tweaking, such as figuring out how many clusters there are or choosing the best similarity metric. The dimensionality reduction and network partitioning methodology you select can impact how well Spectral Clustering performs. Additionally, spectral clustering can have a high computational cost due to the necessary matrix operations, particularly for big datasets.

9.3.11 Mean Shift

Mean shift is a well-known unsupervised machine learning approach for clustering and mode-seeking analysis in large-scale data analysis that excels in locating dense areas or modes in datasets where more conventional clustering algorithms would have trouble [26].

The mean shift technique incrementally moves data points towards the neighbourhood's local mean until convergence [27]. It uses each data point as a kernel and determines the bandwidth by averaging all the data points that are close together. In the following iteration, this mean becomes the new position of the data point. The process is repeated until the data converges to a stable place. The fundamental tenet of mean shift is that data point density increases with increased data point density, which causes a convergence towards modes or dense zones. Data points converge and settle around the methods as they travel towards areas of increased density, forming clusters. The neighbourhood size around each data point is determined by the bandwidth parameter in mean shift. A finer-grained analysis that captures smaller and more distinct clusters is made possible by a narrower bandwidth. In contrast, a wider bandwidth catches bigger clusters but can cause surrounding clusters to merge. Choosing the right bandwidth is essential to getting useful clustering results.

Mean shift offers several benefits for big data analytics. It is appropriate for instances when the cluster structure is unknown since it does not require previous knowledge of the number of clusters. It can deal with asymmetrical datasets, have different densities, and contain outliers. Additionally, as mean shift is a non-parametric technique, it does not assume anything about the distribution of the underlying data. Mean shift has a computational complexity limit, especially for big datasets. Pairwise distance calculations between data points are necessary for the algorithm's iterations, which might take some time. Various methods can increase the algorithm's effectiveness, including kernel density estimation.

Mean shift has been used in various fields, such as data compression, object tracking, picture segmentation, and computer vision. It has also been used in anomaly detection, recommendation systems, and social network research. It is a useful tool for comprehending intricate patterns and structures in large datasets because of its capacity to recognise dense areas and modes in data.

9.4 Neural Networks Algorithms

Neural networks, or artificial neural networks, are typical supervised and unsupervised machine learning algorithms used in big data analytics, which comprise connected layers of nodes that analyse and change incoming data to predict the output [28]. As neural networks are well-suited for big data applications, they can learn complicated patterns and correlations from large-scale datasets, which makes them significantly successful in image and audio recognition, natural language processing, and predictive modelling. However, training and modelling neural networks are computationally costly, requiring substantial computing resources. Besides, the interpretability of neural networks is quite tricky, making it challenging to understand how the model generates its predictions.

9.4.1 The Components of Neural Networks

Neural networks comprise interconnected nodes, or neurons, that process and transmit information. Figure 9.1 shows the basic structure of neural networks consisting of the input layer, hidden layer, and output layer.

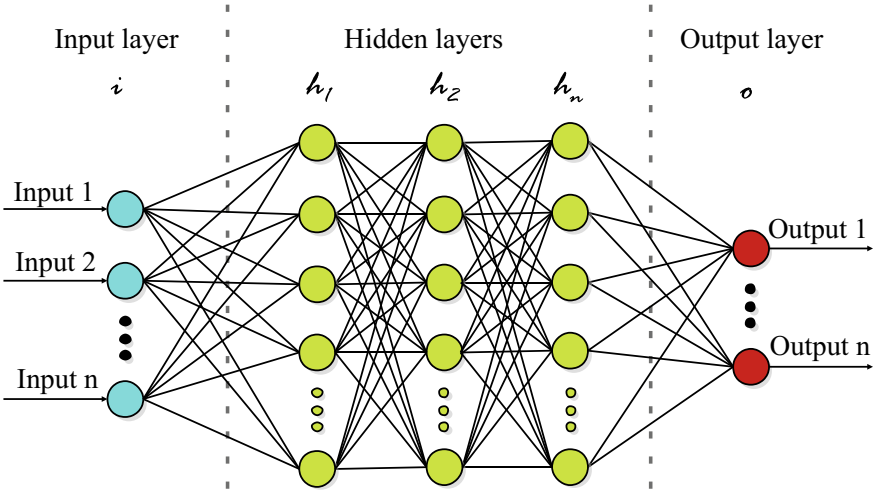


Fig. 9.1 Neural network structure diagram

The input layer is the network's first layer, receiving input data and passing it onto the first hidden layer [29]. The data's input characteristics dictate the number of neurons in the input layer. The hidden layers are the layers between the input and output layers responsible for most of the computation in a neural network. Each neuron in a hidden layer gets input from the preceding layer and produces an output using a nonlinear activation function. The number of hidden layers and neurons in each layer are hyperparameters that may be changed to improve network performance. The output layer is the network's final layer, which delivers predictions or classifications based on the inputs and the network's learned parameters. The number of classes determines the output layer's number of neurons in a classification issue or the number of output variables in a regression task.

9.4.2 The Types of Neural Networks

Neural networks are machine learning approaches inspired by the human brain's neural structure. There are several types of neural networks, each of which is built for certain tasks and problem domains. Here are some examples of neural networks that are commonly used:

9.4.2.1 Feedforward Neural Network (FNN)

A Feedforward Neural Network (FNN) is a neural network in which information goes from the input layer to the output layer in a single direction [30]. It has no loops or feedback links, giving it a simple architecture. FNNs are made up of numerous layers of linked nodes, which are also known as artificial neurons. Each neuron takes inputs, assigns weights, and uses an activation function to generate an output. The outputs of one layer are sent into the next layer until the final output layer gives the network's forecast. FNNs are frequently used for classification and regression tasks, in which the network learns to map input data to match output values. Figure 9.2 depicts a structural diagram of an FNN.

9.4.2.2 Convolutional Neural Network (CNN)

Convolutional neural network is a deep learning model that handles grid-like data structures like pictures or time series data, which features a hierarchical design with numerous layers that conduct two essential operations collectively: convolution and pooling [31]. Learnable filters are used in convolutional layers to convolve across the input, recognising local patterns and recording spatial correlations. This convolutional approach guarantees that the network's learnt representations have translation invariance and spatial hierarchy. Pooling layers then downsample the feature maps, lowering dimensionality while maintaining the most important information. To combine the retrieved data and provide predictions, the last layers of a CNN generally consist of completely linked layers. According to this specialised design, CNNs can automatically learn complicated characteristics from data, making them extremely

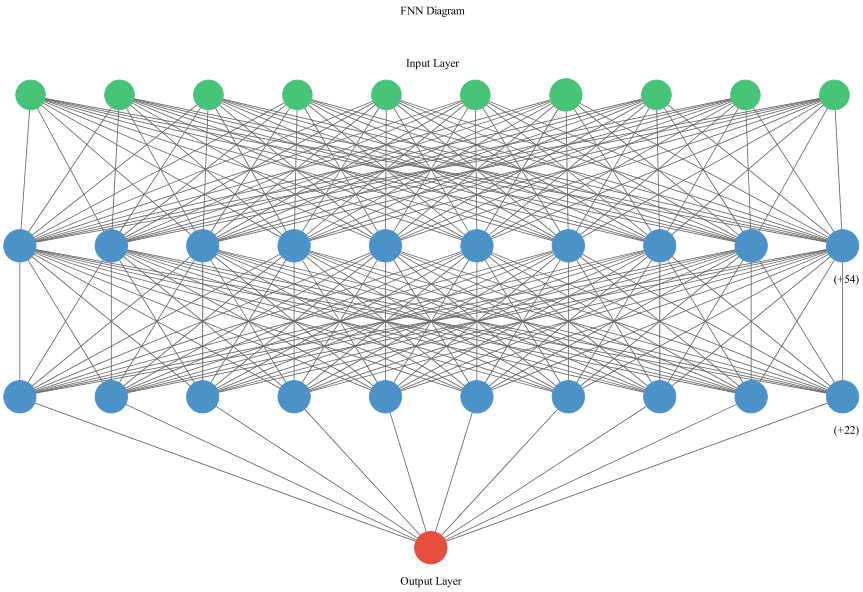


Fig. 9.2 FNN structure diagram

successful in computer vision applications like picture classification, object recognition, and semantic segmentation. Furthermore, CNNs have made major advances in various disciplines, including natural language processing and speech recognition. Figure 9.3 depicts the structure diagram of a CNN.

9.4.2.3 Recurrent Neural Network (RNN)

RNN is a neural network that excels at sequential data processing by including recurrent connections in its architecture [32]. RNNs, instead of feedforward networks, have cyclic links that allow them to store information across time, making them suited for jobs with temporal dependencies. Each RNN unit has an internal memory that stores a summary of its encountered inputs. To create an output and move data to the following time step, this memory is updated and combined with the input that is now being received. The ability of RNNs to handle variable-length sequences and identify long-term dependencies is well known. Conversely, standard RNNs have a limitation in capturing distant links due to the vanishing gradient problem. To address this problem, variants of Gated Recurrent Units (GRU) and Long Short-Term Memory (LSTM) have been developed. Natural language processing, speech recognition, machine translation, and other tasks requiring sequential data analysis frequently use RNNs (Fig. 9.4).

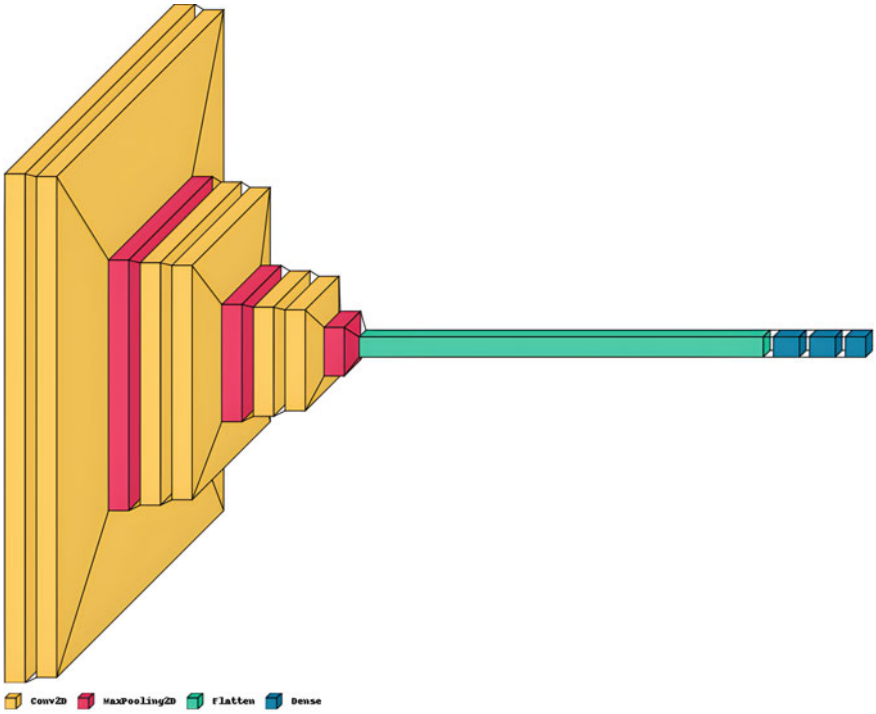


Fig. 9.3 CNN structure diagram

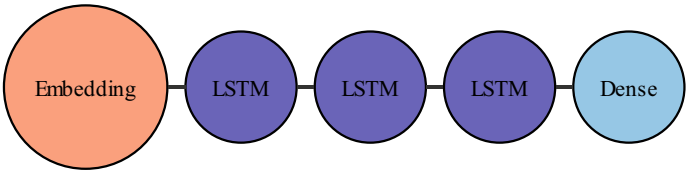


Fig. 9.4 RNN structure diagram

9.4.2.4 Long Short-Term Memory (LSTM) Network

The Long Short-Term Memory (LSTM) network, a more advanced variation of RNNs, solves the vanishing gradient problem and captures long-term relationships in sequential data [33]. LSTM networks use memory cells, critical components that allow the network to store information over lengthy sequences. A memory cell comprises three types of gates: an input gate, a forget gate, and an output gate. These gates control the flow of information into and out of the cell state, allowing the LSTM to keep or forget information as needed. The input gate regulates how much fresh information is added to the cell state, whereas the forget gate controls how much past information is deleted. The output gate controls how much of the cell state is

accessible to the network output. The LSTM network may learn to collect and transport meaningful information across lengthy sequences by adjusting the weights of the gates during training, successfully reducing the vanishing gradient problem [34]. Because of its capacity to capture long-term dependencies, LSTM networks excel in speech recognition, language modelling, and sequence prediction tasks. LSTM versions, including Bidirectional LSTMs and Stacked LSTMs, have also been created to improve their efficacy in capturing complicated relationships in sequential data.

9.4.2.5 Generative Adversarial Network (GAN)

A Generative Adversarial Network (GAN), a deep learning model class comprising two coupled neural networks, a generator and a discriminator, is built to learn and create new data instances like training datasets [35]. The generator network creates synthetic samples by translating random noise into a high-dimensional data space. In contrast, the discriminator network distinguishes between produced and genuine pieces from the training dataset. The two networks are trained in a competitive environment, with the generator attempting to generate realistic illustrations to mislead the discriminator and the discriminator trying to discern between actual and produced data reliably. This negative interaction results in a dynamic training process in which the generator increases its capacity to generate realistic samples. At the same time, the discriminator develops its ability to discriminate between actual and false models. GANs can capture and understand the underlying distribution of the training data through this repeated training process, which allows them to produce innovative and high-quality data examples [36]. GANs have been used effectively in various applications, including picture synthesis, image-to-image translation, text creation, and data augmentation. GANs have been a popular strategy in generative modelling because they generate realistic and varied samples.

9.4.2.6 Autoencoder

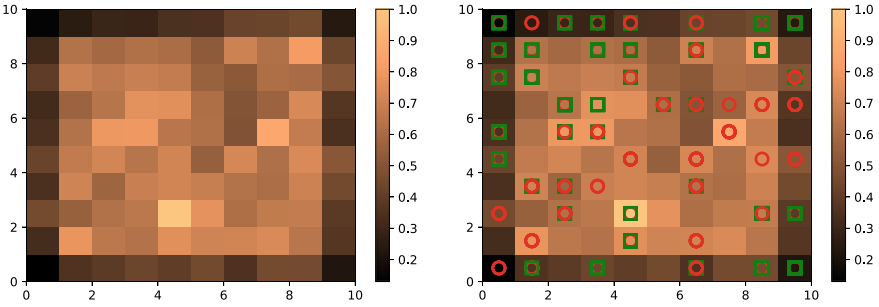
An artificial neural network called an autoencoder is developed for unsupervised learning and dimensionality reduction applications. It comprises two primary parts: an encoder and a decoder. The encoder reduces the input data to a lower dimensional representation, while the decoder attempts to recreate the original input from the compressed form [37]. The goal of the autoencoder is to minimise reconstruction error, which encourages the network to acquire meaningful features that capture the most important information in the input data. Autoencoders learn to effectively capture and compress the underlying structure of the data by limiting the network to recreate the input. The compressed representation, also known as the latent space, represents the input data in a compact and informative manner. Autoencoders are useful for various tasks, including data compression, denoising, and anomaly detection. They may also be utilised as generative models, in which the decoder produces new samples by sampling from the latent space. Autoencoders have been used in various fields, including image processing, natural language processing, and recommendation systems. Autoencoder variants such as Variational Autoencoders (VAEs)

and Sparse Autoencoders (SAEs) have been created to improve their capabilities and solve specific issues in unsupervised learning tasks.

9.4.2.7 Self-Organising Map (SOM)

A Self-Organising Map (SOM), also known as a Kohonen map, is an unsupervised learning method to visualise and cluster complex high-dimensional data. SOMs use a grid-like structure of artificial neurons arranged in a two-dimensional lattice [38]. Each neuron is assigned a weight vector representing a location in the input data space during training. The SOM method modifies these weight vectors iteratively depending on the similarity between the input data and the neurons' weights. The essential idea behind SOM is competitive learning, in which the neuron with the closest weight vector to the input data wins or is referred to as the "best-matching unit" (BMU). Weight updates are performed on the BMU and its neighbouring neurons, allowing them to gradually converge towards representing distinct parts of the input data space. This procedure produces a topological mapping in which comparable input data is projected to neighbouring neurons on the SOM grid. SOMs provide excellent visualisation capabilities by retaining the structural linkages of the input data. They may also be utilised for clustering, where related data points are allocated to the same or neighbouring neurons. Because of their capacity to expose underlying patterns, reduce dimensionality, and give insights into the structure of complicated datasets, SOMs have been widely used in various disciplines, including data mining, image processing, and exploratory data analysis.

Figure 9.5 shows the outputs of the SOM algorithm implemented to predict the health status of machines in a big data cluster [39]. Figure 9.5a shows the background of the SOM distance map, and Fig. 9.5b depicts the predicted nodes as healthy and unhealthy.



(a) The background of SOM distance map

(b) Predicting the health status of machines

Fig. 9.5 Big data failure prediction using SOM

9.5 Probabilistic Learning for Big Data Analytics

Probabilistic learning for big data analytics is the study of large-scale datasets, sometimes called big data, using probabilistic modelling and inference techniques [40]. The amount, pace, and diversity of data present in big data settings provide several issues addressed by combining machine learning and probability theory principles. Probabilistic learning focusses on developing models that incorporate the inherent uncertainty in data and generating predictions or deriving significant insights from those predictions using probabilistic reasoning. These models show interactions and dependencies between variables in a probabilistic graphical form, allowing for a more subtle data comprehension.

Since noisy, partial, or missing data are frequently present in real-world datasets, probabilistic learning has several advantages for big data analytics. It offers an adaptable framework for integrating existing knowledge, incorporating domain experience, and updating models as new data becomes available. Big data analytics uses probabilistic learning, including anomaly detection, clustering, topic modelling, recommendation systems, and probabilistic classification and regression. These applications can offer insightful information, enhance decision-making, and make it possible to find hidden patterns and trends in huge and complicated datasets by utilising probabilistic models. The scalability, interpretability, integration with deep learning, and ethical issues related to probabilistic learning for big data analytics are still being researched.

9.5.1 Fundamentals of Probabilistic Learning

The foundations of probabilistic learning include the essential ideas and methods that support using probabilistic models in machine learning. Understanding probability theory and its application to modelling uncertainty and Bayesian inference techniques for data-driven learning are required. While Bayesian inference enables the integration of previous information and data to update and improve models, probability theory provides a mathematical framework for quantifying uncertainty and expressing correlations between variables. Maximal likelihood estimation, Bayesian learning, and the many categories of probabilistic models, including Bayesian networks, Markov random fields, and hidden Markov models, are important ideas in the foundations of probabilistic learning. By understanding these foundations, practitioners are given the groundwork to use probabilistic learning techniques to model complicated real-world processes, make predictions, and draw conclusions from data while accounting for uncertainty.

9.5.1.1 Probability Theory and Bayesian Inference

Probability theory, a subfield of mathematics, offers a framework for estimating uncertainty and analysing random events [41]. It analyses probability distributions, which ascribe probabilities to various outcomes or occurrences. Probability theory

enables us to represent and analyse uncertainty statistically and base judgements and predictions on the facts. Contrarily, Bayesian inference is a key technique in statistics and machine learning that mixes previous information and facts to update and improve our beliefs or models [42]. Adding new data offers a logical method for updating our knowledge of a situation, enabling us to make probabilistic predictions and reach judgements based on a posterior distribution. Bayesian inference is useful in many fields, including data analysis, decision-making, and machine learning. It enables a flexible and coherent framework for learning from data while accounting for uncertainty and incorporating previous assumptions.

9.5.1.2 Types of Probabilistic Models

Numerous varieties of probabilistic models are often employed in statistics and machine learning. The interactions and dependencies between variables can be represented and captured in many ways depending on the type of model used. Here are the top three probabilistic model categories:

- **Bayesian Networks:** Bayesian networks, known as belief networks or causal networks, use Directed Acyclic Graphs (DAGs) to depict the probabilistic connections between variables. The graph's edges show dependencies, while the nodes represent variables. The conditional probability distributions linked to each node capture the dependencies between the variable and its parent nodes. Bayesian networks are often employed in probabilistic inference, causal modelling, and reasoning under uncertainty.
- **Markov Random Fields:** Markov Random Fields (MRFs), also known as undirected graphical models or Markov networks, use an undirected graph to express the probabilistic connections between variables. Typically described as a set of potential functions over cliques (subsets of variables) in the graph, MRFs represent the local interdependence between variables. Where variables have significant dependence on their immediate surroundings, such as in image processing, computer vision, and spatial modelling, MRFs are frequently utilised.
- **Hidden Markov Models:** Hidden Markov Models (HMMs) are probabilistic models used to represent sequential data when the underlying state is not readily apparent. HMMs comprise several hidden states, observable symbols, and probability for transition and emission. While the seen symbols reflect the quantifiable outputs, the hidden conditions indicate the underlying system dynamics. For applications like speech recognition, part-of-speech tagging, and Deoxyribonucleic acid (DNA) sequence analysis, HMMs are often employed in speech recognition, natural language processing, and bioinformatics.

9.5.1.3 Learning from Data

Learning from data describes the method of employing different algorithms and approaches to extract useful knowledge or patterns from the data that is already

accessible. It includes creating models or algorithms to identify underlying patterns and correlations in the data, make predictions, or categorise novel cases.

- **Maximum Likelihood Estimation:** Maximum Likelihood Estimation (MLE) is a technique for estimating a statistical model's parameters from observable data. MLE entails determining the values of model parameters that maximise the likelihood of the observed data in the context of learning from data. It assumes that the data come from a certain probability distribution, and its objective is to identify the parameter values that increase the likelihood that the observed data are accurate. Numerous machine learning techniques employ MLE extensively, including Gaussian mixture models, logistic regression, and linear regression.
- **Bayesian Learning:** Bayesian learning is a method of learning from data that considers the observed data and previous information or assumptions regarding a model's parameters. The Bayes theorem is used to update the prior beliefs in light of the observed data to produce a posterior distribution over the model parameters. Bayesian learning offers a compelling framework for fusing information from the past with data. In this way, it enables more reliable and adaptable learning. It makes it possible to quantify uncertainty and makes decision-making easier by considering both data-driven evidence and prior knowledge. Among the many fields in which Bayesian approaches are used are classification, regression, and inference in probabilistic graphical models.

