# Data management for production quality deep learning models: Challenges and solutions

Citation for the original published paper (version of record):

Munappy, A., Bosch, J., Olsson, H. et al (2022). Data management for production quality deep learning models: Challenges and solutions. Journal of Systems and Software, 191.
http://dx.doi.org/10.1016/j.jss.2022.111359

(article starts on next page)

Contents lists available at ScienceDirect

The Journal of Systems & Software

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

# Data management for production quality deep learning models: Challenges and solutions[☆]

Aiswarya Raj Munappy [a,*], Jan Bosch [a], Helena Holmström Olsson [b], Anders Arpteg [c], Björn Brinne [c]

[a] Department of Computer Science and Engineering, Chalmers University of Technology, Hörselgången 11, 412 96, Gothenburg, Sweden
[b] Department of Computer Science and Media Technology, Malmö University, Nordenskiöldsgatan 1, 205 06, Malmö, Sweden
[c] Peltarion - the operational AI platform, Holländargatan 17, 111 60, Stockholm, Sweden

ARTICLE INFO

*Article history:*
Received 12 October 2021
Received in revised form 22 March 2022
Accepted 3 May 2022
Available online 12 May 2022

*Keywords:*
Deep learning
Data management
Production quality DL models
Challenges
Solutions
Validation

ABSTRACT

Deep learning (DL) based software systems are difficult to develop and maintain in industrial settings due to several challenges. Data management is one of the most prominent challenges which complicates DL in industrial deployments. DL models are data-hungry and require high-quality data. Therefore, the volume, variety, velocity, and quality of data cannot be compromised. This study aims to explore the data management challenges encountered by practitioners developing systems with DL components, identify the potential solutions from the literature and validate the solutions through a multiple case study. We identified 20 data management challenges experienced by DL practitioners through a multiple interpretive case study. Further, we identified 48 articles through a systematic literature review that discuss the solutions for the data management challenges. With the second round of multiple case study, we show that many of these solutions have limitations and are not used in practice due to a combination of four factors: high cost, lack of skill-set and infrastructure, inability to solve the problem completely, and incompatibility with certain DL use cases. Thus, data management for data-intensive DL models in production is complicated. Although the DL technology has achieved very promising results, there is still a significant need for further research in the field of data management to build high-quality datasets and streams that can be used for building production-ready DL systems. Furthermore, we have classified the data management challenges into four categories based on the availability of the solutions.

## 1. Introduction

Deep learning (DL) is fundamentally a neural network with three or more layers. The software systems that employ DL can learn multiple levels of representations (Zhang et al., 2019). DL is a subset of machine learning techniques that use supervised and/or unsupervised strategies to automatically learn hierarchical representations in deep architectures for classification (Bengio, 2000; Ranzato et al., 2007) contrary to the conventional learning methods, which use shallow-structured learning architectures. Consequently, deep learning is now used extensively for image processing in healthcare applications (Litjens et al., 2017), Self Driving Cars (Daily et al., 2017; Rao and Frtunikj, 2018), Virtual Assistants (Kepuska and Bohouta, 2018), Automatic Machine Translation (Bahdanau et al., 2014) and Fraud Detection (Roy et al., 2018). Large companies such as Facebook (Abadi et al., 2016), Google (Beaufays, 2015), Uber (Sergeev and Del Balso, 2018; Gruener et al., 2018), Amazon (Rastogi, 2018), Microsoft (Deng et al., 2014) have employed DL in a wide range of their products. For instance, Facebook uses DL to identify excessively promotional posts, spam, or clickbait (Abadi et al., 2016). Google incorporates DL into the search engine (Beaufays, 2015) to understand spoken commands and questions. Uber implements automated DL transcription technology to enable scalable, reliable, and quick validation of driver identity when drivers go online (Sergeev and Del Balso, 2018; Gruener et al., 2018). Amazon uses DL to improve customers' experience with Alexa skills (Rastogi, 2018). DL is one of the most refined machine learning techniques that does not often require feature engineering. However, DL is not a widely used technique among industries due to poor explainability, poor traceability, data dependencies, dataset incompleteness, and data management problems (Rao and Frtunikj, 2018; Kahng et al., 2017).

[☆] Editor: Burak Turhan.
[*] Corresponding author.
*E-mail addresses:* aiswarya@chalmers.se (A.R. Munappy), jan.bosch@chalmers.se (J. Bosch), helena.holmstrom.olsson@mau.se (H.H. Olsson), anders.arpteg@gmail.com (A. Arpteg), bjorn@peltarion.se (B. Brinne).

https://doi.org/10.1016/j.jss.2022.111359

Although DL models demand less domain expertise and are capable of learning automatically from input data (i.e. the features are not given by a human) (Zhang et al., 2016), understanding and explaining the learned models is extremely difficult. Even the most well-studied models that operate so well remain a mystery, requiring additional research. Consequently, changing the model, its network structure, and hyperparameters demands more effort and time. Further, it has inherent drawbacks, such as high sensitivity to underlying data compared to machine learning models (LeCun et al., 2015). Furthermore, data is treated as a first-class citizen, equal to code, and therefore data management has a significant impact on model accuracy. Although data is everywhere, searching for the right ones in the right quantity itself becomes a challenge (Whang and Lee, 2020). Moreover, the collected data should be clean and validated to rectify problems in the data, so that it will not affect the performance of the DL model (Whang and Lee, 2020). Even after collecting the right data and cleaning it, data quality may still be an issue during model training (Whang and Lee, 2020). The evolving nature of data leads to problems after the model deployment as well (Munappy et al., 2019). Thus, data management becomes a challenge that affects all phases of a DL model development pipeline.

The objective of this paper is to explore, analyze, and understand the data management challenges encountered by industry practitioners when developing and maintaining DL systems. Further, this study aims to analyze why data management for DL cannot be solved by existing solutions from other domains such as Big Data, data analytics, machine learning, and data mining. Using a multiple interpretive case study, we identified the data management challenges for production-quality DL models and mapped them with the corresponding data lifecycle phase, which is published as a conference paper (Munappy et al., 2019). This study is an extension of it by incorporating the potential solutions to data management challenges from the previous research using a systematic literature review (Kitchenham, 2004). Furthermore, we validated the solutions in the second round of multi-case study research with companies that work on real-world DL projects.

The contribution of this paper is four-fold. First, we identify the data lifecycle phases and describe the data management challenges for DL at each phase of the data lifecycle through a multiple case study. Second, we present the solutions that can mitigate the data management challenges discussed in previous research through a systematic literature review. Third, we analyze why not all of these identified solutions presented in the literature are applicable in practice. Based on the identified limitations, we classify the challenges into four categories. Finally, we identify open research challenges in data management for DL and present them as future research directions.

The remainder of this paper is organized as follows. Background is presented next in Section 2 followed by the description of research method in Section 3. Section 4 presents an overview of the data lifecycle phases and data management challenges encountered at each phase. Section 5 presents the solutions for data management challenges. Section 6 shows the categorization of challenges according to the availability of solutions. The findings and research implications of the study are discussed and concluded in Section 7.

## 2. Background

Over the past few years, DL has spawned a tremendous collection of ideas and techniques that were previously believed to be infeasible. At first glance, this collection of ideas appears to be incoherent and dissimilar. However, over time, patterns and approaches evolve, and today DL (LeCun et al., 2015) represents a significant step forward in overcoming the challenges. DL became the center of attraction after Krizhevsky et al. (2012) corroborated the significant performance of a Convolutional Neural Network (CNN) (Krizhevsky et al., 2012) based model on a challenging large-scale visual recognition task (LeCun et al., 1989) in 2012.

### 2.1. Data - The fuel for DL models

Digital data, in all its forms and sizes, is exploding at an incredible rate (National Security Agency, 2013). It also results in a significant paradigm shift in modern scientific research, with data-driven discovery becoming the norm (Manyika et al., 2011). Big data presents unprecedented challenges to harnessing data and information because of the sheer volume of data available today. On the one hand, it presents big opportunities and transformative potential for various sectors. Deep neural networks are trained using massive datasets to imitate human intelligence. Through a hierarchical learning process, DL algorithms extract high-level, complex abstractions as data representations. At each level of the hierarchy, complex abstractions are learned based on comparatively smaller abstractions formulated at the previous level. There are multiple high-performance algorithms in DL that are designed for diverse purposes. No algorithm, however, can guarantee the same results across all datasets, which is a clear indication of the importance of data and its impact on the output of DL models. The core benefit of DL is the analysis and learning of tremendous amounts of unsupervised data, which makes it a useful tool for Big Data Analytics even when raw data is mostly unlabeled and uncategorized. Furthermore, DL algorithms belong to the representation learning class, which has the capability of handling raw data and automatically extracting useful features as the representations (LeCun et al., 2015). As a result, data quality is crucial in determining the performance of a DL model. This combined power of Big Data and Deep Learning when they are working together clearly illustrates the considerable synergy between them.