Bayesian learning and MLE are both basic methods for learning from data. While Bayesian learning combines previous beliefs and updates them based on observed data to generate a posterior distribution, MLE concentrates on obtaining parameter estimates that maximise the likelihood of observed data. These methods give us the tools for parameter estimation, model fitting, and inference, allowing us to learn from data and make reasoned judgements in various machine learning and statistical modelling problems.

9.5.2 Scalable Algorithms for Probabilistic Learning

Scalable probabilistic methods have been developed to address the computing difficulties of probabilistic learning when working with large-scale datasets. Applying probabilistic models to real-world issues is made possible by these algorithms, designed to handle and analyse enormous volumes of data rapidly. Probabilistic scalable algorithms strive to reduce the computational complexity of probabilistic learning by embracing approaches that allow for efficient processing, parallelisation, and distributed computing.

The main goal of these methods is to get past the constraints that big data's size and complexity impose. To this end, they use tactics that maximise computational efficiency, promote computation processing in parallel, and use distributed computing infrastructures. Probabilistic scalable algorithms allow the use of probabilistic models for actual implementation and utilisation to solve problems in the real world.

9.5.2.1 Variational Inference

Variational inference is a statistical framework that offers a computationally manageable method for approximating posterior distributions in intricate probabilistic models [43]. The difficulty of using Bayesian inference when computing the precise posterior distribution is computationally expensive or analytically challenging is addressed. The main principle of variational inference is to frame the issue as an optimisation challenge. Variational inference looks for an approximation to the posterior that belongs to a tractable family of distributions, such as Gaussian distributions, rather than directly computing the genuine posterior distribution, which entails evaluating intractable integrals.

To perform variational inference, it is necessary to establish a parametric family of distributions, referred to as the variational family, that is adaptable enough to represent the complexity of the posterior distribution. The next step in optimisation is to find the parameters that reduce the KL divergence between the variational distribution and the real posterior.

The scalability and effectiveness of the variational inference make it more popular in Bayesian statistics and machine learning. Researchers and professionals may use it to approximatively represent complicated posterior distributions, allowing them to make inferences and generate probabilistic predictions in various applications, even when accurate belief is impractical.

9.5.2.2 Expectation-Maximisation Algorithm

The EM algorithm, an iterative optimisation method, estimates parameters in statistical models that contain latent or unobserved variables [44]. It deals with the parameter estimation problem when the observable data is insufficient or has missing values. The method functions in the expectation (E) step and the maximisation (M) step.

In the E-step, the method determines the latent variables' posterior probability or predicted values in light of the observed data and the current parameter estimations. The present analyses create a full data likelihood function, and the posterior probabilities are calculated using Bayesian inference or other well-known probabilistic methods. Given the observed data, these probabilities show the latent variable uncertainty.

The approach maximises the anticipated total data log-likelihood while considering the model parameters in the M-step. The anticipated log-likelihood obtained in the E-step is maximised by finding the parameter values that maximise it. The Newton-Raphson method or gradient ascent are common examples of numerical optimisation techniques used in this stage. The process gradually increases the probability that the observed data is real by iteratively updating the parameter estimates based on the predicted values of the latent variables.

The EM algorithm repeats the E and M phases until convergence, frequently indicated by a predetermined threshold or the consistency of the parameter estimations. When the method converges, it produces estimates for the parameters that maximise the likelihood of the observed data. The EM enables parameter estimation in

complicated models with missing or partial data, making it extensively relevant in disciplines like machine learning, statistics, and data analysis. It is especially helpful in scenarios where hidden structures or unknown causes influence observable data because of its capacity to incorporate information from unseen variables.

9.5.2.3 Gibbs Sampling

Gibbs Sampling is a Markov Chain Monte Carlo (MCMC) technique used to sample from complicated probability distributions when direct sampling is impractical [45]. For models with high-dimensional or structured parameter spaces, it works very well. Josiah Willard Gibbs, the mathematical pioneer who popularised the idea of "ensemble sampling" in statistical physics, is credited with giving the approach its name. With Gibbs Sampling, samples are taken from a joint probability distribution across several variables by repeatedly sampling each variable in dependence on the other variables' most recent values. Following the conditional probability principle, each variable is sampled from its conditional distribution based on the importance of the other variables.

The method starts with a default set of values for each variable and then iteratively moves forward. Each time an iteration is performed, one variable is chosen, and a new value is sampled from its conditional distribution depending on the other variables' most recent values. Until convergence is reached, this process is repeated for all variables. Gibbs Sampling is an invaluable tool in many fields, such as Bayesian statistics, machine learning, and inference in graphical models. It is a well-liked option for probabilistic modelling and inference jobs due to its capacity to sample from intricate and high-dimensional distributions even when direct sampling is impossible.

9.5.2.4 Stochastic Gradient Markov Chain Monte Carlo (SG-MCMC)

Stochastic Gradient Markov Chain Monte Carlo (SG-MCMC) is a class of algorithms combining the advantages of stochastic gradient optimisation and Markov Chain Monte Carlo (MCMC) methods [46]. SG-MCMC algorithms are created to conduct Bayesian inference and sampling in large-scale and high-dimensional probabilistic models. In SG-MCMC, a stochastic approximation, rather than the complete dataset, is used to compute gradients. The approach calculates the gradient of the desired posterior distribution using mini-batches or subsets of the data. Due to the computation of the gradient being done on smaller sections of the data, this enables computational efficiency and scalability. Following the Metropolis-Hastings technique, the stochastic gradient estimates are utilised to update the Markov chain's current state. With each sample reliant on the prior model, this approach sequentially creates samples from the target distribution.

SG-MCMC methods have several benefits, including the capacity to handle high-dimensional parameter spaces and scalability to enormous datasets. These algorithms may perform inference and sampling on huge datasets while keeping computational tractability by using stochastic gradients. To achieve convergence and correct

selection, it is necessary to carefully choose learning rates and optimisation strategies since SG-MCMC adds more sources of bias and variation than conventional MCMC approaches.

9.5.2.5 Parallel Markov Chain Monte Carlo (MCMC)

Parallel MCMC describes a group of algorithms that use parallel computer systems to quicken the convergence of MCMC techniques [47]. Regarding handling huge datasets or intricate models, these techniques seek to overcome the computational difficulties of MCMC. In parallel MCMC, many chains are executed concurrently on various processors or computers, each investigating a different parameter space area. To enable exploration of the whole parameter space and guarantee convergence to the desired distribution, these chains regularly connect by sharing information, such as parameter samples or proposals.

Parallel MCMC algorithms use a variety of methodologies for generating proposals and organising communication between the chains. Examples include delayed rejection, where chains submit samples that may be accepted later in the process, and parallel tempering, where chains with various temperatures share information. Through the use of parallel processing, MCMC delivers considerable computational gains. These techniques make it possible to explore the parameter space more effectively, reach convergence more quickly, and handle bigger datasets by dividing the computational work among numerous processors or computers.

However, the efficiency of parallel MCMC algorithms depends on various elements, including the problem's structure, communication costs, load balancing, and the number of available computing resources. The parallelisation method must be designed and optimised properly to guarantee efficient and effective sampling from the target distribution.

9.5.2.6 Streaming Variational Bayes (SVB)

Streaming Variational Bayes (SVB) is a computational algorithm for online learning where data arrives continuously or sequentially. To progressively update probabilistic models as new data becomes available, SVB integrates ideas from variational inference and online learning. SVB aims to maximise the Evidence Lower Bound (ELBO), and stochastic optimisation techniques like stochastic gradient ascent are used to update the model parameters [48]. The marginal probability of the model is approximated by the ELBO, which stands for the actual posterior distribution.

SVB changes the model when new data come in by analysing each point individually or in tiny mini-batches. The method incorporates further information and progressively modifies the parameters to modify the present variational approximation of the posterior distribution. SVB frequently uses mean-field approximations and other simplified variational families to simplify variational families to retain computing performance. To reduce memory use and assure the model's adaptability to evolving data distributions over time, SVB may also use techniques like forgetting mechanisms or windowing.

9.5.3 Applications of Probabilistic Learning in Big Data Analytics

There are several uses for probabilistic learning in big data analytics, which makes it possible to analyse huge datasets effectively and efficiently. Numerous applications of probabilistic learning are highlighted, demonstrating its applicability and importance in big data analytics. Probabilistic learning approaches improve analytics jobs' accuracy, scalability, and interpretability on large-scale datasets by including uncertainty, capturing relationships, and modelling complicated data distributions. The following are examples of applications where probabilistic learning is applied:

9.5.3.1 Anomaly Detection

By simulating the typical behaviour of the system or data distribution, probabilistic learning approaches can detect anomalies or outliers in large data. These techniques can identify unexpected trends, fraudulent activity, or system flaws that could go undetected with conventional rule-based approaches because they capture the uncertainty in the data. Finding anomalies in a dataset involves looking for unusual or unexpected occurrences that differ from the norm. These abnormalities might be signs of system errors, security lapses, fraud, or any other odd behaviour that needs to be addressed. Anomaly detection systems may recognise and highlight such anomalies using probabilistic learning approaches, allowing prompt intervention and decision-making.

Probabilistic learning uses statistical models and probability theory to discover anomalies. These models accurately depict the underlying distributions and patterns of typical data occurrences. Anomalies can be recognised as cases that significantly depart from the predicted probability distributions using probabilistic techniques, such as GMM, HMM, or BN. Anomaly identification in big data analytics confronts particular difficulties because of the amount, velocity, and variety of data. The size and complexity of big data may make it difficult for conventional anomaly detection techniques to manage them. But developments in parallel processing and distributed computing have made it easier to create scalable anomaly detection systems for big data analytics.

9.5.3.2 Clustering and Topic Modelling

Clustering, also known as unsupervised learning, aims to combine comparable data instances based on their genetic similarities or patterns [49]. Without previous knowledge of the class labels or target variables, it aids in revealing hidden patterns and relationships within a dataset. Clustering algorithms can allocate data instances to various clusters according to the probability distributions of their attributes by using probabilistic learning techniques.

Topic modelling, an application of probabilistic learning, aims to discover the underlying themes or subjects in a sizable corpus of textual data. The distribution of words across documents is used to group unstructured text data into meaningful clusters. The most prevalent themes in a dataset may be found using topic modelling

algorithms, which utilise probabilistic models to assign probabilities to words and topics, such as the commonly used Latent Dirichlet Allocation (LDA).

In big data analytics, clustering and topic modelling confront particular difficulties due to the large dimensionality and scalability of the data. The sheer volume and complexity of big data may make it difficult to manage conventional clustering and topic modelling approaches. However, improvements in parallel processing methods and distributed computing frameworks like Apache Spark have made it possible to create scalable and effective clustering and topic modelling algorithms for large data situations. Probabilistic learning offers several benefits in topic modelling and clustering for big data analytics. These methods provide a more flexible and sophisticated depiction of the underlying patterns by considering the probabilistic distributions of data instances and characteristics. The resulting more precise grouping and topic modelling findings are made possible by detecting subtle links and structures within the data. Moreover, combining clustering and topic modelling with other big data analytics approaches, such as classification or recommendation systems, can improve the overall comprehension and use of the data. While topic modelling can help with document classification, sentiment analysis, or personalised content recommendations, clustering can offer insights into customer segmentation, anomaly detection, or pattern identification.

9.5.3.3 Recommendation Systems

Recommendation systems are essential to big data analytics because they offer consumers customised and pertinent suggestions based on their preferences and use habits. These systems employ probabilistic learning techniques to analyse massive amounts of data and produce correct recommendations [50]. This enables organisations to improve user experiences, boost consumer engagement, and spur revenue development.

Recommendation systems frequently utilise probabilistic learning algorithms to represent user preferences and item attributes. One of the main methods used in recommendation systems, collaborative filtering, uses probabilistic learning to forecast user preferences by considering the actions and choices of like users. Collaborative filtering algorithms determine the likelihood of a user's interest in a specific thing by analysing patterns of co-occurrence or similarity between users and objects, and they then produce tailored suggestions.

Probabilistic matrix factorisation (PMF), a well-liked probabilistic learning method, is applied in recommendation systems. The latent features that PMF learns from the observed user-item interactions are represented as users and objects in a low-dimensional space. Even with sparse and high-dimensional data, PMF can forecast user preferences and produce personalised suggestions by capturing the underlying probabilistic correlations between users and goods. Another approach is Bayesian personalised ranking (BPR), which uses probabilistic learning to optimise the order of items for each user. BPR can learn about user preferences by maximising the likelihood of viewing a rated item higher than an unranked item while considering the user's prior interactions and preferences. BPR can increase the efficacy of recommen-

dation systems by using probabilistic learning to create precise and individualised rankings for things.

Recommendation systems in the context of big data analytics confront special difficulties due to the vast amount of data and the requirement for real-time suggestion production. The computational complexity and latency demands in big data settings may be too much for conventional recommendation systems to manage. However, improvements in distributed computing frameworks and scalable probabilistic learning approaches, such as parallelised algorithms and approximate inference techniques, have solved these issues and made it possible to execute large-scale recommendation jobs effectively.

In recommendation systems, probabilistic learning has several benefits. These systems can handle data sparsity and cold-start issues, capture uncertainty in user preferences, and deliver individualised suggestions with greater accuracy using probabilistic models and methodologies. Additionally, probabilistic learning enables the integration of numerous data sources and contextual information, such as demographic information, item qualities, and temporal dynamics, boosting the relevance and contextualisation of suggestions. Furthermore, the constant feedback loop in recommendation systems enables gradual learning and adjustment to shift consumer preferences. The accuracy and relevance of suggestions may be improved over time by adding fresh user feedback and interactions into probabilistic learning algorithms that can update the recommendation models in real time.

9.5.3.4 NLP

NLP focusses on how computers and human language interact. Computational algorithms and models are developed to analyse, comprehend, and produce natural language text or voice [51]. By enabling machines to extract valuable insights from massive volumes of unstructured textual data, including social media postings, customer reviews, news articles, and documents, NLP is essential to big data analytics.

Text classification, sentiment analysis, named entity identification, machine translation, question answering, text summarisation, and language production are just a few of the diverse activities covered by NLP. To handle and analyse natural language data successfully, these tasks need cutting-edge methods from various fields, including machine learning, statistical modelling, linguistics, and information retrieval.

NLP has substantially used statistical methods for tasks like language modelling, part-of-speech tagging, and voice recognition, such as n-gram language and hidden Markov models. The prediction and production of linguistic sequences are made possible by these models, which use massive corpora of annotated text to understand statistical patterns and probabilities. Deep learning techniques have recently transformed NL), notably RNNs and transformer models. This considerably improves sentiment analysis, machine translation, and text production. These models can also capture long-range relationships and contextual information in the text. By learning from enormous volumes of unlabeled text data, pre-trained language models such as Bidirectional Encoder Representations from Transformers (BERT) and Generative Pre-trained Transformer (GPT) have produced impressive results.

9.5.3.5 Image and Video Analysis

Image and video analysis is crucial in big data analytics because it enables the extraction of valuable insights from enormous visual datasets. Advanced computational approaches are required to analyse, comprehend, and extract useful information from these comprehensive multimedia sources in the context of exponential expansion in picture and video data.

A strong framework for modelling uncertainty and producing predictions based on probabilistic inference is provided by probabilistic learning. Probabilistic learning techniques enable the creation of sophisticated algorithms that can successfully manage the complexity and uncertainty of visual data in the context of image and video analysis. Probabilistic learning techniques also provide a flexible framework for integrating limitations or past information into the analytical process. Probabilistic models can improve the accuracy and resilience of image and video analysis by integrating previous probabilistic models or priors, domain-specific knowledge, or contextual information. This is especially useful when working with sparse or noisy training data since it may make up for the data's shortcomings by using prior knowledge.

In image analysis, probabilistic learning methods are deployed for tasks including semantic segmentation, object identification, and picture classification, which predict the likelihood of several classes or labels for a given image by learning from labelled training data, enabling precise classification and identification. Furthermore, probabilistic models may make exact object delineation in complicated situations by semantic segmentation, including spatial connections between picture pixels or areas. Similarly, probabilistic learning techniques can be utilised in video analysis for tasks like action recognition, event detection, and video summarisation, representing the probabilistic links between actions, events, and their visual signals by modelling the temporal dynamics and dependencies in video sequences. This helps with tasks like video surveillance, anomaly detection, and content summarising by enabling the identification and interpretation of actions and events inside films.

9.6 Performance Evaluation and Optimisation Techniques

Machine learning performance evaluation and optimisation techniques are essential for determining the best model, choosing over- or underfitting problems, optimising hyperparameters, and ensuring the model's predictions generalise to new data. These methods direct the creation and enhancement of machine learning models, producing more precise and trustworthy outcomes.

9.6.1 Evaluation Metrics for Supervised Machine Learning Algorithms

Evaluation metrics are the measures used to assess the performance of ML algorithms that help evaluate different models in terms of their ability to make accurate

predictions. Using such metrics, models or algorithms can be compared, and the areas where the model can be improved can be identified.

The following are some evaluation metrics for supervised machine learning algorithms:

9.6.1.1 Accuracy

Accuracy is one of the most straightforward evaluation metrics, which counts how many examples were properly classified out of all the occurrences in the dataset [52]. In other words, it figures out the proportion of accurate forecasts to all of the algorithm's predictions. Accuracy is frequently utilised when the classes in the dataset are evenly distributed.

Let's look at an example to comprehend accuracy better. Suppose we use a classification technique to determine if each instance in a dataset of 1000 examples belongs to class A or B. After the algorithm has run, it properly predicts 800 cases and incorrectly categorises 200 instances. In this situation, the algorithm's accuracy would be determined as follows:

$$Accuracy = \left(\frac{\text{Number of correctly predicted instances}}{\text{Total number of instances}}\right) = \left(\frac{800}{1000}\right) = 0.8 \text{ or } 80\% \quad (9.1)$$

Accuracy is a widely used statistic since accuracy offers a simple and understandable way to gauge performance. However, it is not necessarily the best metric, particularly when working with unbalanced datasets with varying examples in each class. In certain situations, accuracy might be deceptive since a high accuracy score can be obtained by consistently guessing the majority class. Therefore, depending on the particular situation and requirements, it is necessary to consider different assessment measures, such as accuracy, recall, F1 score, or area under the receiver operating characteristic (ROC) curve.

9.6.1.2 Precision

Precision is the ratio of true positives (positive instances that were successfully predicted) to false positives (positive samples that were wrongly predicted) [53]. Precision sheds light on how well a classifier can prevent false positives. The algorithm is more accurate at correctly recognising affirmative instances with higher precision.

The recall is determined by dividing the total number of true positives by the number of false negatives (missed positive events). It shows how well the classifier can recognise occurrences of positivity. An algorithm with a higher recall will likely capture more positive examples and produce fewer false negatives.

To understand precision better, let's consider an example. Suppose we have a problem of binary classification, where the algorithm predicts if an email is spam (positive class) or not (negative class). Following the algorithm's execution, 100 emails are labelled as spam, 90 of which are legitimate spam (true positives (TP))

and 10 of which are not (false positives (FP)). Accordingly, the algorithm's accuracy would be determined in this situation:

$$Precision = \left(\frac{\text{Number of (TP) predictions}}{\text{Number of (TP) predictions + Number of (FP) predictions}}\right) = \left(\frac{90}{90+10}\right) = 0.9 \text{ or } 90\% \quad (9.2)$$

However, accuracy could not fully picture a model's performance. It should be considered with other measures, such as recall, as they frequently compete. The recall measures the model's capacity to properly identify all positive cases, also known as sensitivity or true positive rate. Precision and recall must be balanced, and the F1 score, the harmonic mean of the two, is frequently used as one metric to assess a model's overall performance.

9.6.1.3 F1 Score

Precision and recall are harmonically summed to produce the F1 score. It offers a fair assessment of the classifier's performance, considering precision and recall [54]. These two criteria are combined into one value, the F1 score, which serves as a single performance indicator. Given that it prioritises accuracy and recall equally, it is helpful when the dataset is unbalanced.

Precision and recall are given equal weight in calculating the F1 score, which is the harmonic mean of both measures. It fairly evaluates a model's capacity to accurately identify positive cases (precision) and capture every positive instance (recall). The F1 score combines precision and recall to provide a metric that considers false positives and false negatives. It is calculated using the following formula:

$$\text{F1 score} = 2 \times \left(\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}\right) \quad (9.3)$$

The F1 score highlights the significance of balancing these measures by considering the harmonic mean of accuracy and recall. It is helpful when we want to take a broader view of a model's performance than just concentrating on precision or recall. The F1 score ranges between 0 and 1, where 1 represents the best possible F1 score, indicating perfect precision and recall. The F1 score is commonly used when the dataset is imbalanced or when precision and recall are equally important. For instance, it's critical to accurately identify fraudulent transactions (recall) while reducing false alarms (precision) in fraud detection.

9.6.1.4 Area Under the ROC Curve (AUC-ROC)

The classifier's performance at various classification thresholds is depicted graphically by the ROC curve [55]. It charts the ratio of true positives (recall) to false positives (specificity). The classifier's performance is indicated by the AUC-ROC, which is the area under this curve. A better classifier with greater discrimination between positive and negative examples has a higher AUC-ROC.

A higher value for the AUC-ROC, which ranges from 0 to 1, implies better performance. A model with an AUC-ROC of 1 represents a classifier that can properly

discriminate between positive and negative examples. An AUC-ROC of 0.5 indicates a classifier that doesn't do better than guessing at random.

Let's explore an example to grasp the AUC-ROC better. Consider a model for binary classification that can determine if a patient has a certain disease (positive class) or does not have the disease (negative class). On a test dataset, the model is assessed, and for various threshold values, the true positive rate (sensitivity) and false positive rate (1-specificity) are computed. The ROC curve is then drawn using these values. The area under this curve is used to calculate the AUC-ROC.

Interpreting the AUC-ROC:

- **AUC-ROC = 1:** The model has perfect discrimination ability, accurately separating positive and negative instances.
- **AUC-ROC > 0.5:** The model performs better than random guessing. A higher AUC-ROC value indicates better discrimination ability.
- **AUC-ROC = 0.5:** The model performs no better than random guessing. It has no discriminatory power.
- **AUC-ROC < 0.5:** The model performs worse than random guessing, indicating a reversal in classification.

Due to its robustness to class imbalance and ability to measure the model's performance across a wide range of thresholds, the AUC-ROC is a frequently used evaluation statistic. It is particularly helpful when the cost of false positives (FP) and false negatives (FN) is unequal.