### 2.2. Synergy between big data and DL

Machine learning and deep learning algorithms are capable of extracting every last detail from the input data, which is then utilized to develop new rules to fulfill the function (Labrinidis and Jagadish, 2012). Data and DL forms a symbiotic relationship in which DL is useless without data and data management is almost impossible to overcome without DL. Business decisions, previously reliant on speculation or painstakingly crafted models of reality, are now based on big data (Oguntimilehin and Ademola, 2014). The sheer volume and variety of data ingested by modern analytical pipelines considerably enhances the links between data integration and machine learning (Dong and Rekatsinas, 2018). Data management systems are increasingly using AI models like machine learning to automate parts of data life cycle tasks. Examples include data cataloging and inferring the schema of raw data (Halevy et al., 2016). Data analytics drives almost every prospect of our contemporary society, including mobile services, retail manufacturing, financial services, life sciences, and physical sciences (Oguntimilehin and Ademola, 2014). Established companies and newcomers alike prefer to use data-driven tactics to develop, compete, and gain value from deep and up-to-date information in most industries (Manyika et al., 2011). However, in the current scenario organizations struggle with collecting, integrating, and managing the data. Instead of solving these data issues, DL will only make them more noticeable.

### 2.3. Data management for DL models

The dimensions of big data are marked by three Vs namely Volume, Velocity, and Variety. Volume denotes the amount of data, variety denotes the number of types of data and velocity denotes the speed of data processing (Tole et al., 2013). The expansion of all three qualities results in the issues of big data management. As users come up with new ways to scrub and process data, the amount of data that can be extracted from the digital universe continues to grow.

Data management for DL can be defined as a process that includes collecting, processing, analyzing, validating, storing, protecting, and monitoring data to ensure the consistency, accuracy, and reliability of the data. Industry products that make use of tremendous volume of digital data can successfully employ DL. However, real-world data needs to be processed and managed before fed as input to the DL models. Training a DL model with such massive and variegated data sets is challenging, and several aspects need to be considered. e.g. data sparsity, redundancy, and missing values. To ensure the high performance from DL models, a set of good data management practices such as data pipelines (Raj et al., 2020) and DataOps (Munappy et al., 2020) should be followed from data collection, through data processing and analysis, dataset preparation, and deployment of the model. Wang et al. (2016) describe how certain challenges like data dependency, memory management, concurrency, data inconsistency can be solved by combining database techniques and deep neural networks. Moreover, ML based data management solutions have a GUI to help understand, visualize, and curate data for training, uncover corrupted data like mislabeled examples, and identify difficult edge cases. Solutions tie into an ML model and its data to determine the data that is helpful, hurtful, or useless for training algorithms. However, they are not good at multiple adjacencies like data labeling, label split balancing, embeddings-as-a-service, model monitoring, and model robustness verification.

Popular companies like Apple (Chen and Lin, 2014), Facebook, Microsoft is collecting a copious amount of data daily through applications like Siri, Google translator, Bing voice search (Jones, 2014) to provide a variety of other services such as reminders, weather reports, personalized recommendations. Although big data can render numerous opportunities, it also enforces consequential engineering challenges (Najafabadi et al., 2015). X. W. Chen et al. describe the big data challenges such as streaming data, high-dimensional data, scalability of models, and distributed computing (Tur and De Mori, 2011). However, in these papers, DL is considered a solution for the management of data. Data management challenges involved in implementing DL models are not seriously considered, and our paper intends to focus on that perspective.