9.6.2 Cross-Validation Techniques

Cross-validation is a technique for calculating how well an ML model will perform on new data [56]. The available dataset is divided into training and validation subsets, and the model is trained on the training set and assessed on the validation set in this method. Cross-validation is mostly used to evaluate how effectively a model generalises to new, untested data.

Cross-validation lets us get performance measurements like accuracy, precision, recall, F1 score, or area under the ROC curve (AUC-ROC). These metrics aid in evaluating the model's functionality and comprehension of how well it performs predictions on new data.

By averaging the results across several iterations, cross-validation approaches like k-fold cross-validation, stratified cross-validation, or leave-one-out cross-validation offer a systematic and reliable way to gauge the model's performance. They aid in evaluating the model's capacity for generalisation and identifying potential problems like over- or underfitting.

Cross-validation is not an optimisation approach, although it is frequently employed throughout the model selection and optimisation process. The comparison helps find the optimal combination of hyperparameter settings or models.

Cross-validation testing allows us to assess the models' performance, allowing us to choose the one that works the best for the given situation.

9.6.3 Hyperparameter Optimisation Techniques

Hyperparameter optimisation techniques determine the best values for hyperparameters in machine learning models. Hyperparameters are configuration options chosen before training the model and not learned during training. They have a substantial impact on the model's functionality and generalisation ability.

Here are some well-known methods for hyperparameter optimisation:

9.6.3.1 Grid Search

Grid search is a tuning method to determine the ideal hyperparameter values. It thoroughly examines all potential combinations of hyperparameter settings and assesses the model's performance. Each hyperparameter's range or set of values must be specified when using grid search, which also analyses the model numerous times using various combinations. Then, based on a predetermined assessment criterion, it chooses the best performance combination. Grid search is simple and guarantees a thorough search over the hyperparameter space. Still, it can be computationally costly, especially when working with many hyperparameters or a wide range of potential values.

9.6.3.2 Random Search

Random search is an alternative to grid search that selects random hyperparameter combinations from predetermined distributions or ranges. Random search chooses a predetermined amount of random combinations and evaluates the model's performance for each combination rather than exhaustively exploring the full hyperparameter space. When the search space is vast, or some hyperparameters' effects are more substantial than others, this method may be more effective than grid search. Unlike exhaustive grid search, random search enables the study of several portions of the hyperparameter space, which might improve performance.

9.6.3.3 Bayesian Optimisation

Bayesian optimisation, a sequential model-based optimisation method, considers previous information about the objective function under optimisation. It creates a substitute model that roughly approximates the goal function, usually using a Gaussian process. The surrogate model chooses the next set of hyperparameters based on an acquisition function that balances exploration (testing various hyperparameters) with exploitation (exploiting good locations). By updating the surrogate model and improving the estimation of the ideal hyperparameters, Bayesian optimisation continuously assesses the model's performance for certain combinations of

hyperparameters. This method saves time because it narrows the search to areas most likely to have the best hyperparameter values based on the surrogate model.

9.6.3.4 Evolutionary Algorithms

Evolutionary algorithms apply the principles of natural evolution to find the best hyperparameters. To evolve and enhance the population across iterations, they start with a population of potential solutions (hyperparameter combinations) and apply mechanisms including mutation, crossover, and selection. The effectiveness of each possible solution is assessed, and the most suitable ones are chosen for replication in the following generation. Up till a good answer is discovered, this iterative procedure is continued. Evolutionary algorithms may handle non-differentiable or non-continuous search spaces and investigate many hyperparameter combinations. They can, however, be computationally costly, particularly for challenging optimisation issues.

9.7 Learning Outcomes of the Chapter

- **Machine Learning for Big Data Analytics:** Overview of machine learning for big data analytics.
- **Supervised Machine Learning for Big Data Analytics:** Gaining the insights into the challenges of applying supervised machine learning to big data analytics, pre-processing big data, and popular supervised machine learning algorithms for big data analytics.
- **Unsupervised Machine Learning for Big Data Analytics:** Detailing the unsupervised machine learning algorithms encompassing clustering, dimensionality reduction techniques, and other relevant approaches.
- **Neural Networks Algorithms:** Learning the components and types of neural networks.
- **Probabilistic Learning for Big Data Analytics:** Understanding the fundamentals of probabilistic learning, scalable algorithms, and applications in big data analytics.
- **Performance Evaluation and Optimisation Techniques:** Understanding the evaluation metrics for supervised machine learning algorithms, cross-validation techniques, and hyperparameter optimisation techniques.

References

1. M.I. Jordan, T.M. Mitchell, Machine learning: trends, perspectives, and prospects. Science **349**(6245), 255–260 (2015)
2. S. Garg, K. Kaur, G.S. Aujla, G. Kaddoum, P. Garigipati, M. Guizani, Trusted explainable AI for 6G-enabled edge cloud ecosystem. IEEE Wirel. Commun. **30**(3), 163–170 (2023)

3. P. Gupta, A. Sharma, R. Jindal, Scalable machine-learning algorithms for big data analytics: a comprehensive review. Wiley Interdisc. Rev. Data Min. Knowl. Discov. **6**(6), 194–214 (2016)
4. S. Mittal, O.P. Sangwan, Big data analytics using machine learning techniques, in *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. IEEE (2019), pp. 203–207
5. A. Zheng, A. Casari, *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists* (O'Reilly Media, Inc., 2018)
6. J. Cai, J. Luo, S. Wang, S. Yang, Feature selection in machine learning: a new perspective. Neurocomputing **300**, 70–79 (2018)
7. D.C. Montgomery, E.A. Peck, G.G. Vining, *Introduction to Linear Regression Analysis* (Wiley, 2021)
8. T.G. Nick, K.M. Campbell, Logistic regression. *Topics in Biostatistics* (2007), pp. 273–301
9. J.F. Magee, *Decision Trees for Decision Making* (Harvard Business Review Brighton, MA, USA, 1964)
10. G. Biau, E. Scornet, A random forest guided tour. Test **25**, 197–227 (2016)
11. R.G. Brereton, G.R. Lloyd, Support vector machines for classification and regression. Analyst **135**(2), 230–267 (2010)
12. I.B.A. Peling, I.N. Arnawan, I.P.A. Arthawan, I.G.N. Janardana, Implementation of data mining to predict period of students study using Naive Bayes algorithm. Int. J. Eng. Emerg. Technol. **2**(1), 53 (2017)
13. M. Irfan, W. Uriawan, O.T. Kurahman, M. Ramdhani, I. Dahlia, Comparison of Naive Bayes and k-nearest neighbor methods to predict divorce issues, in *IOP Conference Series: Materials Science and Engineering*, vol. 434, no. 1 (IOP Publishing, 2018), p. 012047
14. T. Widiyaningtyas, M.I.W. Prabowo, M.A.M. Pratama, Implementation of k-means clustering method to distribution of high school teachers, in *2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*. IEEE (2017), pp. 1–6
15. A. Kassambara, *Practical Guide to Cluster Analysis in R: Unsupervised Machine Learning*, vol. 1 (Sthda, 2017)
16. W.-T. Wang, Y.-L. Wu, C.-Y. Tang, M.-K. Hor, Adaptive density-based spatial clustering of applications with noise (DBSCAN) according to data, in *2015 International Conference on Machine Learning and Cybernetics (ICMLC)*, vol. 1. IEEE (2015), pp. 445–451
17. D.A. Reynolds et al., Gaussian mixture models. *Encyclopedia of Biometrics*, vol. 741, no. 659–663 (2009)
18. N. Kambhatla, T.K. Leen, Dimension reduction by local principal component analysis. Neural Comput. **9**(7), 1493–1516 (1997)
19. A. Maćkiewicz, W. Ratajczak, Principal components analysis (PCA). Comput. Geosci. **19**(3), 303–342 (1993)
20. M.C. Cieslak, A.M. Castelfranco, V. Roncalli, P.H. Lenz, D.K. Hartline, t-distributed stochastic neighbor embedding (t-sne): a tool for eco-physiological transcriptomic analysis. Marine Genomics **51**, 100723 (2020)
21. Y.S. Koh, S.D. Ravana, Unsupervised rare pattern mining: a survey. ACM Trans. Knowl. Discov. Data (TKDD) **10**(4), 1–29 (2016)
22. F.T. Liu, K.M. Ting, Z.-H. Zhou, Isolation forest, in *2008 Eighth IEEE International Conference on Data Mining*. IEEE (2008), pp. 413–422
23. D. Xu, Y. Wang, Y. Meng, Z. Zhang, An improved data anomaly detection method based on isolation forest, in *2017 10th International Symposium on Computational Intelligence and Design (ISCID)*, vol. 2. IEEE (2017), pp. 287–291
24. T.K. Moon, The expectation-maximization algorithm. IEEE Signal Process. Mag. **13**(6), 47–60 (1996)
25. I. Kerenidis, J. Landman, Quantum spectral clustering. Phys. Rev. A **103**(4), 042415 (2021)
26. Y. Cabanes, F. Barbaresco, M. Arnaudon, J. Bigot, Unsupervised machine learning for pathological radar clutter clustering: the p-mean-shift algorithm, in *C&ESAR 2019* (2019)

27. J. Yang, S. Rahardja, P. Fränti, Mean-shift outlier detection and filtering. Pattern Recognit. **115**, 107874 (2021)
28. J. Lawrence, *Introduction to Neural Networks* (California Scientific Software, 1993)
29. J. Sietsma, R.J. Dow, Creating artificial neural networks that generalize. Neural Netw. **4**(1), 67–79 (1991)
30. T.L. Fine, *Feedforward Neural Network Methodology* (Springer, 2006)
31. J. Wu, Introduction to convolutional neural networks. *National Key Lab for Novel Software Technology*, vol. 5, no. 23 (Nanjing University, China, 2017), p. 495
32. L.R. Medsker, L. Jain, Recurrent neural networks. Design Appl. **5**(64–67), 2 (2001)
33. K. Qadeer, W.U. Rehman, A.M. Sheri, I. Park, H.K. Kim, M. Jeon, A long short-term memory (lstm) network for hourly estimation of pm2. 5 concentration in two cities of South Korea. Appl. Sci. **10**(11), 3984 (2020)
34. P. Malhotra, L. Vig, G. Shroff, P. Agarwal et al., Long short term memory networks for anomaly detection in time series, in *Esann*, vol. 2015 (2015), p. 89
35. L. Gonog, Y. Zhou, A review: generative adversarial networks, in *2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA)*. IEEE (2019), pp. 505–510
36. A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, A.A. Bharath, Generative adversarial networks: an overview. IEEE Signal Process. Mag. **35**(1), 53–65 (2018)
37. W. Wang, Y. Huang, Y. Wang, L. Wang, Generalized autoencoder: a neural network framework for dimensionality reduction, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (2014), pp. 490–497
38. T. Kohonen, The self-organizing map. Proc. IEEE **78**(9), 1464–1480 (1990)
39. W. Wen, U. Demirbaga, A. Singh, A. Jindal, R.S. Batth, P. Zhang, G.S. Aujla, Health monitoring and diagnosis for geo-distributed edge ecosystem in smart city. IEEE Internet Things J. (2023)
40. A. Singh, S. Garg, R. Kaur, S. Batra, N. Kumar, A.Y. Zomaya, Probabilistic data structures for big data analytics: a comprehensive review. Knowl. Based Syst. **188**, 104987 (2020)
41. T.L. Fine, *Theories of Probability: An Examination of Foundations* (Academic Press, 2014)
42. K. Mitra, S. Saguna, C. Åhlund, R. Ranjan, Alpine: a Bayesian system for cloud performance diagnosis and prediction, in *2017 IEEE International Conference on Services Computing (SCC)*. IEEE (2017), pp. 281–288
43. D.M. Blei, A. Kucukelbir, J.D. McAuliffe, Variational inference: a review for statisticians. J. Am. Stat. Assoc. **112**(518), 859–877 (2017)
44. F. Dellaert, The expectation maximization algorithm. College of Computing (Georgia Institute of Technology, 2002)
45. C.M. Carlo, Markov chain Monte Carlo and Gibbs sampling. Lect. Notes EEB **581**(540), 3 (2004)
46. C. Nemeth, P. Fearnhead, Stochastic gradient Markov chain Monte Carlo. J. Am. Stat. Assoc. **116**(533), 433–450 (2021)
47. W. Neiswanger, C. Wang, E. Xing, Asymptotically exact, embarrassingly parallel MCMC, arXiv preprint arXiv:1311.4780 (2013)
48. D.A. Nguyen, K.A. Nguyen, C.H. Nguyen, K. Than et al., Boosting prior knowledge in streaming variational bayes. Neurocomputing **424**, 143–159 (2021)
49. D. Greene, P. Cunningham, R. Mayer, Unsupervised learning and clustering. *Machine Learning Techniques for Multimedia: Case Studies on Organization and Retrieval* (2008), pp. 51–90
50. F.E. Jamiy, A. Daif, M. Azouazi, A. Marzak, The potential and challenges of big data-recommendation systems next level application, arXiv preprint arXiv:1501.03424 (2015)
51. D. Khurana, A. Koli, K. Khatter, S. Singh, Natural language processing: state of the art, current trends and challenges. Multimedia Tools Appl. **82**(3), 3713–3744 (2023)
52. M. Hossin, M.N. Sulaiman, A review on evaluation metrics for data classification evaluations. Int. J. Data Mining Knowl. Manag. Process **5**(2), 1 (2015)
53. T. Saito, M. Rehmsmeier, The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PloS One **10**(3), e0118432 (2015)

54. H. Huang, H. Xu, X. Wang, W. Silamu, Maximum f1-score discriminative training criterion for automatic mispronunciation detection. IEEE/ACM Trans. Audio Speech Lang. Process. **23**(4), 787–797 (2015)
55. Z.H. Hoo, J. Candlish, D. Teare, What is an ROC curve? (2017), pp. 357–359
56. G.H. Golub, U. Von Matt, Generalized cross-validation for large-scale problems. J. Comput. Graph. Stat. **6**(1), 1–34 (1997)

Further Reading

57. S. Suthaharan, *Machine Learning Models and Algorithms for Big Data Classification: Thinking with Examples for Effective Learning*, vol. 36 (Springer, 2015)
58. J. Brownlee, *Probability for Machine Learning: Discover How to Harness Uncertainty with Python* (Machine Learning Mastery, 2019)
59. C. Molnar, *Interpretable Machine Learning*. Lulu. com (2020)

Real-World Big Data Analytics Case Studies

10

Every company has big data in its future and every company will eventually be in the data business.

—Thomas H. Davenport

This chapter unfolds a panoramic view across diverse sectors, unveiling the transformative impact of big data analytics on real-world challenges. The exploration commences in the government sector, where data-driven governance enhances public services, enables predictive analytics for smart city planning, fortifies security and surveillance, and even extends to election forecasting and voter analytics. Transitioning to the healthcare industry, the chapter delves into the revolutionary role of big data analytics in tailoring treatments through precision medicine and predicting and preventing disease outbreaks. The entertainment industry takes centre stage, showcasing applications such as content personalization, recommendation systems, box office predictions, revenue optimization, and audience engagement through social media analytics. The banking sector comes to life with risk assessment, credit scoring, customer relationship management, personalization, fraud detection, security, and strategic decision-making. The retail industry follows suit, emphasising inventory management, demand forecasting, customer segmentation, personalization, supply chain optimization, and in-store analytics. The chapter finally highlights the energy and utilities sector by illuminating applications in grid management, smart grids, predictive maintenance, asset optimization, energy generation, renewable integration, energy efficiency, demand response, and environmental sustainability.

Ü. Demirbaga et al., *Big Data Analytics*,
https://doi.org/10.1007/978-3-031-55639-5_10

10.1 Government Sector

Integrating big data analytics has ushered in an era of data-driven governance in the government sector. Government entities worldwide increasingly harness vast and complex datasets to inform policy decisions, enhance public services, and address critical challenges [1]. By leveraging data-driven insights, governments improve service delivery and resource allocation while bolstering security, optimising urban planning, and refining electoral strategies. This section explores real-world case studies and applications within the government sector, highlighting how big data analytics revolutionises governance across various domains, from city planning and security to public service enhancement and election forecasting.

10.1.1 Enhancing Public Services Through Data-Driven Governance

The concept of data-driven governance constitutes a pivotal paradigm shift within contemporary governance, signifying a profound transformation in the methodologies employed to provide public services. This transformation is distinguished by the synergistic amalgamation of computational techniques, robust data acquisition, and processing frameworks, and the development of policy formulations, all collectively defining this transformative trajectory [2].

The rigorous examination of vast and heterogeneous datasets from many public service sectors is at the core of the data-driven governance concept. These statistics include a variety of topics, such as urban planning, transportation, social welfare, and demographics of the populace. Governmental organisations capture priceless insights into temporal trends, anomalies, and forecast patterns using computer approaches like machine learning and predictive modelling. Because of this analytical skill, proactive policies can be created, and resources may be allocated accurately. Predictive analytics, for instance, can be used in the healthcare industry to identify areas that are likely to see disease outbreaks, enabling proactive steps to protect the public's health [3].

Furthermore, citizen-centric governance models and data-driven government are perfectly compatible. By utilising data-driven insights, governments may modify public services to meet their citizens' changing demands and preferences. Digital interfaces and feedback systems increase citizen participation, encouraging iterative improvement of public service offerings [4]. Additionally, data-driven governance is seen in public safety and security by implementing sophisticated surveillance and threat detection systems supported by machine learning algorithms for anomaly detection and risk assessment [5].

10.1.2 Predictive Analytics for Smart City Planning

Predictive analytics is a sophisticated data-driven methodology employed in urban planning to enhance the efficacy of smart city development. It leverages historical data, real-time information, and advanced statistical models to forecast future trends

and patterns, aiding city planners in making informed decisions [6]. This approach is instrumental in optimising resource allocation, infrastructure development, and service delivery within urban environments. By analysing diverse datasets encompassing demographics, transportation, energy consumption, and environmental factors, predictive analytics empowers municipalities to address urban challenges and anticipate future demands proactively.

Optimising resource use is one of the main advantages of predictive analytics in smart city design. By analysing historical data, urban planners can spot trends in population increase, transportation congestion, and energy usage [7]. Through this knowledge, they may effectively manage resources by constructing transport systems to handle anticipated increases in traffic or using energy-saving technology to meet rising energy needs. Predictive analytics also helps spot future bottlenecks and weaknesses in urban infrastructure, enabling proactive measures to improve resilience [8].

Predictive analytics is also essential to improving public services in smart cities. City authorities can identify locations vulnerable to accidents or disease outbreaks by analysing public health, safety, and emergency response data. This makes allocating funds for early emergency reactions and prevention actions possible. Personalised services and suggestions based on user behaviour and preferences can also increase citizen involvement and raise the standard of living in smart cities [9].

10.1.3 Security and Surveillance: Big Data in Government

Big data use in government, particularly in security and surveillance, represents a paradigm change in how policymakers see the maintenance of national security and public safety. Big data, which consists of enormous and complicated datasets, has emerged as a priceless tool for government organisations entrusted with protecting their citizens and assets. This transformation requires systematically collecting, processing, and analysing large data streams from several sources, including sensors, cameras, social media, and communication networks, to improve decision-making, threat detection, and law enforcement activities [10].

Enhancing predictive capacities is one of the main benefits of using big data in government security and surveillance. By combining and analysing enormous databases, authorities can find patterns, abnormalities, and trends that could otherwise go undetected. This predictive capability enables governments to take proactive steps to minimise risks and protect the safety and stability of their territories by allowing for the preemptive identification of security threats, cyberattacks, and criminal activity [11].

Using big data in government security and surveillance, a deeper knowledge of population behaviour and movement is also possible. This includes monitoring people's whereabouts, analysing the tone of social media posts, and identifying potential threats. By assisting in placing personnel, resources, and actions where they are most required, these insights help establish focussed and effective security plans [12]. Big data use in this setting also prompts questions about privacy, civil liberties, and the ethical use of data, requiring a careful balance between security requirements and personal freedoms.

10.1.4 Election Forecasting and Voter Analytics

Political science and data analytics encompass the multifaceted subjects of election predictions and voter analytics. They deftly reveal the forces that control voting procedures and results. These fields rely on meticulous statistical modelling, systematic data collecting, and cutting-edge analytical methods [13]. Their objective is to offer significant insights into the choices made by voters, the preferences of candidates, and the likelihood of election outcomes. This project integrates data from various sources, including sentiment analysis from social media, demographics, candidate performance, and historical voting patterns. The foundation comprises predictive models, which use regression analysis and machine learning to uncover hidden links and trends. These models provide insightful information on electoral determinants and the impact of numerous variables on voter choices.

Voter analytics expands into political campaigns in a useful way. Campaign strategists use data-driven insights to design specialised messages and effectively manage resources. Campaigns can customise messaging to particular voter segments using sophisticated microtargeting techniques, maximising outreach efforts. This combination of data analytics and campaign strategy has revolutionised contemporary electioneering and highlighted the significance of data-driven decision-making in determining election outcomes [14]. Voter analytics and election forecasting, which shed insight into voter behaviour and reshape political campaigns through data-driven methods, are crucial in political science and data analysis.

Furthermore, it is crucial to stress that voter analytics significantly impacts political campaigns, a practical and high-stakes arena well beyond the theoretical spheres of academia [15]. Indeed, the strategic planning of political campaigns uses the insights gained through data-driven approaches in real-world, applicable ways. Campaign strategists are prepared to create highly targeted and precisely adapted communications plans in this area thanks to their arsenal of data-driven insights. Such tactics are specifically developed to appeal to certain voter demographics, increasing their outreach initiatives' effectiveness. The creation of advanced microtargeting techniques, which enable political campaigns to identify swing voters and distribute their resources with exactitude, has allowed them to change course with a previously unheard-of speed. This interaction between campaign strategy and data analytics has significantly altered how political canvassing is done today, illuminating the critical role that data-driven decision-making processes play in influencing the outcomes of political campaigns.

10.2 Healthcare Industry

A new age of accuracy, efficiency, and improved patient outcomes driven by data is beginning to emerge in the healthcare industry due to the integration of big data analytics. In this part, we explore the broad implications of data analytics for health care and show how they have revolutionised patient care, therapeutic approaches,

and public health programmes. The fusion of health care and big data analytics promises to fundamentally alter our approach to health care, providing unrivalled opportunities for innovation and improved healthcare delivery. These opportunities range from deepening our understanding of diseases to forecasting outbreaks and personalising treatments for each patient. The following subsections will examine particular applications in this dynamic environment to highlight how data-driven insights change the healthcare paradigm.