## 3. Research methodology

We have used a three-step research process as shown in Fig. 1. In the first step, we identified data management challenges using interpretative multi-case research. As the second step, we conducted a systematic literature review and identified potential solutions for the identified challenges. Finally, and as the third step, we conducted a second round of multiple case study to validate the applicability of identified solutions on real-world industrial datasets. Detailed steps of the research process are illustrated in Fig. 2

### 3.1. Definition of research questions

The primary objective of this study is to identify data management challenges and solutions specific to DL in real-world settings. The secondary objective is to identify the open research questions in the area of data management for DL. We developed the following research questions to achieve our high-level goal:

**RQ1. What are the data management challenges experienced in the industry while developing DL models?**
**RQ2. What solutions are proposed in literature to address the identified challenges?**
**RQ3. What are the limitations of the existing solutions, and what remain as open challenges in data management for DL?**

### 3.2. Step 1: Interpretive multiple-case study (Exploration)

A multiple case study research method was adopted in this study. According to Yin et al. a case study is most suitable for 'how' and 'why' questions, as well as exploratory 'what' questions (Yin, 2013). In this study, 'what' questions are the key research questions justifying the choice of a case study approach. Further, Stuart et al. (2002) and Meredith (1998) propose that a case study can be considered as the appropriate method to explore new phenomena and generate new knowledge. Furthermore, a multiple case study method facilitates the exploration of the real-life challenges in its context through a variety of lenses (Baxter et al., 2008; Yin, 2003) and enhances the robustness of research findings, compared to a single case study, by reducing the risk of observer bias (Eisenhardt, 1989). Although our primary source of data collection is interviews, information collected through ethnographical observations and minutes from meetings are also incorporated by the two authors from the company wherever necessary, which in turn implements triangulation. The objective of this study is to identify challenges specifically concerned to the management of data in different real-world DL applications. An interpretive multiple-case study approach that adheres to the guidelines by Runeson et al. (Runeson and Höst, 2009) and Verner et al. (Verner et al., 2009) is employed in this research. The challenges identified are grounded on our interpretations of the experiences of experts who build and maintain DL systems in a real-time scenario with real-world datasets. The overall research design and the major steps in the research process of the study are described below.

#### 3.2.1. Overview of Deep Learning use cases

This section describes real-world DL cases that has been chosen for this research. We focus on the system including the DL model, even if our focus is predominantly on the DL part of the system. DL use cases were selected based on the availability of experts working on it. The DL systems discussed are *Online recommender service*, *Medical imaging*, *Energy Prediction*, *Real-Estate Forecast*, *Manufacturing*, and *Financial Systems* as shown in Table 1. All of these are using real-world dataset and are operational.

**Case A — Online recommender services:** Case A is an online recommender system used by an electric vendor. DL components in recommender systems are trained on user reviews and the purchase history. When a customer visits the website, the recommender system predicts users' interest and recommends electric products based on previous customer reviews and purchase history. Online recommender services help the company boost sales by leveraging the power of data. Many customers tend to look at the website for their recommendations. Personalized recommendations from the system thus increase customer satisfaction and thus customer retention. It not only creates personalized