10.2.1 Revolutionising Healthcare with Big Data Analytics

Big data analytics' incorporation into the healthcare industry represents a significant paradigm shift that will profoundly change current healthcare procedures. The systematic gathering, processing, and analysis of vast and varied healthcare data sources fundamentally fuel this transformation. These include patient-generated data, genetics, electronic health records (EHRs), and medical imaging [16]. A patient-centric and data-driven approach to health care is now possible with the convergence of these diverse datasets, which provides previously unheard-of insights on patient health, disease trends, and treatment effectiveness.

Diagnostics and prognostics are critical aspects of this transforming journey. Big data analytics enables the creation of advanced machine learning models to identify complex patterns and correlations in medical data. These models allow medical personnel to stratify risk variables accurately, improve early illness identification, and predict patient outcomes [17]. For instance, predictive algorithms locate critical tumour indicators in cancer and customise treatment plans for each patient, increasing therapeutic effectiveness while reducing side effects.

Big data analytics, moreover, has significantly changed healthcare management and resource allocation [18]. The allocation of resources, hospital operations, and healthcare spending are now precisely calibrated by hospital managers and lawmakers using data-driven insights. This effective resource allocation is made possible by developing real-time patient admissions, discharges, and bed utilisation monitoring. This strategy allows healthcare facilities to operate fully, minimising patient wait times and resource waste. As a result, healthcare organisations are better equipped to deliver prompt and effective treatment, consistent with their overarching objective of enhancing patient experiences while preserving cost-effectiveness.

10.2.2 Precision Medicine: Tailoring Treatments with Data

Precision medicine is a cutting-edge healthcare method that uses big data analytics to adapt medical interventions and therapies for each patient individually [19]. This section explores how data-driven insights reshape the medical industry and enable more individualised and efficient health care. Precision medicine understands that each patient is unique and that genetic variances, lifestyle choices, and other individual features may prevent what works for one person from being suited for another.

Big data analytics plays a crucial role in enabling precision medicine by:

- **Genomic Data Analysis**: Massive volumes of data are produced by genomic sequencing and big data analytics can be used to find genetic variants linked to specific diseases [20]. By examining a patient's genetic composition, Doctors can identify the most likely effective treatments or medications with negative side effects.
- **Patient Profiling**: Big data analytics uses real-time health monitoring, genetic data, lifestyle data, and medical history to construct comprehensive patient profiles [21]. This profile enables medical professionals to decide on treatment approaches, medication dosages, and customised preventive measures to meet each patient's unique needs.
- **Treatment Personalisation**: Healthcare professionals can use machine learning algorithms to forecast a patient's response to a certain therapy or treatment [22]. This enables treatment strategies to be tailored to maximise effectiveness and reduce negative effects.
- **Early Detection**: Big data analytics can spot early warning signs of diseases or ailments before symptoms appear. Early detection makes it possible for proactive therapies, which may stop the progression of diseases and enhance patient outcomes [23].

10.2.3 Disease Outbreak Prediction and Prevention

Big data analytics is used for disease outbreak prediction and prevention to foresee and slow the spread of diseases. Healthcare organisations and authorities can predict possible epidemics by combining and analysing various data sources, such as travel data, social media trends, and medical records. Using real-time monitoring and advanced algorithms, it is possible to identify strange patterns and symptoms early on and take quick action [24]. Epidemiological models based on these data insights help with illness trajectory prediction and resource allocation optimisation. In addition to improving public health preparation, this data-driven strategy is essential for reducing the effects of infectious diseases, eventually saving lives and protecting communities.

Here are some key points to predict and prevent the disease outbreak:

- **Data Sources**: To predict disease outbreaks, big data analytics uses a variety of data sources, including travel data, social media, medical records, and environmental factors. These resources offer insightful information on the emergence and transmission of illnesses.
- **Early Warning Systems**: Advanced algorithms examine data in real time for unexpected patterns or clusters of symptoms, assisting in the early detection of illness epidemics before they spread widely. It is possible to respond quickly and take containment measures thanks to this early warning system.

- **Epidemiological Modelling**: Big data analytics can create complex epidemiological models that simulate disease spread depending on population density, travel habits, and weather conditions. These models help in outbreak trajectory prediction and intervention planning.
- **Resource Allocation**: Public health authorities can more effectively allocate resources during outbreaks with data-driven insights, including sending medical equipment, vaccines, and medical staff to high-risk areas.
- **Public Awareness Campaigns**: Healthcare organisations can conduct targeted public awareness efforts to inform the populace about preventive measures and symptoms connected with the outbreak by studying social media and Internet search trends.

10.3 Entertainment Industry

Big data analytics is revolutionising the entertainment industry by providing unprecedented understanding and strategic insight into audience behaviour, content creation, and distribution. With the massive amount of data produced by digital platforms, streaming services, and social media, entertainment businesses can now use data analytics to optimise marketing efforts, create content tailored to individual preferences, and predict trends [25]. Big data analytics is reshaping the entertainment landscape by enabling industry stakeholders to produce more compelling and profitable content while enhancing the overall entertainment experience. Examples include suggesting personalised content to viewers, forecasting box office success, or engaging with audiences on social media.

10.3.1 Content Personalization and Recommendation Systems

Big data analytics have transformed personalisation and recommendation technologies in the entertainment sector. It uses cutting-edge algorithms and data analysis to develop tailored content recommendations for users. These systems can forecast individual interests and make content suggestions based on user choices, viewing patterns, and demographic data. Collaborative filtering and deep learning models can be used to accurately forecast user preferences, increasing user engagement and customer loyalty. The entertainment sector uses content personalisation and recommendation algorithms to deliver content that appeals to viewers, creating a more engaging and gratifying entertainment experience while maximising the potential for content consumption and revenue creation.

10.3.2 Box Office Predictions and Revenue Optimization

In the entertainment sector, big data analytics significantly impacts methods for revenue optimisation and box office forecasting. Sophisticated data models and machine learning techniques must be used to estimate how well films will perform in theatres. These analytical tools allow studios and distributors to make data-informed decisions about marketing campaigns, release dates, and distribution channels by considering numerous criteria like genre, cast, release schedule, and historical statistics. This data-driven strategy not only aids in estimating future revenue but also reduces financial risks, resulting in ultimately more fruitful endeavours and successful productions. Big data analytics is essential for improving the sustainability and profitability of movie releases, as well as for optimising the financial aspects of the entertainment sector.

10.3.3 Audience Engagement and Social Media Analytics

Big data analytics has emerged as a crucial tool for raising audience engagement and utilising social media analytics in the entertainment sector. It entails tracking user sentiment, engagement data, and real-time conversations around entertainment material on social media sites. The sector learns important information about audience reactions by utilising sentiment analysis and social listening solutions. These insights allow producers and marketers to efficiently alter tactics, address issues, and take advantage of good trends. The entertainment sector gains crucial marketing information and strengthens customer loyalty by engaging with its audience on social media. Future content creation and marketing initiatives are informed by this data-driven strategy, which results in more focussed and interesting interactions with the audience and raises satisfaction and engagement levels generally.

10.4 Banking Sector

Big data analytics is becoming a game-changing force in the banking industry, profoundly changing how financial institutions run their businesses, reach key decisions, and engage with their clientele. Its numerous financial applications include improved risk management, better customer service, cutting-edge fraud detection techniques, and streamlined strategic planning [26]. The banking sector reduces risks and improves operational processes through effective data use. Still, it also builds stronger client relationships, assures regulatory conformance, and promotes innovation in the digital finance era.

10.4.1 Risk Assessment and Credit Scoring

Big data analytics completely transforms Credit risk evaluation in banking. Banks use sophisticated data models and machine learning algorithms to analyse huge datasets

containing customer financial histories, transaction records, and external economic factors. With these instruments' aid, banks can establish loan eligibility, precisely measure creditworthiness, and set interest rates. Additionally, they enable financial institutions to continuously assess and modify risk profiles, improving portfolio stability overall and lowering the risk of default on loans.

10.4.2 Customer Relationship Management (CRM) and Personalization

Customer-centricity is crucial in the banking industry, and big data analytics empowers banks to offer customised financial products and services. Banks use customer data to segment their clientele, build detailed customer profiles, and forecast customers' financial needs and behaviour. Banks can provide pertinent product recommendations, anticipate consumer questions, and improve marketing campaigns using CRM platforms and predictive analytics. This data-driven strategy boosts cross-selling opportunities, long-term loyalty, and customer involvement while increasing customer engagement.

10.4.3 Fraud Detection and Security

Big data analytics is essential for detecting and preventing fraud, and maintaining the security and integrity of financial transactions is a major responsibility for banks. Banks use machine learning models, anomaly detection techniques, and real-time transaction monitoring to spot odd trends or questionable behaviour. Financial institutions may immediately protect their consumers and assets by quickly identifying suspected fraud. Furthermore, big data analytics enables banks to continuously modify and enhance their security protocols in response to changing threats, allowing them to remain ahead of new fraud strategies.

10.4.4 Strategic Decision-Making and Regulatory Compliance

Banks must negotiate challenging compliance regulations due to the regulatory environment's constant change. Through automated reporting, accurate data assurance, and easier audit trails, big data analytics helps banks manage regulatory compliance. Big data analytics also enables leaders to make strategic decisions by thoroughly understanding market trends, consumer behaviour, and operational performance. While still adhering to legal requirements, banks can allocate resources efficiently, plan for future expansion, and create cutting-edge financial products.

10.5 Retail Industry

Big data analytics are being seamlessly integrated into the retail industry, which is causing a significant transition to take place. This change spans various retail-related applications for data-driven insights, including improving customer experiences, fine-tuning marketing tactics, and optimising inventory control and supply chain processes [27]. Big data analytics is used by retailers to enable them to make data-driven decisions that are highly informed, leading to increased operational efficiency as well as a dynamic environment that fosters innovation. The application of big data analytics is influencing the future of retail in a sector where an in-depth knowledge of consumer behaviour and market dynamics is crucial.

10.5.1 Inventory Management and Demand Forecasting

In the retail industry, the pivotal role of big data analytics in demand forecasting and inventory management cannot be overstated. Retailers employ a sophisticated approach that draws upon a multifaceted array of data sources, including historical sales data, real-time inventory status, and external factors such as weather patterns and economic indicators, to forecast future demand meticulously. This multifaceted approach to demand forecasting is underpinned by cutting-edge algorithms and advanced machine learning models, which serve as instrumental tools for optimising inventory levels and effecting cost reductions by minimising carrying costs and mitigating situations of stockouts or overstocking. The profound implications of this data-driven approach reverberate throughout the retail landscape as retailers are empowered to refine inventory management practices, enhancing customer satisfaction, reducing wastage, and ultimately augmenting profitability by precisely aligning their inventory with the dynamic ebb and flow of consumer demand.

10.5.2 Customer Segmentation and Personalization

It is impossible to exaggerate how crucial it is for the retail sector to be customer-centric, and big data analytics integration has become a key tool in this effort. Retailers have started a data-driven journey, including thoroughly studying large datasets, ranging from customer purchase history to online behavioural patterns and containing vital demographic data. This complex investigation results from the excellent segmentation of their consumer base, a crucial part of creating personalised shopping experiences. Retailers use these data-driven insights to adapt their marketing campaigns, suggest products that closely match unique client tastes, and adjust pricing strategies to target particular customer segments. This personalised strategy has repercussions across the retail landscape since it improves consumer engagement, fosters enduring loyalty, and considerably boosts conversion rates, all of which ultimately support the primary objective of revenue development in the retail industry.

10.5.3 Supply Chain Optimization and Vendor Management

Big data analytics plays a crucial role in optimising supply chain operations in the complex world of the retail industry by enabling a thorough and insightful perspective into the whole supply chain ecosystem. In this situation, merchants use their vast data resources to systematically monitor the complex flow of goods, allowing them to spot any bottlenecks quickly and proactively anticipate and handle any potential interruptions. This proactive strategy results from a more simplified and effective inventory replenishment process, appreciable lead time reductions, and the development of more effective and mutually beneficial vendor relationships. Retailers can improve vendor management and put themselves in a better position to engage in more advantageous negotiations by wisely utilising the possibilities of data analytics, which also ensures a steady and reliable flow of goods throughout the supply chain. These carefully honed supply chain optimisation initiatives have far-reaching effects, including cost reduction, improved order fulfilment, and a significant improvement in overall operational efficiency, giving retailers better tools to survive and thrive in the competitive retail environment.

10.5.4 Enhanced Customer Experience Through In-Store Analytics

Empowered by in-store analytics, big data's capabilities mark a paradigm shift in the retail industry. Retailers use data collected from various sources, such as in-store cameras, sensors, and consumer Wi-Fi usage, in this context to get profound insights into customer behaviour patterns inside the walls of actual brick-and-mortar stores. Retailers gain actionable data to optimise key aspects of the in-store environment by utilising sophisticated techniques, including heatmap analysis, dwell-time assessment, and observing customer flow patterns. These include thoughtful product placements, efficient staff scheduling, and retail layouts. This data-driven strategy is created to develop a more engaging, effective, and customer-centric in-store experience, which results in increased customer satisfaction and a perceptible increase in sales, positioning retailers to succeed in an evolving and competitive retail environment.

10.6 Energy and Utilities

Big data analytics is transforming the energy and utilities sector, revolutionising processes by utilising enormous datasets. It allows energy suppliers and utility businesses to manage grids more effectively, forecast energy production, maintain assets better, increase energy efficiency, and promote sustainability. Predictions based on data-driven analysis help integrate renewable energy sources, while real-time insights facilitate predictive maintenance, load balancing, and outage prediction [28]. This analytical method promotes energy efficiency programmes and demand response tactics while extending asset lifespans, decreasing downtime, and improving resource

allocation. Monitoring emissions and adhering to regulations promote environmental sustainability, equipping the industry to navigate the changing energy landscape with data-informed decisions and increased operational efficiency.

10.6.1 Grid Management and Smart Grids

Big data analytics plays a crucial part in grid management and the development of smart grids within the dynamic environment of the energy and utilities sector. Smart metres, sensors, and various data sources are all included in the sophisticated integration of data analytics, which provides real-time insights into grid performance, discerns complex demand patterns, and anticipates future operational concerns. A greatly improved grid stability and efficiency result from the orchestration of load balancing, predictive maintenance, and outage prognosis, all aspects of this multidimensional data analysis. A disruptive force, smart grids result from the synergy of data analytics. They pave the way for a higher paradigm of energy distribution marked by decreased waste, increased reliability, and improved overall operational efficacy.

10.6.2 Predictive Maintenance and Asset Optimization

It is impossible to exaggerate the crucial role of big data analytics in asset optimisation in the energy and utilities sector. It has significant ramifications for predictive maintenance plans, which rely on carefully examining data gathered from infrastructure and equipment in the past and present. This analytical technique excels at anticipating maintenance needs before equipment failure using complex predictive algorithms, which reduce downtime, lower operational costs, and extend the useful lives of assets. As a result of such diligent asset optimisation, utility firms have an advantage in their active endeavours thanks to optimal asset performance, wise resource allocation, and careful capital expenditure planning.

10.6.3 Energy Generation and Renewable Integration

Energy production and the seamless integration of renewable energy sources beckon profound changes fueled by big data analytics. The rigorous study of a wide range of data sources, including climatic forecasts, sensor inputs from equipment, and perceptible energy use patterns, distinguishes these crucial projects-the precise estimates of energy production and consumption trends result from this extensive data analysis. As a result, the energy grid's integration of renewable energy sources, such as solar and wind, is carried out with an unmatched efficiency. As a result, there will be less reliance on traditional fossil fuels and a stronger commitment to sustainable energy practices. This also marks the beginning of an era with reduced emissions.

10.6.4 Energy Efficiency and Demand Response

The energy and utilities sector actively promotes energy efficiency and the clarification of demand response techniques under the transformative influence of big data analytics. The thorough examination of past consumption statistics and current insights forms the basis of this admirable endeavour. With the insights gained from data analytics, utility companies are empowered to give their valued clients individualised energy-saving recommendations and incentives. The benefits of such a data-driven strategy go beyond cost savings because it promotes effective control of peak energy demand, strengthening the sustainability and resilience of the energy system.

10.6.5 Environmental Sustainability and Emissions Reduction

The imperatives of environmental sustainability and emissions reduction loom large in today's energy and utilities industry discourse. The Centre stage examines sizable datasets that include various aspects of energy production, use, and emissions. These analytical findings serve as a compass, guiding methods for reducing carbon footprints and pointing towards areas needing improvement. Data-driven insights have the potential to be used to improve energy sources, monitor emissions thoroughly, and uphold strict environmental standards, among other things. When these efforts are combined, they result in a solid framework that is committed to a sustainable energy landscape and a brighter, cleaner, and more environmentally conscious future.

10.7 Learning Outcomes of the Chapter

- **Explore Real-World Applications**: Delving into practical applications of big data analytics in diverse sectors, ranging from government and health care to entertainment, banking, retail, and energy.
- **Sector-Specific Insights**: Gaining insights into how big data is utilised in different services and their impact on such industries.

References

1. J.C. Bertot, H. Choi, Big data and e-government: issues, policies, and recommendations," in *Proceedings of the 14th Annual International Conference on Digital Government Research* (2013), pp. 1–10
2. E. Gummesson, Case theory in business and management: Reinventing case study research, in *Case Theory in Business and Management* (2017), pp. 1–368

3. A. Jindal, A. Dua, N. Kumar, A.V. Vasilakos, J.J. Rodrigues, An efficient fuzzy rule-based big data analytics scheme for providing healthcare-as-a-service, in *2017 IEEE International Conference on Communications (ICC)* (2017), pp. 1–6
4. J.C. Bertot, E. Estevez, T. Janowski, Digital public service innovation: Framework proposal, in *Proceedings of the 9th International Conference on Theory and Practice of Electronic Governance* (2016), pp. 113–122
5. S. Xu, Y. Qian, R.Q. Hu, Data-driven network intelligence for anomaly detection. IEEE Network **33**(3), 88–95 (2019)
6. K. Soomro, M.N.M. Bhutta, Z. Khan, M.A. Tahir, Smart city big data analytics: An advanced review. Wiley Interdiscip. Rev. Data Min. Knowl. Discov. **9**(5), e1319 (2019)
7. B. Wilson, A. Chakraborty, The environmental impacts of sprawl: Emergent themes from the past decade of planning research. Sustainability **5**(8), 3302–3327 (2013)
8. Y. Kaluarachchi, Potential advantages in combining smart and green infrastructure over silo approaches for future cities. Front. Eng. Manage. **8**, 98–108 (2021)
9. L. Quijano-Sánchez, I. Cantador, M.E. Cortés-Cediel, O. Gil, Recommender systems for smart cities. Inf. Syst. **92**, 101545 (2020)
10. Y. El-Ghalayini, H. Al-Kandari, Big data regulatory legislation: Security, privacy and smart city governance. JL Pol'y Global. **95**, 19 (2020)
11. M. Mahbub, Progressive researches on iot security: An exhaustive analysis from the perspective of protocols, vulnerabilities, and preemptive architectonics. J. Network Comput. Appl. **168**, 102761 (2020)
12. G.C. Oatley, Themes in data mining, big data, and crime analytics. Wiley Interdiscip. Rev. Data Min. Knowl. Disc. **12**(2), e1432 (2022)
13. W. Chen, A. Quan-Haase, Big data ethics and politics: Toward new understandings. Soc. Sci. Comput. Rev. **38**(1), 3–9 (2020)
14. E.F. Judge, M. Pal, *Voter Privacy and Big-data Elections*, vol. 58 (Osgoode Hall LJ, 2021), p. 1
15. F. Gilardi, *Digital Technology, Politics, and Policy-Making*. (Cambridge University Press, 2022)
16. N. Mehta, A. Pandit, M. Kulkarni, *Elements of Healthcare Big data Analytics. Big Data Analytics in Healthcare* (2020), pp. 23–43
17. S. Huang, J. Yang, S. Fong, Q. Zhao, Artificial intelligence in cancer diagnosis and prognosis: Opportunities and challenges. Cancer Lett. **471**, 61–71 (2020)
18. S. Khanra, A. Dhir, A.N. Islam, M. Mäntymäki, Big data analytics in healthcare: a systematic literature review. Enterpr. Inf. Syst. **14**(7), 878–912 (2020)
19. T. Hulsen, S.S. Jamuar, A.R. Moody, J.H. Karnes, O. Varga, S. Hedensted, R. Spreafico, D.A. Hafler, E.F. McKinney, From big data to precision medicine. Front. Med. **6**, 34 (2019)
20. A. O'Driscoll, J. Daugelaite, R.D. Sleator, Big data, hadoop and cloud computing in genomics. J. Biomed. Inf. **46**(5), 774–781 (2013)
21. M. Herrero-Zazo, T. Fitzgerald, V. Taylor, H. Street, A.N. Chaudhry, J.R. Bradley, E. Birney, V.L. Keevil, Using machine learning to model older adult inpatient trajectories from electronic health records data. Iscience **26**(1) (2023)
22. K.Y. Ngiam, W. Khor, Big data and machine learning algorithms for health-care delivery. Lancet Oncol. **20**(5), e262–e273 (2019)
23. M.I. Razzak, M. Imran, G. Xu, Big data analytics for preventive medicine. Neural Comput. Appl. **32**, 4417–4451 (2020)
24. A.N. Desai, M.U. Kraemer, S. Bhatia, A. Cori, P. Nouvellet, M. Herringer, E.L. Cohn, M. Carrion, J.S. Brownstein, L.C. Madoff et al., Real-time epidemic forecasting: challenges and opportunities. Health Sec. **17**(4), 268–275 (2019)
25. H. Lippell, Big data in the media and entertainment sectors, in *New Horizons for a Data-Driven Economy: A Roadmap for Usage and Exploitation of Big Data in Europe* (2016), pp. 245–259
26. U. Srivastava, S. Gopalkrishnan, Impact of big data analytics on banking sector: Learning for indian banks. Procedia Comput. Sci. **50**, 643–652 (2015)

27. M.G. Dekimpe, Retailing and retailing research in the age of big data analytics. Int. J. Res. Market. **37**(1), 3–14 (2020)
28. K. Zhou, C. Fu, S. Yang, Big data driven smart energy management: From big data to big insights. Renew. Sustain. Energy Rev. **56**, 215–225 (2016)

Further Reading

29. J.R. Owens, B. Femiano, J. Lentz, *Hadoop Real World Solutions Cookbook*. (Packt Publishing, 2013)
30. T. Dunning, E. Friedman, *Real-World Hadoop*. (O'Reilly Media, Inc., 2015)
31. M. Grover, T. Malaska, J. Seidman, G. Shapira, *Hadoop Application Architectures: Designing Real-world Big Data Applications*. (O'Reilly Media, Inc., 2015)

Big Data Analytics in Smart Grids

11

Smart-grid technologies allow for the integration of renewable energy into the grid as well as energy from distributed sources.

—Joe Kaeser

In this chapter, the exploration explores applying big data analytics within the smart grid domain. The journey commences with a comprehensive examination of the smart grid concept, setting the stage for a nuanced understanding. The discourse seamlessly transitions to an in-depth analysis of various analytics types viable in smart grids, intricately detailing the essential reasons driving the need for such analytical interventions. Culminating the chapter is a practical illustration showcasing the application of big data analytics—specifically, predicting societal load demand. This example serves as a tangible demonstration of how sophisticated analytics can be wielded to gain valuable insights within the dynamic landscape of smart grids.

11.1 Smart Grids

The technological reforms and improvements in the last decade have revolutionised many sectors, including smart cities. In smart cities, the key factor is to provide various services to the users, ranging from smart governance to public safety, by integrating Information and Communication Technologies (ICT). One of the major services provided in smart cities, energy management, has been of significant importance. Given its limitations, energy generation should be managed to create a demand and supply balance. However, energy demand growth has been enormous recently, which calls for quick action. Figure 11.1 shows the energy consumption growth in different sectors in the U.S. alone in recent years along with its predicted growth [1].

Ü. Demirbaga et al., *Big Data Analytics*,
https://doi.org/10.1007/978-3-031-55639-5_11

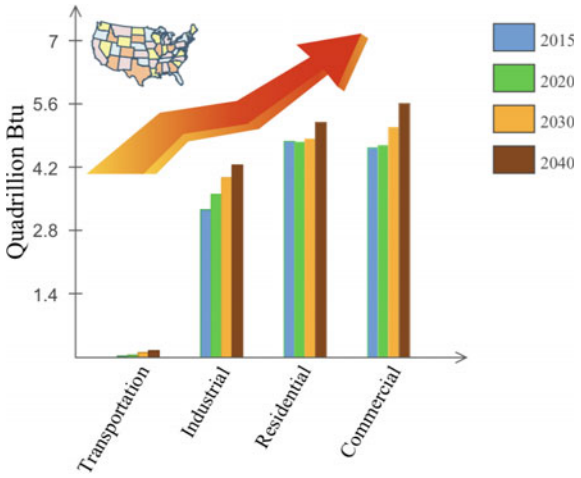


Fig. 11.1 Energy growth in different sectors

The flow of energy has also emerged from "traditional" to "smart" in smart cities, given the adoption of *smart grids*. These grids can efficiently manage the energy generated from distributed and centralised sources. Figure 11.2 shows a scenario of a smart city where different entities at different levels such as—residential, industrial & commercial users, electric vehicles, generation, transmission & distribution units are connected using communication lines (in addition to the power line), thereby, making it a complex network of energy [2]. The data gathered in this environment can be analysed to provide various solutions to the distributor system operators (DSOs), transmission system operators (TSOs), and end-users.

As seen in Fig. 11.2, different entities in the smart grid can communicate with the remote control and monitoring station (or server) located at a dedicated place or on the cloud. These entities use different communication protocols as summarised in Table 11.1. In the table, the object can be referred to as a smart home, electric vehicle, or similar and the access point can be referred to as the local controller, roadside unit, or similar. Therefore, the type of communications in Table 11.1 signify short-range, medium-range, and long-range communications, respectively.

11.2 Big Data Analytics in Smart Grid

Big data analytics, in general, can be categorised into four categories: descriptive, diagnostic, predictive, and prescriptive. *Descriptive analytics* relates to the insights on what has happened and helps to create a credible reasoning for it. This analytics type focuses primarily on finding important information from the available data that could be leveraged for further evaluation. The second type is the *Diagnostic analytics*,

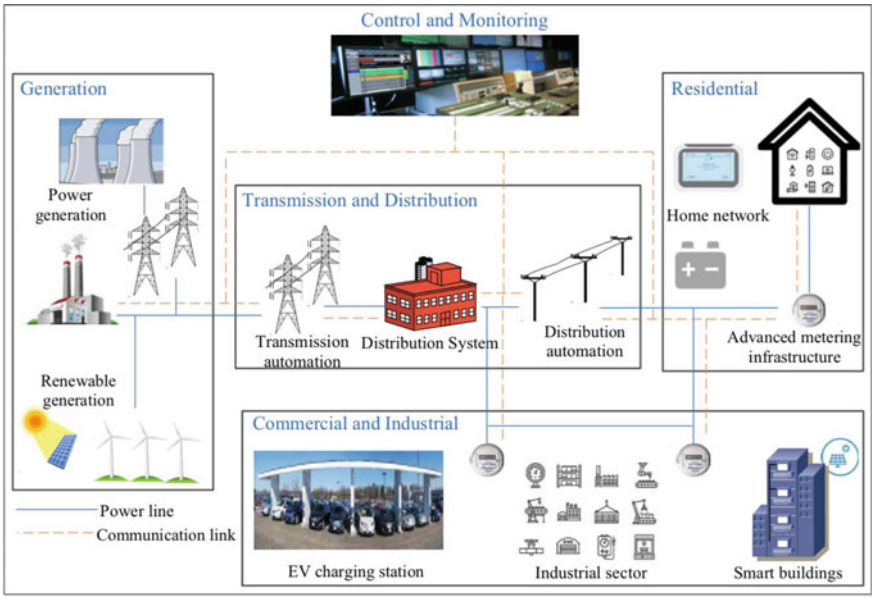


Fig. 11.2 Energy and data flow in smart grid

Table 11.1 Data communication protocols

Type of communication	Technology	Protocol	Frequency range	Data rate
Object-to-object	Dynamic short-range communication	IEEE 802.11p	5.85–5.925 GHz	3–27 Mbps
	Dynamic spectrum access	IEEE 802.11af	476–494 MHz	1 Mbps
Object-to-access point	Wireless access in vehicular environment	IEEE 802.11p	5.85–5.925 GHz	3.27 Mbps
Access point-to-server	Wi-Fi	IEEE 802.11 a/b/g	2.4–5 GHz	1–54 Mbps
	WiMAX	IEEE 802.16	1.25–20 MHz	30 Mbps–1Gbps
	Long term evolution (advanced)	–	20–100 MHz	300 Mbps–3 Gbps

which assists in finding the reasoning behind a particular event's occurrence and helps to understand how the system works by determining current issues and ways to improve. The third type, *Predictive analytics*, computes probabilistic predictions to compute future patterns based on the present information set. Lastly, *Prescriptive*

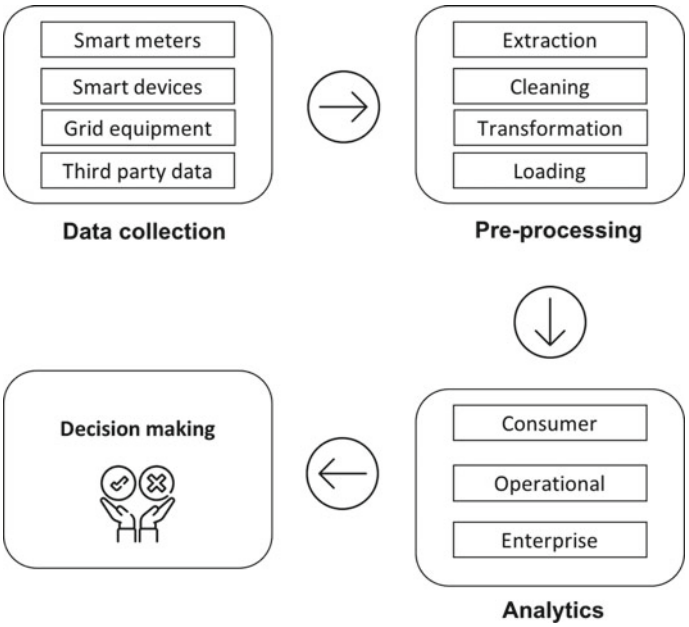


Fig. 11.3 Process of big data analytics in smart grid

analytics helps to identify the outcome of the particular events based on available data and formulate plans for handling such events.

In the field of smart grid, such analytics are applied for managing grid resources and to improve the users' quality of service [3]. The process of performing data analytics in stages, from data collection to decision-making in the smart grid ecosystem, is presented in Fig. 11.3.

In the initial stage, data collection or gathering from various data sources, including smart metres, smart devices, grid equipment, and third-party datasets, is performed. After data collection, the data must be pre-processed to remove any missing or erroneous data. Since the data is collected from multiple sources, it may be in a variety of formats and may contain non-consistent entries. These values are cleaned to cater to the erroneous values in the following way.

- Extraction: Extraction involves extracting the data values from multiple sources in different formats.
- Cleaning: After extraction, the values could be erroneous or no values. These values need to be removed or corrected before further processing, and this process is called data cleaning.
- Transformation: The data from many formats is converted to a target format.
- Loading: Data is loaded into a storage repository where it is stored for further processing.

In smart grids, these techniques relate to consumer, operational, and enterprise. These are generally carried out to process data to gather relevant information. BDA in the smart grid sector can be bifurcated into three generic domains: consumer, operational, and enterprise analytics [4]. *Consumer analytics* can range from demand forecasting to consumer behaviour analysis. *Operational analytics* include equipment maintenance, outage prediction, etc. *Enterprise analytics* varies from providing real-time grid awareness to data visualisation. Based on information gathered after performing one or more types of analytics, informed decisions can be made. Big data analytics has helped to cater to many issues about smart grids, utilities, and consumers in the past such as—load forecasting, data visualisation, and theft detection [5]. To understand more about the role of big data analytics in the smart grid, we will discuss its need in the following section.

11.2.1 Need of Big Data Analytics for Smart Grids

It plays an important role in managing consumers' present and future requirements and giving insights for many smart grid applications.

Big data analytics plays a pivotal role in analysing data in these categories. For instance, loss/theft detection is a good example of big data analytics saving utilities hundreds of millions due to losses from theft and mismanagement of power. Analytics performed on advanced metering infrastructure data can help track this unbilled power, saving millions for the utilities. The demand response analytics helps to optimise the energy consumption for the consumers, thereby reducing the load on the grid resources. The analytics performed on-demand response can forecast the energy demands for the users and help to plan required resources better to cater to the forecasted demands. The predictive analytics on smart grid data would be able to forecast any faults that may occur in the near future, any systems that need to be replaced, and the other outages that may occur in distribution systems. It could also forecast how much power can be accounted for with the adoption of solar PV and electric vehicles in homes and smart grids. The outage detection, asset management, and distribution optimization can all be handled properly with predictive analytics. It also helps create financial models for utilities, considering all these factors for the future that are reliable and meet the customers' requirements. Thus, big data analytics is essential nowadays to gain insights into the factors that influence smart grid decision-making.

11.2.2 Big Data and Cloud Computing

The data can be classified as big data if it has one or many of the following attributes: volume (quantity of data generated is enormous), variety (data types of the collected data are heterogeneous), velocity (speed with which new data is generated and moves around), or veracity (uncertainty in data). All these V's constitute another V, i.e. Value. The big data analytics are performed on the data to gain a value. It means gaining useful information that will help provide new insights about the business.

In terms of smart grid data, the time-series data for load, price, consumption, usage, etc. contribute to the big data.

As cloud infrastructure has an abundance of resources and these can be accessed anywhere and at any time, cloud computing is the reliable choice for big data analytics. Cloud-based big data platforms make accessing massive computing resources for analysing big data practical. Cloud computing offers three types of services, viz., IaaS, PaaS, and SaaS. Cloud services are used to store and process the data as the resources in the cloud are abundant in computation and storage power. Cloud services are also preferred because the cloud can provide uninterrupted service to the end-users anywhere. Cloud services are a viable solution for handling big data for the same reasons.

11.3 Example of Big Data Analytics in Smart Grid

Various studies exist which have applied big data analytics in smart grid environments. However, to give a basic example of big data analytics in the context of a smart grid, we will consider the simple load forecasting scenario in a smart grid. Load forecasting is a very important aspect of smart grid power planning. Calculating the load accurately can help plan the utilities of when and how many generators are required to cater to the load demand. In the considered example, we will take the scenario where the time-series load data for a society consisting of many apartments is predicted using machine learning. The performance of various machine learning models will be compared against different evaluation parameters to determine which model best suits this purpose. The dataset that is considered for this analysis is the UMass Smart dataset [6]. This dataset contains the aggregated energy consumption data for 114 single-family apartments for two years. Moreover, the weather data corresponding to this region is also considered for predicting the load demand of these apartments. We will use different machine learning models on the UMass Smart dataset to predict the load demand. Machine learning can improve the predictions based on available data by determining the most relevant features from the dataset and performing specific actions. The overall process of applying machine learning model(s) is depicted in Fig. 11.4 and discussed as follows.

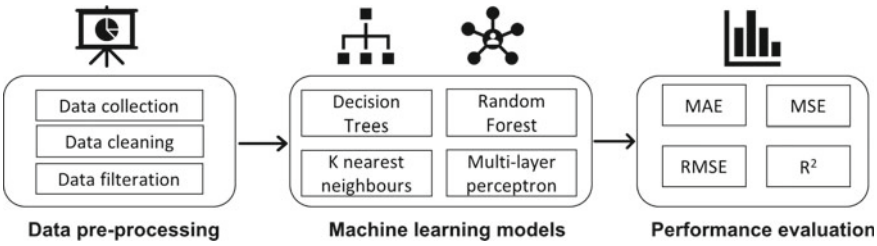


Fig. 11.4 Machine learning process

11.3.1 Data Pre-processing

Data pre-processing is essential for ensuring that data is of high quality and ready for analysis. This helps to remove noise, inconsistencies, and errors from the dataset, making it suitable for machine learning and data analysis.

- *Data Collection*: Initially, data must be collected from the source(s) and converted to a required format.
- *Data Cleaning*: Data cleaning focuses on identifying and rectifying dataset errors, inconsistencies, and anomalies.
- *Data Filtration* : Data filtration aims to extract relevant subsets of the data based on specific criteria. This can be especially important when working with large datasets to focus the analysis on specific aspects.

In the considered case study, the load and climate data are combined to create a unified dataset after removing the missing entries.

11.3.2 Machine Learning Models

The machine learning methods discussed in our analysis are given below.

11.3.2.1 Linear Regression

Linear Regression is a regression model that takes input features and makes predictions for continuous outcomes, such as stock prices or salaries. As its name implies, Linear Regression seeks a linear solution to various problems. Linear Regression uses least squares to fit a line to the data points. The output of the linear regression model can be expressed as:

$$Y_b = b_0 + b_1 \times x_1 + b_2 \times x_2 + ... + b_n \times x_n \tag{11.1}$$

where, b represents the weight parameter assigned to each of the input features ($X = x_1, x_2...x_n$). n is the number of features to predict the output Y. The values of b are chosen randomly initialised, which are then updated to minimise the loss. In the case of linear regression, the loss is defined by the variation of the expected value from that of the predicted. The mean squared error calculates the loss in the linear regression.

11.3.2.2 Decision Trees

The Decision Tree algorithm is a straightforward and comprehensible approach within supervised learning. It operates by constructing a tree-like structure, where each internal node corresponds to a specific feature, each branch represents a possible outcome, and each leaf node signifies a class label in the context of classification or a numerical value in regression tasks. This algorithm can be visualised as a binary tree

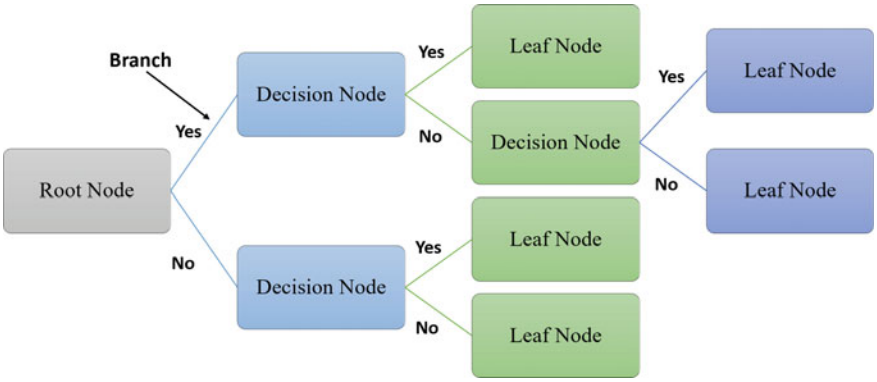


Fig. 11.5 Structure of decision tree

that systematically divides the dataset until only pure leaf nodes remain, indicating instances of a single data class. The essential components of a decision tree are illustrated in Fig. 11.5, showcasing its hierarchical structure and the way it recursively partitions the data.

- Nodes: Nodes represent decision points in the tree.
- Branch: Branch represents possible outcomes of decisions.
- Root Node: The top node that makes the initial decision.
- Leaf Nodes: Terminal nodes that provide the final predictions or classifications.
- Decision Nodes: Nodes other than the root and leaf nodes that make intermediate decisions.

11.3.2.3 Random Forest

Random Forest stands as a powerful supervised learning algorithm that leverages the fundamental principles of decision trees to tackle both regression and classification challenges. It belongs to ensemble learning methods, excelling in its capacity to enhance prediction accuracy while mitigating the risk of overfitting.

The core concept of Random Forest lies in constructing a multitude of decision trees, and it brilliantly synthesises their collective wisdom to make highly accurate predictions. Each tree in this forest is honed on a distinct subset of the original dataset, thanks to a technique known as bootstrapping. Here, bootstrapping involves the random selection of data points with replacement, yielding subsets that are typically smaller in scale compared to the original dataset.

In classification, every decision tree extends its "vote" towards predicting the class to which an instance belongs. The majority vote determines the final verdict, ensuring a robust and reliable classification outcome. Meanwhile, in the context of regression tasks, the individual decision trees work in unison to forecast continuous

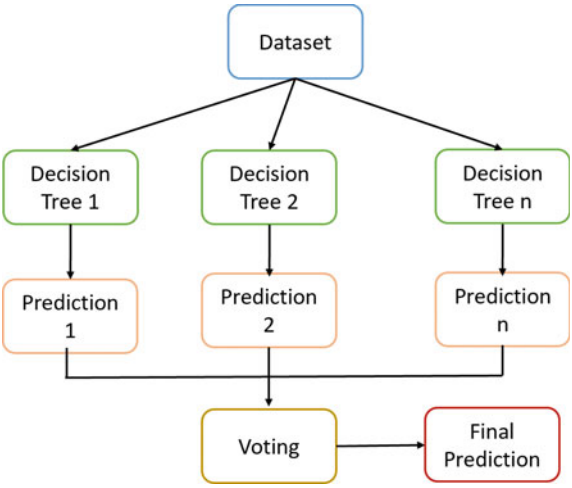


Fig. 11.6 Structure of random forest

values, and the ultimate prediction is drawn from the collective average of these tree-generated predictions. To visualise this collaborative process, refer to the illustrative representation in Fig. 11.6, which serves as a concise yet informative visualisation of the Random Forest algorithm at work.

11.3.2.4 K Nearest Neighbours

KNN is another supervised machine learning algorithm that uses the neighbouring data points to make predictions. In KNN, data is represented in a feature space, and to make predictions, the algorithm identifies the "K" nearest data points to the new input based on one of the distance metrics, such as Euclidean or Manhattan distance. KNN is one of the go-to models for classic classification problems in machine learning and is often used as a baseline model for comparison with other algorithms. It is particularly suitable for datasets with complex or nonlinear decision boundaries.

For classification problems, KNN assigns the most common class among the K nearest neighbours as the prediction class, while for regression problems, KNN computes the average of the target values of neighbouring data points to make a prediction. A simple example to depict its working for the classification task is illustrated in Fig. 11.7. KNN is intuitive, easy to understand, and doesn't make strong assumptions about the data distribution. However, it can be computationally expensive, especially for large datasets, and its performance is influenced by the choice of "k" and the distance metric. Some modifications for selecting the optimal value of "k" for initialisation can help improve the KNN algorithm's performance.

11.3.2.5 Multi-layer Perceptron

The Multi-layer Perceptron (MLP) is a foundational neural network architecture primarily employed in machine learning and deep learning applications. Comprising

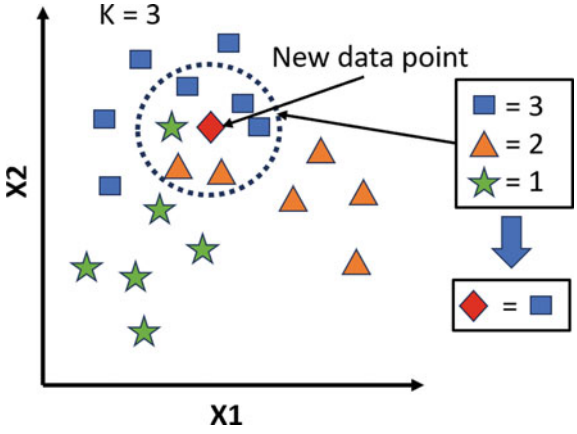


Fig. 11.7 Structure of KNN

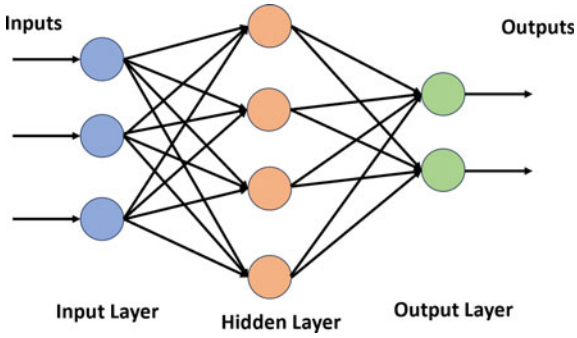


Fig. 11.8 Multi-layer perceptron

numerous layers of interconnected nodes, MLPs are purpose-built for processing intricate data and resolving a wide array of tasks, including classification, regression, and pattern recognition. The network structure of MLP, as illustrated in Fig. 11.8, encompasses an input layer responsible for receiving feature data, one or more hidden layers dedicated to intricate data transformations, and an output layer that delivers the ultimate predictions.

In the intricacies of the MLP, every node, commonly referred to as a neuron, brings its uniqueness by applying an activation function to the weighted sum of its inputs. This seemingly elementary operation introduces the vital element of non-linearity, enabling the model to grasp intricate data patterns.

The training regimen for MLP typically follows the path of supervised learning, a journey where the network acquires the art of precise predictions through fine-tuning connection weights, often employing techniques like backpropagation. Deep MLPs, characterised by their profusion of hidden layers, wield the power to master hierarchical data representations, rendering them indispensable instruments for tasks

such as image recognition and natural language processing. While MLPs proffer immense flexibility and hold the mantle of universal function approximators, their performance profoundly relies on meticulous hyperparameter tuning, the quality of the data they encounter, and the capacity to evade overfitting, particularly in deep architectural configurations.

11.3.3 Results and Evaluations

Different libraries including Pandas, Numpy, Matplotlib, and Sklearn are used in Python to implement the discussed machine learning models. The result of their implementation for predicting values against the actual values in the dataset is shown in Fig. 11.9. It is to be noted that the individual models are tuned for their hyperparameter values to improve their performances.

From this figure, it can be inferred that random forest, kNN, and MLP could be good candidates for predicting the load demand. To find the best model, the performance of the above-specified machine learning models is further evaluated on the following evaluation parameters: mean absolute error (MAE), mean square error (MSE), root mean square error (RMSE), and R-squared, which are defined as below.

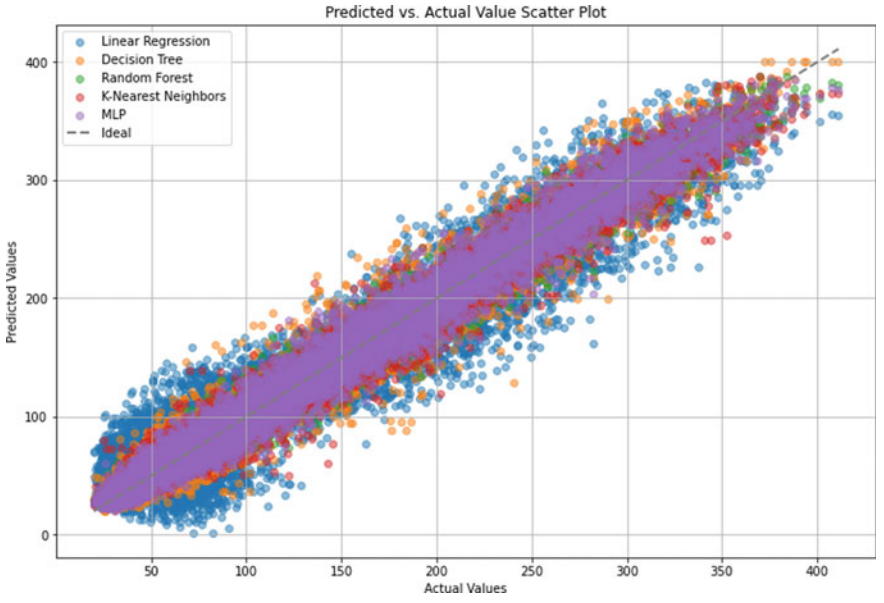


Fig. 11.9 Predicted versus actual values for various models

1.
$$\text{MAE} = \sum_{i=1}^{N} |x_i - y_i| \tag{11.2}$$

2.
$$\text{MSE} = \sum_{i=1}^{N} (x_i - y_i)^2 \tag{11.3}$$

3.
$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N} (x_i - y_i)^2}{N}} \tag{11.4}$$

4.
$$\text{R}^2 = 1 - \frac{\sum_{i=1}^{N} (x_i - y_i)^2}{\sum_{i=1}^{N} (x_i - \bar{x})^2} \tag{11.5}$$

where, x_i depicts the ith value of the actual data, y_i depicts the ith value of the predicted data, $\bar{x}$ is the mean of data values, and N represents the total number of values in the dataset.

The values of MAE and MSE for all five machine learning models are depicted in Figs. 11.10 and 11.11 respectively. It can be seen from these figures that Random Forest outperforms all other models. However, the error values for kNN and MLP are also not far off.

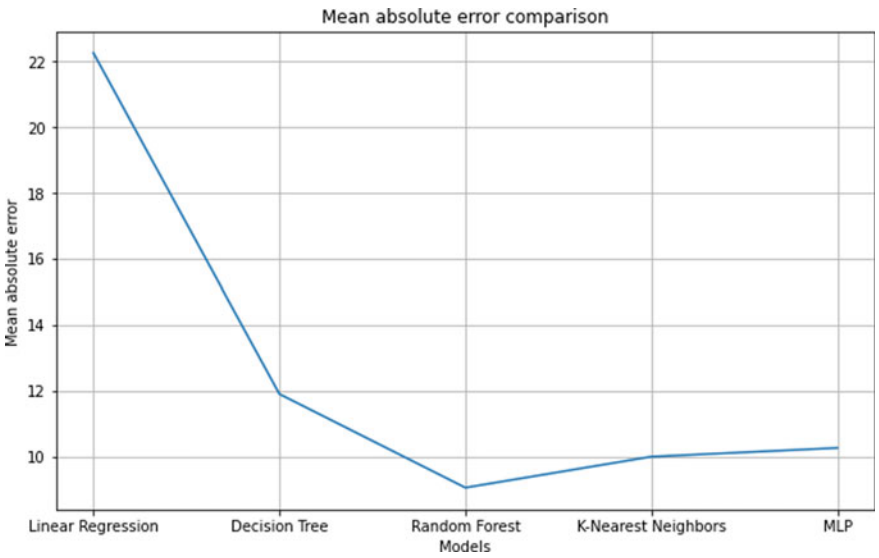


Fig. 11.10 MAE for different models

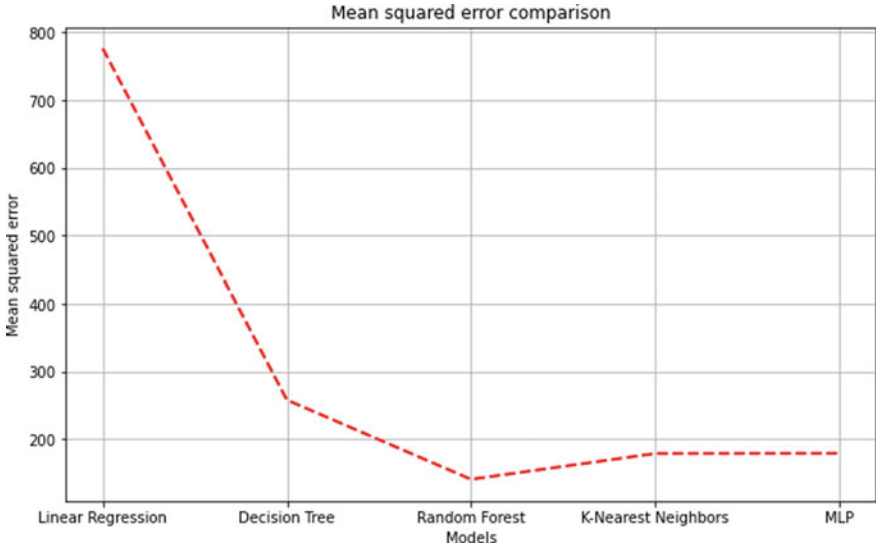


Fig. 11.11 MSE for different models

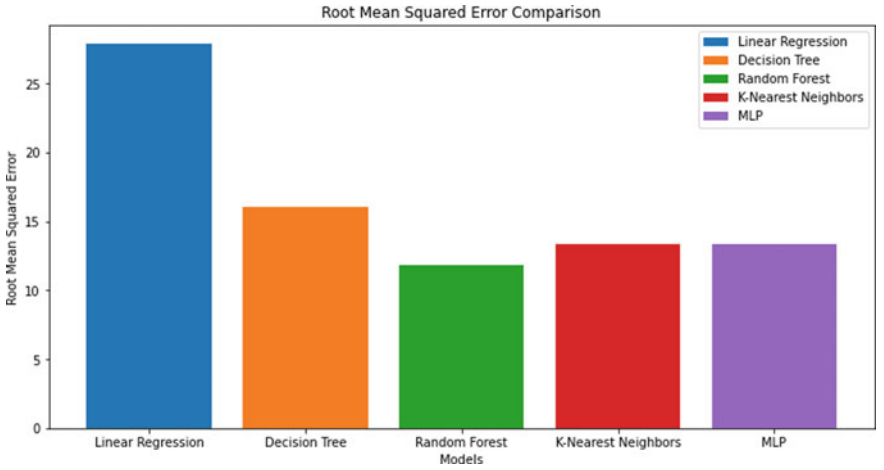


Fig. 11.12 RMSE for different models

Regarding the RMSE, the performance of the Random Forest model is the best out of all the five considered models; however, kNN and MLP also have comparable performances. The comparison of RMSE values for different machine learning models is depicted in Fig. 11.12.

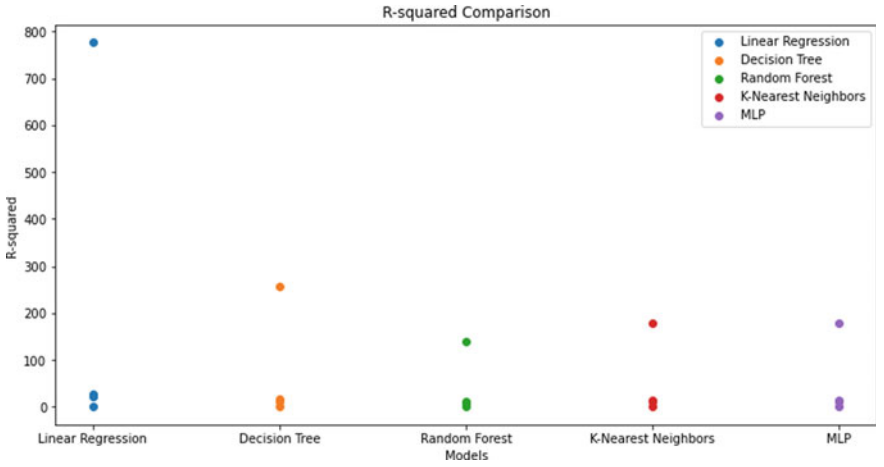


Fig. 11.13 R^2 for different models

The value for the coefficient of determination, i.e. R-squared gives insights about the goodness of fit for the model. For the considered five models, Fig. 11.13 illustrates the value of R^2 for each model. As with other performance matrics, Random Forest again outperforms all other models.

11.4 Learning Outcomes of the Chapter

- **Smart Grid big data analytics**: Exploring the realm of smart grids, their role in modern infrastructure, and the pivotal application of big data analytics in optimising smart grid operations. Investigating the implementation of big data analytics in the context of smart grids, uncovering its implications for data-driven decision-making and system enhancements.
- **Big Data and Cloud Computing**: Examining the synergy between big data and cloud computing within the smart grid landscape, understanding how these technologies collaborate to address challenges and drive innovation.
- **Example of big data analytics in Smart Grid**: Illustrating real-world scenarios showcasing the effective use of big data analytics in improving smart grid functionalities, focussing on tangible outcomes and benefits.

References

1. U.S. Energy Information Administration, Annual energy outlook, table a8. electricity supply, disposition, prices, and emissions, reference case: 2015 (2016)
2. A. Jindal, *Data Analytics of Smart Grid Environment for Efficient Management of Demand Response* (2018)
3. A. Jindal, N. Kumar, M. Singh, A unified framework for big data acquisition, storage, and analytics for demand response management in smart cities. Fut. Gen. Comput. Syst. **108**, 921–934 (2020). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X17324780
4. C.L. Stimmel, *Big Data Analytics Strategies for the Smart Grid*. (CRC Press, 2014)
5. A. Jindal, A. Dua, K. Kaur, M. Singh, N. Kumar, S. Mishra, Decision tree and svm-based data analytics for theft detection in smart grid. IEEE Trans. Ind. Inf. **12**(3), 1005–1016 (2016)
6. UMass Trace Repository, Umass smart dataset (2017). [Online]. Available: https://traces.cs.umass.edu/index.php/smart/smart

Further Reading

7. C.S. Lai, L.L. Lai, Q.H. Lai, *Smart Grids and Big Data Analytics for Smart Cities*. (Springer, 2021)
8. R. Viral, D. Asija, S. Salkuti, *Big Data Analytics Framework for Smart Grids*. (CRC Press, 2023)

Big Data Analytics in Bioinformatics

12

The future of medicine is in biomedical research and bioinformatics.

—Leroy Hood

This chapter introduces the intricate interplay between big data analytics and bioinformatics, providing a comprehensive perspective on leveraging large-scale genomic data. Delving into the challenges posed by big data in bioinformatics, the narrative unfolds to explore frameworks tailored for managing extensive genomic datasets and the pivotal role of biological databases. The core focus is applying big data analytics in bioinformatics, spanning the employment of Hadoop, MapReduce, and deep learning methodologies. A detailed case study exemplifies the practical implementation of variant detection in genomes, illustrating processes like data copying to HDFS, MapReduce-based data processing, and the multistep intricacies of variant calling and interpretation. This chapter serves as a roadmap by navigating the synergy between cutting-edge analytics and the intricate nuances of bioinformatics.

12.1 Big Data: Bioinformatic Perspective

Bioinformatics, or computational biology, is a multidisciplinary science mainly using computers and statistics to collect, store, analyse, and share biological data and information (Fig. 12.1). With the advancement of technology and techniques in biomedical research, there has been exponential growth in the amount of data produced in omic disciplines such as genomics, proteomics, metabolomics, and pharmacogenomics. In genomics, for example, the drastic decrease in the cost of next-generation sequencing technologies leads to an unprecedented proliferation of genomic data. Applying

Ü. Demirbaga et al., *Big Data Analytics*,
https://doi.org/10.1007/978-3-031-55639-5_12

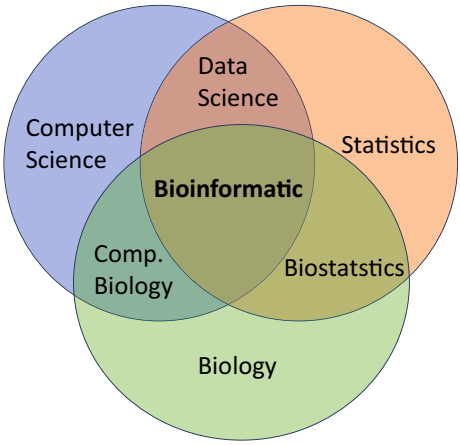


Fig. 12.1 Bioinformatics as a multidisciplinary field

genomic studies to medicine and medical research is key to advancing precision medicine. However, handling big data is becoming more and more difficult due to this exponential growth and requires cloud computing and big data technologies. In this chapter, big data analytics in bioinformatics are elaborated, and an example of workflow using Hadoop MapReduce, a common analysis in medical genetics, is shown for variant detection in the genome.

A notable trend in modern biomedical research is the growth of enormous and complex datasets called big data. These expanding databases take on a crucial role when seen through the perspective of bioinformatics, a multidisciplinary subject that combines biology with computational science. Bioinformatics on big data is an all-encompassing strategy that uses sophisticated computational methods, statistical approaches, and algorithmic tools to decipher, analyse, and draw valuable conclusions from voluminous biological data. These datasets include a wide range of biological disciplines, such as genomics, transcriptomics, proteomics, and metabolomics, and they have a variety of experimental methods, including mass spectrometry, high-throughput sequencing, and large-scale screening assays [1].

The bioinformatics viewpoint on big data is essential for data-driven research in the biological sciences within this framework. It entails the development of complicated computational tools and analytical procedures designed specifically to address the particular difficulties brought on by the size, complexity, and variety of biological data. To permit thorough and complete studies, it also necessitates the integration of multiple data sources, ranging from publicly accessible databases to confidential experimental results. This method makes it possible to grasp complex biological processes and promises to revolutionise areas such as personalised treatment, drug development, and our fundamental knowledge of living systems [2,3]. The fusion of big data and bioinformatics is emerging as a vital confluence, catalysing revolutionary advances in our quest to understand the complexities of life as the frontiers of biological knowledge continue to be pushed.

12.1.1 Big Data Problems in Bioinformatics

In the contemporary era of life sciences, handling and interpreting large-scale biological data is of utmost relevance. This enormous increase in data production from multiple omics and high-throughput sequencing technologies has created formidable obstacles and amazing potential. Genomic, transcriptomic, proteomic, and metabolomic datasets now encompass petabytes of information (Fig. 12.2). Data processing, storage, and analysis are all impacted by this exponential growth. Additionally, it is difficult to integrate and harmonise these disparate sources due to the complexity of biological data types, which range from DNA sequences to protein-protein interactions. Furthermore, the data velocity, which is the rate at which data is produced in bioinformatics, is astonishing [4]. Developing effective data processing pipelines is required because high-throughput sequencing devices generate data at speeds that can strain conventional computational capabilities. Lastly, the inherent complexity of biological systems, from molecular interactions to ecosystems, requires advanced algorithms and computational techniques to derive useful insights. However, these difficulties also present a huge opportunity. Big data in bioinformatics has the potential to provide a deeper understanding of the underpinnings of life. Customising treatments to a person's genetic make up and disease profile offers the possibility of precision medicine [5]. Big data analytics can also speed up discovering new drugs by locating prospective drug targets, forecasting drug interactions, and improving medication design [6].

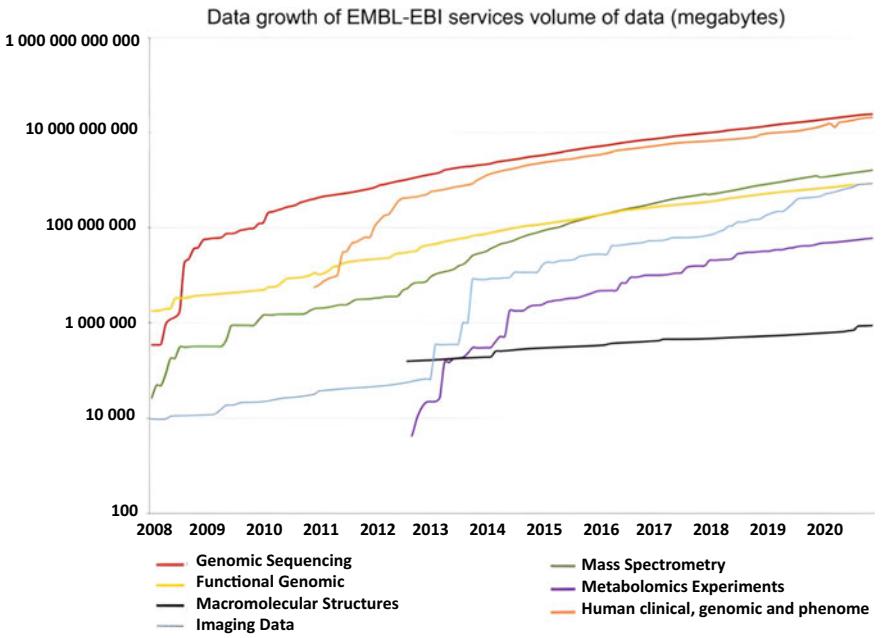


Fig. 12.2 Data growth of EMBL-EBI services by data type [7]

The creation of biotechnological solutions such as genetically modified crops, biopharmaceuticals, and biofuels also heavily relies on bioinformatics. It enables the study of cellular heterogeneity using single-cell sequencing, investigation of epigenetic alterations in epigenomics, and exploration of the genetic variety of microbial communities in metagenomics. Genomic research and the production of bioinformatics data have been transformed by high-throughput sequencing techniques, often known as next-generation sequencing (NGS). These technologies make it possible to sequence DNA and Ribonucleic acid (RNA) quickly and affordably. Whole genome sequencing, transcriptomics, metagenomics, epigenomics, and single-cell sequencing have all been made possible by NGS. These developments have ushered in a brand-new era of information gathering, enabling scientists to investigate genetic variances, gene expression patterns, microbial diversity, epigenetic changes, and cellular heterogeneity at unprecedented depths and breadths (Fig. 12.3) [8]. Data acquisition in bioinformatics spans multiple omics technologies:

- Genomics: Genomic data acquisition involves sequencing an organism's genome, offering comprehensive insights into genes, non-coding regions, and structural variations.
- Transcriptomics: Focussed on RNA molecules, RNA sequencing (RNA-Seq) quantifies mRNA levels, detects alternative splicing events, and identifies non-coding RNAs, shedding light on gene regulation and cellular responses.
- Proteomics: Analysing the complete set of proteins expressed in a cell or organism, proteomics leverages mass spectrometry-based techniques to identify and quantify proteins, providing insights into their functions and interactions.
- Metabolomics: By studying small molecule metabolites, metabolomics elucidates metabolic pathways, biochemical reactions, and metabolic phenotypes. This is valuable in disease diagnosis and understanding metabolic dysregulation.

Given the enormous number of data generated daily in bioinformatics, effective data management and storage are essential. To handle this data flood, researchers must consider scalable storage options, including cloud-based and high-performance computing clusters. Data integration pipelines are crucial to combine disparate datasets and enable thorough cross-omics analysis and biological insights. Security and privacy concerns are crucial when working with private and sensitive clinical and human genetic data. Promoting data sharing and open-access programmes stimulates collaboration and speeds up scientific discoveries. Proper metadata annotation is vital for simplifying data retrieval and understanding (Fig. 12.3).

In conclusion, the bioinformatics big data age represents a crucial turning point for the life sciences. Unlocking biological mysteries, personalising medical care, accelerating drug discovery, and revolutionising biotechnology are all possible thanks to it. To fully utilise large-scale biological datasets and spur innovation in genomics, medicine, and biotechnology, it is necessary to address the issues of data volume, diversity, velocity, and complexity while assuring reliable data storage and management.

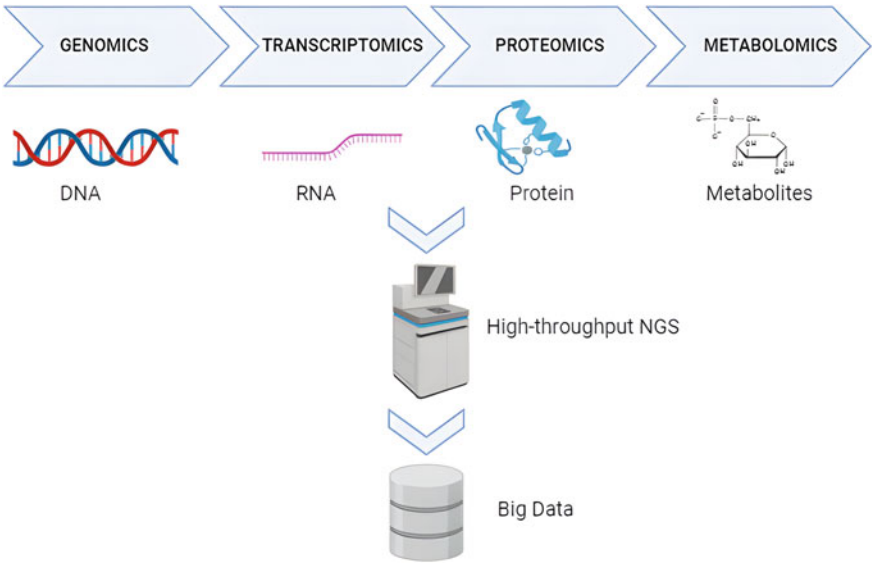


Fig. 12.3 Generating Big Data by high-throughput NGS techniques

12.2 Frameworks for Big Genome Data

Big genome data handling and analysis require specific frameworks and tools to quickly process, store, and retrieve valuable insights from enormous amounts of genetic data. The following are some prevalent platforms and frameworks for handling and analysing massive genomic data:

- Hadoop: Large datasets can be processed and stored in a distributed manner using Hadoop, an open-source platform. It is appropriate for activities like variant calling, alignment, and sequence assembly since it can be used to parallelize genome data analyses [9].
- Apache Spark: Another distributed computing platform that provides high-level APIs for big data processing is Apache Spark. It has in-memory data processing capabilities, which can greatly speed up activities involving the study of genetic data [10].
- Galaxy: Galaxy is an open-source platform created primarily for studying genomics-related data in the biomedical field. It provides a user-friendly interface and various tools and workflows for analysing genetic data. Additionally, Galaxy encourages research reproducibility [11].
- Bioconductor: Bioconductor is an R-based platform emphasising bioinformatics and genomics. For the processing and visualisation of high-throughput genomics

data, such as that from microarrays and next-generation sequencing, it offers a wide range of packages and tools [12].
- Genome Analysis Toolkit (GATK): GATK, a powerful software suite for variant detection in high-throughput sequencing data, was created by the Broad Institute. It is frequently employed for activities like genotyping and variation calling [13].
- Ensembl: Ensembl is a genome annotation project offering many genome datasets and tools. It is very helpful for researching genetic diversity and annotating genomes [14].
- Docker and Singularity: Containerization tools like Docker and Singularity are useful for building portable and reproducible environments for analysing genetic data. To simplify sharing and duplicate findings, researchers might package the tools and workflows they use for analysis into containers [15].
- UCSC Genome Browser: The Genome Browser from the University of California, Santa Cruz (UCSC) is a popular tool for examining and displaying genomic data. It provides many genomes and annotations [16].
- DECPHER: A platform called DECPHER combines clinical, genetic, and therapeutic data to promote precision medicine in oncology and haematology. It enables clinicians and researchers to examine patient-specific data for better cancer detection and care [17].

These platforms and frameworks support various aspects of big genomic data research, including variant calling, annotation, and visualisation. Researchers frequently select a tool and platform combination to effectively meet their unique research goals and computing resources.

12.3 Biological Databases

The backbone of bioinformatics and biological databases is essential to big data analytics in the life sciences. They provide crucial benefits by acting as central archives for enormous amounts of biological data from diverse omics technologies. They first facilitate effective data retrieval, giving researchers quick access to a wealth of biological data and facilitating study by removing the need to generate or curate data repeatedly. Second, these databases enable cross-referencing and data integration, allowing for sophisticated analysis and investigation of complicated biological interactions. Additionally, by providing a consistent data storage and annotation platform, they improve scientific discoveries' reproducibility by enabling experiment replication and result validation. Managing enormous datasets produced by high-throughput technologies is made possible by biological databases, which are crucial in the age of big data analytics. They encourage discoveries in genomes, proteomics, and other fields, enabling researchers to unearth occult patterns, locate illness biomarkers, and create novel treatments, ultimately boosting our understanding of biology and enhancing human health. Here's an overview of some commonly used biological databases and their significance:

- **National Centre for Biotechnology Information (NCBI)**
  The NCBI is the biggest resource in biological and genomic research. The NCBI harbours several important databases, including BLAST, a crucial tool for sequence alignment. PubMed is a major resource for reading scientific literature, and GenBank has a variety of nucleotide sequences. Researchers and scientists regularly use the NCBI for various data categories, including structural data, gene annotations, protein sequences, genomic sequences, biomedical literature, etc. The data provided by NCBI is used for various purposes, including genome assembly, functional annotation, homology searches, sequence retrieval, and extensive literature searches. The bulk of biological data and information are accessible and comprehended by scientists through NCBI, a vital resource for the life sciences and genomics [18].
- **Ensembl**
  One of the most well-known genomics databases is Ensembl. It offers various data types such as gene annotations, genomic assemblies, regulatory elements, and variation data containing SNPs and indels. Furthermore, this comprehensive platform facilitates comparative genomics by allowing researchers to explore evolutionary relationships and identify traits conserved among species. The applicability of Ensembl facilitates a wide range of scientific endeavours, such as identifying functional components within genomes and genomic analysis. Moreover, it is essential to variant annotation as it facilitates the interpretation of genetic variations, which is particularly important for figuring out the genetic basis of phenotypic traits and disorders. It is a valuable asset for both fundamental and translational research because of its capacity to provide consistent, high-quality data and tools that allow scientists to investigate and understand this complex realm of genomes across a wide range of species [14].
- **UniProt**
  The Universal Protein Resource, or UniProt, is an essential and extensive database. It is a crucial resource for functional data and protein sequences, including a wide range of well-selected records. Researchers have access to various data types within UniProt, such as comprehensive pathways, functional annotations, post-translational modifications, protein sequences, and even three-dimensional structures. Its many applications include protein identification, which allows researchers to locate and obtain information about a particular protein quickly; functional annotation, which clarifies the functions and characteristics of proteins; pathway analysis, which makes it easier to investigate biological pathways and networks; and structural biology, which offers priceless insights into the three-dimensional structures of proteins [19].
- **GenBank**
  GenBank is essential to the NCBI database, providing genomic DNA, mRNA, and protein-coding sequences. It is a tool that helps researchers deposit and retrieve various genetic data worldwide. GenBank is a comprehensive source of genetic information since it supports many data formats, including nucleotide sequences, whole genomes, plasmids, and Expressed Sequence Tags (ESTs). The platform plays an equally important role in sequence retrieval, providing researchers with

access to a multitude of genetic data that they can use in their research. Beyond this, GenBank facilitates more sophisticated applications like comparative genomics, which looks for patterns in genomes to reveal functional and evolutionary insights, and phylogenetic analysis, which examines the evolutionary relationships between various organisms [20].

- **Protein Data Bank (PDB)**
  One of the most important worldwide databases for 3D structural information on biological macromolecules, including proteins, nucleic acids, and their complexes, is the Protein Data Bank (PDB). The fields of structural biology and drug design benefit immensely from this resource, which is an actual mine of data essential to understanding the complex structures of these biomolecules. Researchers can explore molecular structures using PDB's enormous dataset, which includes valuable crystallographic data, precise ligand information, and 3D protein and nucleic acid structures. Its diversified applications include protein-ligand interactions, drug discovery, understanding the functions and interactions of biomolecules, structural biology, molecular interactions at the atomic level, and molecular modelling, essential for developing predictive models and simulations to comprehend intricate biological processes. [21].
- **Kyoto Encyclopedia of Genes and Genomes (KEGG)**
  A key tool in bioinformatics, the KEGG is well known for its importance in clarifying the complex mechanisms of biological systems and how they relate to diseases. It provides an extensive database of pathway data, including metabolic pathways, pathway maps, functional annotations, and disease-related pathways for various species. Because of KEGG's numerous applications, including pathway analysis, functional enrichment studies, and system biology study facilitation, scientists and researchers rely on it extensively. Because of KEGG's abundance of information, scientists have a greater ability to understand the molecular underpinnings of health and illness, identify the molecular components participating in different pathways, and discover the interwoven linkages among biological processes [22].
- **STRING**
  A widespread and highly significant database of protein-protein interactions is called STRING. It is a vital resource for researchers in genomics, molecular biology, and bioinformatics since its main function is to compile information on known and predicted protein interactions. STRING sheds light on the complicated relationships between different proteins by allowing scientists to explore the complex world of protein networks and functional linkages. Functional enrichment analyses, predictions of possible connections, and protein-protein interaction networks are just a few of the data forms it provides. Researchers employ STRING for various use cases to better understand the mechanisms behind biological processes and disorders, including network analysis, functional annotation, and protein complex identification. In systems biology and bioinformatics, STRING is crucial in expanding our comprehension of protein interactions and their functions in biological processes [23].

These databases represent a fraction of the vast array of biological resources at the researchers' access. They are necessary for many biological research areas, such as structural biology, functional annotation, proteomics, genomics, and pathway analysis. Access to carefully chosen and structured biological data is crucial to improve biological knowledge and support breakthroughs in disciplines such as genetics, molecular biology, and biomedicine.

12.4 Big Data Analytics in Bioinformatics

The convergence of bioinformatics and big data has created a revolutionary age in the life sciences. The extensive use of Hadoop and the MapReduce programming model in bioinformatics analytics, the evolving landscape of bioinformatics pipelines and workflows created for handling big data, and the crucial role of deep learning and neural networks in addressing important bioinformatics tasks are all covered in this section.

12.4.1 Hadoop and MapReduce in Bioinformatics Analytics

Hadoop, known for its distributed file system and MapReduce programming paradigm, has made considerable strides in bioinformatics analytics. It is used to solve various computational problems in genomics and other fields. Aligning DNA and RNA sequences to reference genomes is one such use [24]. MapReduce seamlessly parallelizes this task, which has the potential to be extremely resource-intensive. The genetic data is dispersed throughout a network of computers in digestible bits. The time needed for alignment is greatly reduced thanks to the separate processing of each fragment by each node. Investigating large-scale biological networks, such as gene co-expression networks or protein-protein interaction (PPI) networks, is another excellent use. Using the MapReduce approach, researchers can use the computational power of distributed computing clusters to analyse complex network features, locate important nodes, and unearth hidden patterns. Scalability, fault tolerance, cost-effectiveness, and parallel processing are advantages. Conversely, algorithm complexity, data serialisation, and performance considerations might be difficult, especially for real time or interactive studies that might not fit Hadoop's batch-oriented approach.

12.4.2 Bioinformatics Pipelines and Workflows for Big Data

The foundation of biological data analysis is the use of bioinformatics pipelines, which coordinate computing operations to reveal crucial biological insights. These pipelines have experienced a transition in the age of big data, adapting to handle the effective and efficient processing of enormous datasets easily. Data pre-processing,

sequence alignment, variant calling, variant annotation, and further downstream analysis are common phases in these pipelines. The complexity of contemporary bioinformatics analyses demands advanced workflow management systems. These systems, including Galaxy [11], Nextflow [25], and Snakemake [26], empower researchers to design, execute, and monitor intricate bioinformatics pipelines. They facilitate the streamlined automation of data-driven processes, ensuring reproducibility and enabling researchers to focus on scientific discovery.

12.4.3 Analysis Pipelines and Tools with Hadoop (MapReduce) Framework

Bioinformatics analysis pipelines have undergone significant adaptations to harness the capabilities of Hadoop and the MapReduce programming model. For instance, variant calling, a pivotal task in genomics, benefits immensely from the parallelism inherent in MapReduce. Tools like Hadoop-BAM have emerged to efficiently handle Binary Alignment Map (BAM)/Sequence Alignment Map (SAM) files, the predominant formats for DNA sequence alignment. These tools harness the distributed computing power of Hadoop clusters to accelerate variant identification and annotation. Moreover, custom MapReduce algorithms have become instrumental in addressing unique bioinformatics challenges. Researchers have developed tailored MapReduce solutions for tasks like DNA sequence clustering or identifying functional elements within genomics data. These bespoke algorithms empower the bioinformatics community to dissect complex biological questions at scale. Additionally, a variety of specialised tools have been developed to enhance further bioinformatics analysis on Hadoop, including CloudBLAST, CloudBurst, Biodoop, Crossbow, Genome Analysis Toolkit (GATK), Myrna, Galaxy, SEAL, CloudAligner, Contrail, FX, BioPig, SeqPig, and Halvade, each offering unique capabilities for handling specific bioinformatics tasks efficiently [27].

12.4.4 Deep Learning in Bioinformatics

Deep learning has become a game-changing force in bioinformatics, improving our capacity to extract knowledge from massive biological datasets. This section examines deep learning's substantial influence on bioinformatics, emphasising its uses, difficulties, and connections to big data analytics [28].

12.4.4.1 Applications of Deep Learning in Bioinformatics

Applications for deep learning can be found in many areas of bioinformatics. For instance, CNNs expertly extract characteristics from DNA and RNA sequences in genomic sequence analysis, where it excels. These networks can locate regulatory areas, forecast binding sites, and group DNA sequences according to their intended use. Deep learning algorithms have also transformed protein structure prediction,

with AlphaFold [29] showing astounding accuracy in inferring 3D protein structures. Deep learning's capacity to forecast gene expression levels, recognise functional components of genomes, and disclose gene regulatory networks is advantageous for functional genomics. Deep learning models, such as RNNs and CNNs, can identify disease subtypes, forecast patient outcomes, and help search for disease biomarkers [28,30].

12.4.4.2 Challenges and Considerations

Several challenges accompany the integration of deep learning into bioinformatics. Data availability remains a critical hurdle, as deep learning's efficacy hinges on large, high-quality labelled datasets, which can be scarce, especially for rare diseases. Interpretability poses another challenge, as complex neural network architectures can obscure the rationale behind their decisions. Computational resources, such as high-performance computing clusters with GPUs or TPUs, are essential for training large-scale deep learning models, potentially limiting accessibility [30,31].

12.4.4.3 Leveraging Big Data Analytics in Deep Learning

Combining deep learning and big data analytics increases its influence on bioinformatics. The enormous volume and complexity of biological data are perfectly suited to autonomous feature extraction and deep learning pattern identification capabilities. Deep learning methods simplify feature engineering and data transformation during data pre-processing, reducing the manual work required of researchers. Deep learning frameworks are naturally parallel and scalable, which makes them suited for distributed training and large-scale computing clusters. This ensures effective management of big biological datasets. Deep learning-enabled real-time analytics are also crucial for applications like disease monitoring and diagnosis in high-throughput clinical settings [32].

Our comprehension of biological data has undergone a paradigm shift due to integrating deep learning with bioinformatics. It is necessary to solve issues with data accessibility, interpretability, and computational resources. Deep learning has great potential, especially when combined with big data analytics. This synergy can potentially speed up scientific advancements and advance precision medicine.

12.5 Variant Detection in Genome: A Case Study

12.5.1 Genom Data Copying to HDFS

Whole exome sequencing data of adult female colon cancer patients (SRA Accession Number: **SRR25243226**) was retrieved from the NCBI in Fastq format using the SRA toolkit. This dataset was transferred from a local machine to the HDFS by leveraging

a combination of Hadoop's command-line utilities. The executed command line is shown below:

```
$ hadoop fs –copyFromLocal /home/hduser/Desktop/genomData.txt /user
/hduser/genom/
```

12.5.2 Big Genome Data Processing Using MapReduce

In this case study, variant detection—a common analysis in genomic medicine—is shown by using big data analytics approaches. The process begins with cleaning and quality control, followed by alignment, marking duplicates, coverage analysis, variant calling, and annotation (Fig. 12.4). These procedures are essential for identifying

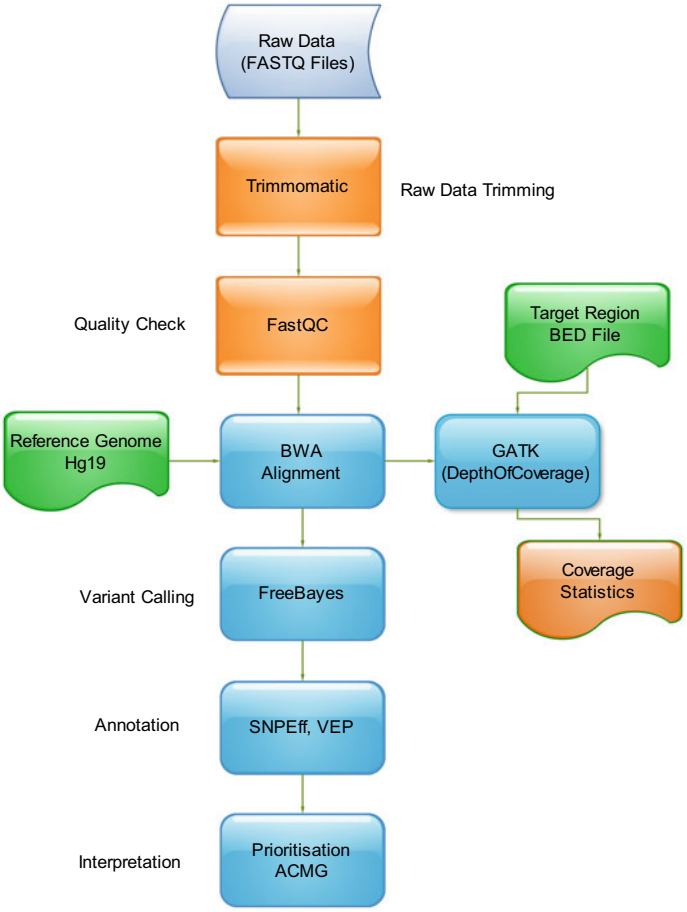


Fig. 12.4 Outline of the pipeline for case study

genetic variations, which form the basis for many genomics and genetics research fields.

12.5.2.1 Data Cleaning and Quality Control

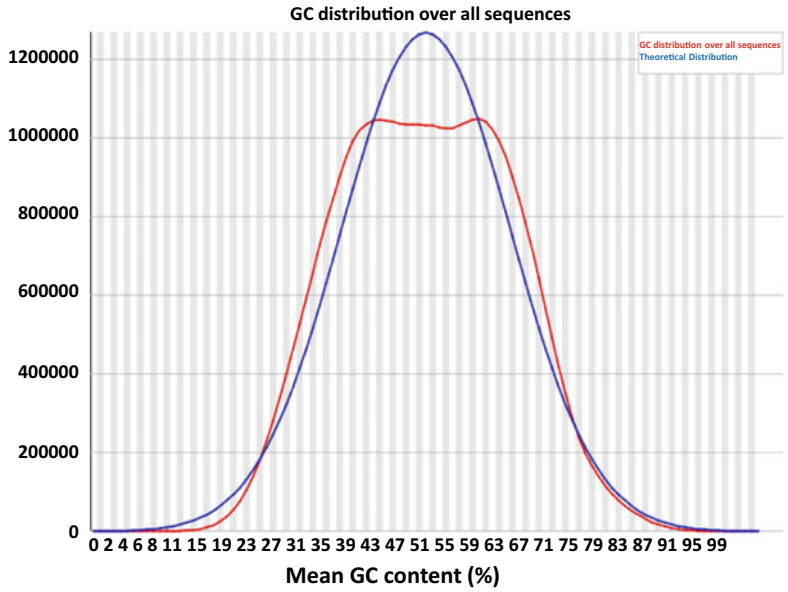
Ensuring the genetic data is accurate, clean, and of the highest quality is the main goal of the first stage of genome analysis. Eliminating low-quality sequences and any artefacts induced during sequencing is the main aim. Quality control reports for the input FASTQ files are generated using FastQC to do this. These statistical and graphical reports evaluate many aspects of the data quality, including sequence quality, sequence length distribution, and overrepresented sequences. These reports play a crucial role in locating any problems in the data that may need further examination. In our case, it can be clearly seen from the guanine-cytosine (GC) and quality score distribution plots that the data has enough quality for further analysis. A bell-shaped curve in a GC distribution without deviations from the theoretical distribution (blue line) indicates no contamination or biases in sequencing. It displays the distribution of quality scores assigned to each sequence base. Quality scores reflect the accuracy of base calls, with higher scores indicating better confidence (Fig. 12.5). The cleaning stage uses Trimmomatic, an application widely used for trimming and cleaning FASTQ files. Trimmomatic can eliminate low-quality bases, adapter sequences, and sequences that don't meet a predetermined quality level. This is required because contaminated or low-quality sequences can produce incorrect variant calls and reduce the accuracy of downstream analysis.

12.5.2.2 Alignment

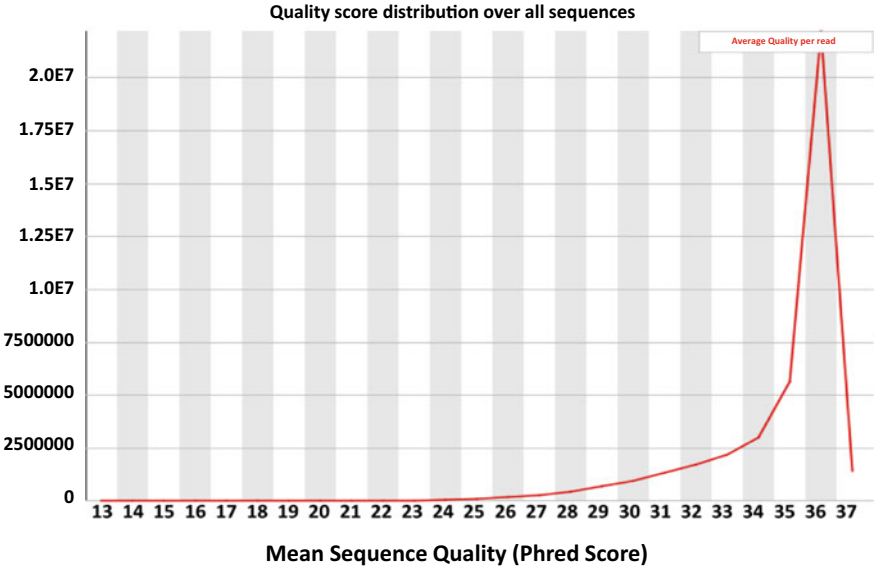
After quality control and data cleaning, the clean reads are aligned to a reference genome using BWA-MEM, an extremely precise and efficient tool that uses the Memory Efficient Mapping (MEM) method and the Burrows-Wheeler Transform (BWT) to produce high-quality alignments. After alignment, the aligned reads are sorted using Samtools. This is a crucial step that ensures the reads are arranged correctly in the BAM file, improving the accuracy and efficiency of downstream analysis. It's important to remember that this stage frequently calls for a substantial amount of computational power. For this reason, MapReduce, a distributed computing paradigm capable of processing massive amounts of data efficiently, is frequently used (Fig. 12.6).

12.5.2.3 Coverage Statistics

At this step, we use the GATK DepthOfCoverage programme to compute coverage statistics on specific genomic areas. The objective is to determine how well the genome was sequenced in targeted locations, with a focus on finding regions with low and high coverage. Finding these places is important for assessing the data quality since low coverage might lead to missed variations, and high-coverage regions are more likely to have artefacts.



(a) GC distribution plot



(b) The quality scores per sequence plot

Fig. 12.5 Figures for data quality check

Fig. 12.6 Alignment of sequence reads to a reference genome

12.5.2.4 Marking Duplicates and Quality Control of BAM file

Duplicate readings from sources such as Polymerase Chain Reaction (PCR) amplification can distort the findings of variant calling; thus, it's critical to identify and manage them to avoid undue influence. Quality control measures are also generated to evaluate the alignment's quality. To help remove redundant data and ensure that each unique read is only considered once, the programme Picard MarkDuplicates marks duplicate reads in the BAM file.

12.5.2.5 Variant Calling

At this stage, FreeBayes is used to identify genomic variants, such as insertions, deletions, and single nucleotide variants (SNVs). FreeBayes is a tool that accurately detects variants within the aligned reads using a Bayesian statistical framework. This tool is well known for its sensitivity in identifying even low-frequency variants. In our case, around 113 thousand single nucleotide variants are detected.

12.5.2.6 Variant Annotation and Interpretation

Once variants are called, it is essential to add functional and structural information to them. Tools for annotating variations in the Variant Call Format (VCF) file that are often used are SNPEff and Variant Effect Predictor (VEP). SNPEff helps researchers understand the functional implications of SNVs by providing data on how they affect genes, transcripts, and proteins. In the same way, VEP provides thorough annotations for variants that include information on how variants affect transcripts, known variations in databases, and protein-coding areas. These annotation tools are essential to further understanding the variations' functional consequences. Then, these huge numbers of variants need to be prioritised and interpreted, relying on the annotations. Variant interpretation is based on established rules, such as those issued by the American College of Medical Genetics and Genomics (ACMG) [33]. In particular, these recommendations provide a framework for assessing the clinical importance of genetic variations for identifying genetic diseases. The procedure entails evaluating several variables, such as the variant's allele frequency in the population, its possible effect on gene function, and contradictory or supportive data

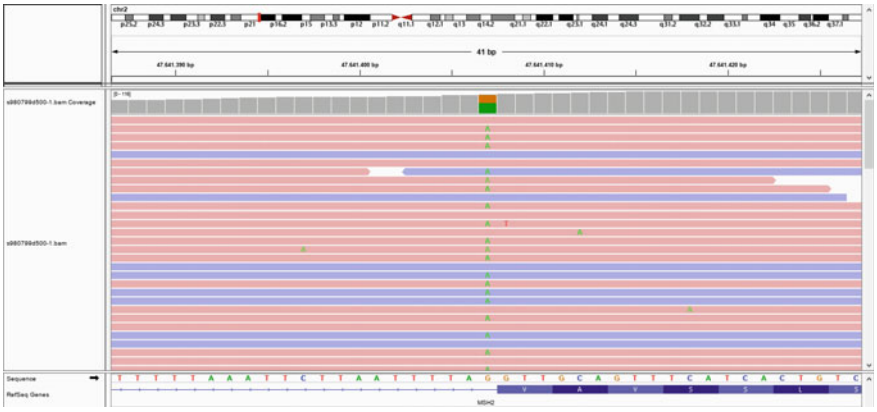


Fig. 12.7 IGV plot of the identified causative SNV in the patient with colon cancer

from several sources. To identify whether a variation is pathogenic, likely pathogenic, unknown significance, likely benign, or benign, variant interpretation incorporates genetic and clinical data. In our case, by prioritising variants based on established rules and patient phenotype, we identify one causative SNV for colon cancer among the 113,000 variants. The SNV occurs in one of the intron-exon boundaries of *MSH2* gene, which disrupts normal splicing (Fig. 12.7). The variant associated with Lynch syndrome is consistent with the patient phenotype [34].

12.6 Learning Outcomes of the Chapter

- **Big Data Analytics in Bioinformatics**: Exploring the challenges and perspectives at the intersection of big data and bioinformatics.
- **Big Data: Bioinformatics Perspective**: Examine the unique challenges of handling big data in bioinformatics.
- **Frameworks for Big Genomic Data**: Investigating key frameworks designed to manage large genomic datasets efficiently.
- **Biological Databases**: Explore the role of biological databases in storing and retrieving complex biological information.
- **Big Data Analytics in Bioinformatics**: Analysing how big data analytics contributes to extracting meaningful insights from vast biological datasets.
- **Case Study in Genomic Medicine**: Exploring a case study in genomic medicine, focussing on applying big data analytics in variant detection within the genome.

References

1. X. Dai, L. Shen, Advances and trends in omics technology development. Front. Med. **9** (2022). [Online]. Available: https://doi.org/10.3389/fmed.2022.911861
2. H. Askr, E. Elgeldawi, H.A. Ella, Y.A. M.M. Elshaier, M.M. Gomaa, A.E. Hassanien, Deep learning in drug discovery: an integrative review and future challenges. Artif. Intell. Rev. **56**(7), 5975–6037 (2022). [Online]. Available: https://doi.org/10.1007/s10462-022-10306-1
3. K. Batko, A. Ślęzak, The use of big data analytics in healthcare. J. Big Data **9**(1) (2022). [Online]. Available: https://doi.org/10.1186/s40537-021-00553-4
4. S. Pal, S. Mondal, G. Das, S. Khatua, Z. Ghosh, Big data in biology: The hope and present-day challenges in it. Gene Rep. **21**, 100869 (2020). [Online]. Available: https://doi.org/10.1016/j.genrep.2020.100869
5. M. Hassan, F.M. Awan, A. Naz, E.J. deAndrés Galiana, O. Alvarez, A. Cernea, L. Fernández-Brillet, J.L. Fernández-Martínez, A. Kloczkowski, Innovations in genomics and big data analytics for personalized medicine and health care: A review. Int. J. Mol. Sci. **23**(9), 4645 (2022). [Online]. Available: https://doi.org/10.3390/ijms23094645
6. B. Chen, A. Butte, Leveraging big data to transform target selection and drug discovery. Clin. Pharmacol. Therapeut. **99**(3), 285–297 (2016). [Online]. Available: https://doi.org/10.1002/cpt.318
7. G. Cantelli, A. Bateman, C. Brooksbank, A.I. Petrov, R.S. Malik-Sheriff, M. Ide-Smith, H. Hermjakob, P. Flicek, R. Apweiler, E. Birney, J. McEntyre, The european bioinformatics institute (EMBL-EBI) in 2021. Nucleic Acids Res. **50**(D1), D11–D19 (2021). [Online]. Available: https://doi.org/10.1093/nar/gkab1127
8. H. Satam, K. Joshi, U. Mangrolia, S. Waghoo, G. Zaidi, S. Rawool, R.P. Thakare, S. Banday, A.K. Mishra, G. Das, S.K. Malonia, Next-generation sequencing technology: Current trends and advancements. Biology **12**(7), 997 (2023). [Online]. Available: https://doi.org/10.3390/biology12070997
9. Apache Software Foundation. Hadoop [Online]. Available: https://hadoop.apache.org
10. M. Zaharia, R.S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, X. Meng, J. Rosen, S. Venkataraman, M.J. Franklin, A. Ghodsi, J. Gonzalez, S. Shenker, I. Stoica, Apache spark. Commun. ACM **59**(11), 56–65 (2016). [Online]. Available: https://doi.org/10.1145/2934664
11. E. Afgan, A. Nekrutenko, B.A. Grüning, D. Blankenberg, J. Goecks, M.C. Schatz, A.E. Ostrovsky, A. Mahmoud, A.J. Lonie, A. Syme, A. Fouilloux, A. Bretaudeau, A. Nekrutenko, A. Kumar, A.C. Eschenlauer, A.D. DeSanto, A. Guerler, B. Serrano-Solano, B. Batut, B.A. Grüning, B.W. Langhorst, B. Carr, B.A. Raubenolt, C.J. Hyde, C.J. Bromhead, C.B. Barnett, C. Royaux, C. Gallardo, D. Blankenberg, D.J. Fornika, D. Baker, D. Bouvier, D. Clements, D.A. de Lima Morais, D.L. Tabernero, D. Lariviere, E. Nasr, E. Afgan, F. Zambelli, F. Heyl, F. Psomopoulos, F. Coppens, G.R. Price, G. Cuccuru, G.L. Corguillé, G.V. Kuster, G.G. Akbulut, H. Rasche, H.-R. Hotz, I. Eguinoa, I. Makunin, I.J. Ranawaka, J.P. Taylor, J. Joshi, J. Hillman-Jackson, J. Goecks, J.M. Chilton, K. Kamali, K. Suderman, K. Poterlowicz, L.B. Yvan, L. Lopez-Delisle, L. Sargent, M.E. Bassetti, M.A. Tangaro, M. van den Beek, M. Čech, M. Bernt, M. Fahrner, M. Tekman, M.C. Föll, M.C. Schatz, M.R. Crusoe, M. Roncoroni, N. Kucher, N. Coraor, N. Stoler, N. Rhodes, N. Soranzo, N. Pinter, N.A. Goonasekera, P.A. Moreno, P. Videm, P. Melanie, P. Mandreoli, P.D. Jagtap, Q. Gu, R.J.M. Weber, R. Lazarus, R.H.P. Vorderman, S. Hiltemann, S. Golitsynskiy, S. Garg, S.A. Bray, S.L. Gladman, S. Leo, S.P. Mehta, T.J. Griffin, V. Jalili, V. Yves, V. Wen, V.K. Nagampalli, W.A. Bacon, W. de Koning, W. Maier, P.J. Briggs, The galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2022 update. Nucleic Acids Res. **50**(W1), W345–W351 (2022). [Online]. Available: https://doi.org/10.1093/nar/gkac247
12. R. Ihaka, R. Gentleman, R: A language for data analysis and graphics. J. Comput. Graph. Stat. **5**(3), 299–314 (1996). [Online]. Available: https://doi.org/10.1080/10618600.1996.10474713

13. A. McKenna, M. Hanna, E. Banks, A. Sivachenko, K. Cibulskis, A. Kernytsky, K. Garimella, D. Altshuler, S. Gabriel, M. Daly, M.A. DePristo, The genome analysis toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data. Genome Res. **20**(9), 1297–1303 (2010). [Online]. Available: https://doi.org/10.1101/gr.107524.110
14. F.J. Martin, M.R. Amode, A. Aneja, O. Austine-Orimoloye, A.G. Azov, I. Barnes, A. Becker, R. Bennett, A. Berry, J. Bhai, S.K. Bhurji, A. Bignell, S. Boddu, P.R.B. Lins, L. Brooks, S.B. Ramaraju, M. Charkhchi, A. Cockburn, L.D.R. Fiorretto, C. Davidson, K. Dodiya, S. Donaldson, B.E. Houdaigui, T.E. Naboulsi, R. Fatima, C.G. Giron, T. Genez, G.S. Ghattaoraya, J.G. Martinez, C. Guijarro, M. Hardy, Z. Hollis, T. Hourlier, T. Hunt, M. Kay, V. Kaykala, T. Le, D. Lemos, D. Marques-Coelho, J.C. Marugán, G.A. Merino, L.P. Mirabueno, A. Mushtaq, S.N. Hossain, D.N. Ogeh, M.P. Sakthivel, A. Parker, M. Perry, I. Piližota, I. Prosovetskaia, J.G. Pérez-Silva, A.I.A. Salam, N. Saraiva-Agostinho, H. Schuilenburg, D. Sheppard, S. Sinha, B. Sipos, W. Stark, E. Steed, R. Sukumaran, D. Sumathipala, M.-M. Suner, L. Surapaneni, K. Sutinen, M. Szpak, F.F. Tricomi, D. Urbina-Gómez, A. Veidenberg, T.A. Walsh, B. Walts, E. Wass, N. Willhoft, J. Allen, J. Alvarez-Jarreta, M. Chakiachvili, B. Flint, S. Giorgetti, L. Haggerty, G.R. Ilsley, J.E. Loveland, B. Moore, J.M. Mudge, J. Tate, D. Thybert, S.J. Trevanion, A. Winterbottom, A. Frankish, S.E. Hunt, M. Ruffier, F. Cunningham, S. Dyer, R.D. Finn, K.L. Howe, P.W. Harrison, A.D. Yates, P. Flicek, Ensembl 2023. Nucleic Acids Res. **51**(D1), D933–D941 (2022). [Online]. Available: https://doi.org/10.1093/nar/gkac958
15. D. Merkel, Docker: lightweight linux containers for consistent development and deployment. Linux J. **2014**(239), 2 (2014)
16. W.J. Kent, C.W. Sugnet, T.S. Furey, K.M. Roskin, T.H. Pringle, A.M. Zahler, D. Haussler, The human genome browser at UCSC. Genome Res. **12**(6), 996–1006 (2002). [Online]. Available: https://doi.org/10.1101/gr.229102
17. H.V. Firth, S.M. Richards, A.P. Bevan, S. Clayton, M. Corpas, D. Rajan, S.V. Vooren, Y. Moreau, R.M. Pettett, N.P. Carter, DECIPHER: Database of chromosomal imbalance and phenotype in humans using ensembl resources. Am. J. Human Genet. **84**(4), 524–533 (2009). [Online]. Available: https://doi.org/10.1016/j.ajhg.2009.03.010
18. E.W. Sayers, E.E. Bolton, J.R. Brister, K. Canese, J. Chan, D.C. Comeau, R. Connor, K. Funk, C. Kelly, S. Kim, T. Madej, A. Marchler-Bauer, C. Lanczycki, S. Lathrop, Z. Lu, F. Thibaud-Nissen, T. Murphy, L. Phan, Y. Skripchenko, T. Tse, J. Wang, R. Williams, B.W. Trawick, K.D. Pruitt, S.T. Sherry, Database resources of the national center for biotechnology information. Nucleic Acids Res. **50**(D1), D20–D26 (2021). [Online]. Available: https://doi.org/10.1093/nar/gkab1112
19. A. Bateman, M.-J. Martin, S. Orchard, M. Magrane, S. Ahmad, E. Alpi, E.H. Bowler-Barnett, R. Britto, H. Bye-A-Jee, A. Cukura, P. Denny, T. Dogan, T. Ebenezer, J. Fan, P. Garmiri, L.J. da Costa Gonzales, E. Hatton-Ellis, A. Hussein, A. Ignatchenko, G. Insana, R. Ishtiaq, V. Joshi, D. Jyothi, S. Kandasaamy, A. Lock, A. Luciani, M. Lugaric, J. Luo, Y. Lussi, A. MacDougall, F. Madeira, M. Mahmoudy, A. Mishra, K. Moulang, A. Nightingale, S. Pundir, G. Qi, S. Raj, P. Raposo, D.L. Rice, R. Saidi, R. Santos, E. Speretta, J. Stephenson, P. Totoo, E. Turner, N. Tyagi, P. Vasudev, K. Warner, X. Watkins, R. Zaru, H. Zellner, A.J. Bridge, L. Aimo, G. Argoud-Puy, A.H. Auchincloss, K.B. Axelsen, P. Bansal, D. Baratin, T.M.B. Neto, M.-C. Blatter, J.T. Bolleman, E. Boutet, L. Breuza, B.C. Gil, C. Casals-Casas, K.C. Echioukh, E. Coudert, B. Cuche, E. de Castro, A. Estreicher, M.L. Famiglietti, M. Feuermann, E. Gasteiger, P. Gaudet, S. Gehant, V. Gerritsen, A. Gos, N. Gruaz, C. Hulo, N. Hyka-Nouspikel, F. Jungo, A. Kerhornou, P.L. Mercier, D. Lieberherr, P. Masson, A. Morgat, V. Muthukrishnan, S. Paesano, I. Pedruzzi, S. Pilbout, L. Pourcel, S. Poux, M. Pozzato, M. Pruess, N. Redaschi, C. Rivoire, C.J.A. Sigrist, K. Sonesson, S. Sundaram, C.H. Wu, C. N. Arighi, L. Arminski, C. Chen, Y. Chen, H. Huang, K. Laiho, P. McGarvey, D. A. Natale, K. Ross, C.R. Vinayaka, Q. Wang, Y. Wang, J. Zhang, UniProt: the universal protein knowledgebase in 2023. Nucleic Acids Res. **51**(D1), D523–D531 (2022). [Online]. Available: https://doi.org/10.1093/nar/gkac1052

20. K. Clark, I. Karsch-Mizrachi, D.J. Lipman, J. Ostell, E.W. Sayers, GenBank. Nucleic Acids Res. **44**(D1), D67–D72 (2015). [Online]. Available: https://doi.org/10.1093/nar/gkv1276
21. H.M. Berman, The protein data bank. Nucleic Acids Res. **28**(1), 235–242 (2000). [Online]. Available: https://doi.org/10.1093/nar/28.1.235
22. M. Kanehisa, KEGG: Kyoto encyclopedia of genes and genomes. Nucleic Acids Res. **28**(1), 27–30 (2000). [Online]. Available: https://doi.org/10.1093/nar/28.1.27
23. D. Szklarczyk, A. Franceschini, S. Wyder, K. Forslund, D. Heller, J. Huerta-Cepas, M. Simonovic, A. Roth, A. Santos, K.P. Tsafou, M. Kuhn, P. Bork, L.J. Jensen, C. von Mering, STRING v10: protein–protein interaction networks, integrated over the tree of life. Nucleic Acids Res. **43**(D1), D447–D452 (2014). [Online]. Available: https://doi.org/10.1093/nar/gku1003
24. M.C. Schatz, CloudBurst: highly sensitive read mapping with MapReduce. Bioinformatics **25**(11), 1363–1369 (2009). [Online]. Available: https://doi.org/10.1093/bioinformatics/btp236
25. P.D. Tommaso, M. Chatzou, E.W. Floden, P.P. Barja, E. Palumbo, C. Notredame, Nextflow enables reproducible computational workflows. Nat. Biotechnol. **35**(4), 316–319 (2017). [Online]. Available: https://doi.org/10.1038/nbt.3820
26. F. Mölder, K.P. Jablonski, B. Letcher, M.B. Hall, C.H. Tomkins-Tinch, V. Sochat, J. Forster, S. Lee, S.O. Twardziok, A. Kanitz, A. Wilm, M. Holtgrewe, S. Rahmann, S. Nahnsen, J. Köster, Sustainable data analysis with snakemake. F1000Research **10**, 33 (2021). [Online]. Available: https://doi.org/10.12688/f1000research.29032.2
27. L. Shi, Z. Wang, W. Yu, X. Meng, A case study of tuning MapReduce for efficient bioinformatics in the cloud. Parallel Comput. **61**, 83–95 (2017). [Online]. Available: https://doi.org/10.1016/j.parco.2016.10.002
28. S. Min, B. Lee, S. Yoon, Deep learning in bioinformatics. Brief. Bioinform. bbw068 (2016). [Online]. Available: https://doi.org/10.1093/bib/bbw068
29. J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S.A.A. Kohl, A.J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A.W. Senior, K. Kavukcuoglu, P. Kohli, D. Hassabis, Highly accurate protein structure prediction with AlphaFold. Nature **596**(7873), 583–589 (2021). [Online]. Available: https://doi.org/10.1038/s41586-021-03819-2
30. N. Sapoval, A. Aghazadeh, M.G. Nute, D.A. Antunes, A. Balaji, R. Baraniuk, C.J. Barberan, R. Dannenfelser, C. Dun, M. Edrisi, R.A.L. Elworth, B. Kille, A. Kyrillidis, L. Nakhleh, C.R. Wolfe, Z. Yan, V. Yao, T.J. Treangen, Current progress and open challenges for applying deep learning across the biosciences. Nat. Commun. **13**(1) (2022). [Online]. Available: https://doi.org/10.1038/s41467-022-29268-7
31. A. Sharma, R. Kumar, Recent advancement and challenges in deep learning, big data in bioinformatics, in *Studies in Big Data* (Springer International Publishing, 2022), pp. 251–284. [Online]. Available: https://doi.org/10.1007/978-3-030-95419-2_12
32. Y. Kumar, A. Koul, R. Singla, M.F. Ijaz, Artificial intelligence in disease diagnosis: a systematic literature review, synthesizing framework and future research agenda. J. Ambient Intell. Humanized Comput. **14**(7), 8459–8486 (2022). [Online]. Available: https://doi.org/10.1007/s12652-021-03612-z
33. S. Richards, N. Aziz, S. Bale, D. Bick, S. Das, J. Gastier-Foster, W.W. Grody, M. Hegde, E. Lyon, E. Spector, K. Voelkerding, H.L. Rehm, Standards and guidelines for the interpretation of sequence variants: a joint consensus recommendation of the american college of medical genetics and genomics and the association for molecular pathology. Genetics Med. **17**(5), 405–424 (2015). [Online]. Available: https://doi.org/10.1038/gim.2015.30
34. S. Haraldsdottir, H. Hampel, C. Wu, D.Y. Weng, P.G. Shields, W.L. Frankel, X. Pan, A. de la Chapelle, R.M. Goldberg, T. Bekaii-Saab, Patients with colorectal cancer associated with lynch syndrome and MLH1 promoter hypermethylation have similar prognoses. Genetics Med. **18**(9), 863–868 (2016). [Online]. Available: https://doi.org/10.1038/gim.2015.184

Further Reading

35. D. Mrozek, Scalable big data analytics for protein bioinformatics, in *Computational Biology* (2018)
36. S.C. Basak, M. Vracko, *Big Data Analytics in Chemoinformatics and Bioinformatics: With Applications to Computer-Aided Drug Design, Cancer Biology, Emerging Pathogens and Computational Toxicology*. (Elsevier, 2022)
37. R. Malviya, P.K. Sharma, S. Sundram, R.K. Dhanaraj, B. Balusamy, *Bioinformatics Tools and Big Data Analytics for Patient Care*. (CRC Press, 2022)