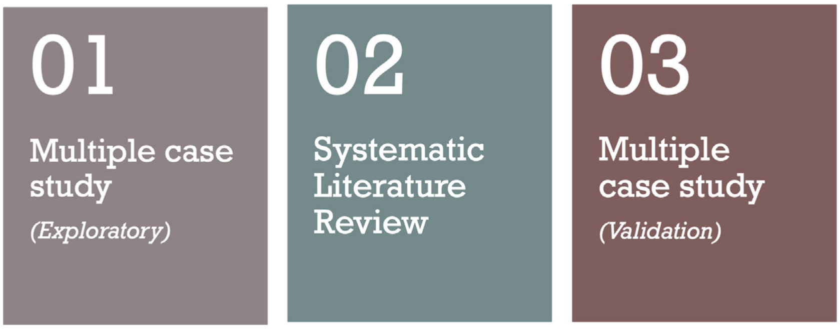


**Fig. 1.** Three-step Research Process.

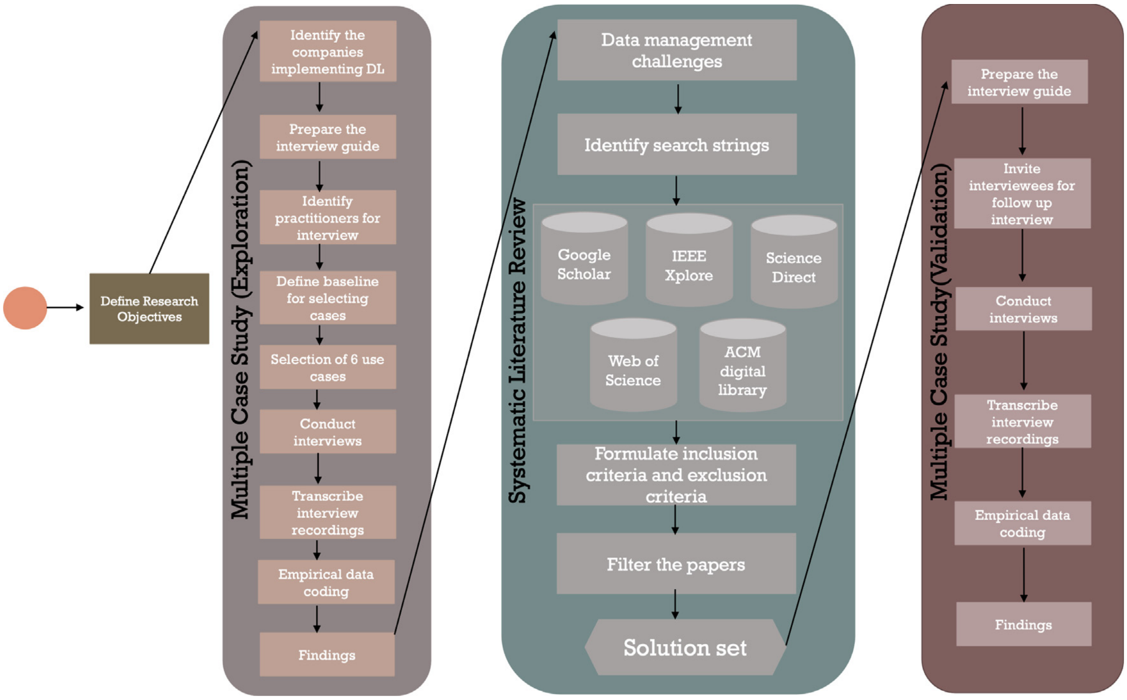

Define Research Objectives
Multiple Case Study (Exploration)
Identify the companies implementing DL
Prepare the interview guide
Identify practitioners for interview
Define baseline for selecting cases
Selection of 6 use cases
Conduct interviews
Transcribe interview recordings
Empirical data coding
Findings
Systematic Literature Review
Data management challenges
Identify search strings
Google Scholar
IEEE Xplore
Science Direct
Web of Science
ACM digital library
Formulate inclusion criteria and exclusion criteria
Filter the papers
Solution set
Multiple Case Study(Validation)
Prepare the interview guide
Invite interviewees for follow up interview
Conduct interviews
Transcribe interview recordings
Empirical data coding
Findings

**Fig. 2.** Research methods and process for conducting the study.

**Table 1**
Deep Learning use cases and description.

| Case No | Deep Learning Use case | Description |
|---|---|---|
| Case A | Online Recommender Service | Predicts users' interest and recommends electric products for them based on previous customer reviews and purchase history |
| Case B | Medical imaging | Automated classification of skin lesions into malicious and benign |
| Case C | Energy Prediction | Wind power is predicted based on the meteorological data |
| Case D | Real-Estate Forecast | Predicts the property prices based on historical data |
| Case E | Manufacturing | Predicts the quality of cartons made from pulp |
| Case F | Financial Systems | Automated classification of transactions into fraudulent and normal |

informational flows independently for each user, but also takes into account the behavior of all users of a service. Along with the information about users' interactions with items, there is usually data describing users and items separately. This data could be assorted and heterogeneous — items and users contain textual descriptions, numerical characteristics, categorical features, images, and other types of data.

**Case B — Medical imaging:** Case B is a melanoma detection system. Melanoma is a type of skin cancer, which is not usual like basal cell and squamous carcinoma, but it has dangerous implications since it tends to migrate to other parts of the body. Therefore, early detection is required to prevent it from spreading to other parts; otherwise, it becomes incurable. The skin cancer detector not only intends to diagnose whether a person has skin cancer or not, but also the type of cancer and severity. Here, the DL system is used for the diagnostic classification of dermoscopic images of lesions of melanocytic origin. Although datasets such as MED-NODE, ISIC Archive, are publicly available, dealing with real-world data is still challenging. Automated classification of skin lesions using images is a difficult task because of the unavailability of fine-grained varieties of the appearance of skin lesions. In this use case, datasets are formed over several years by working in close collaboration with clinics. The company has regulations on the usage of the dataset and the data is not allowed to leave the servers. With these regulations, the practitioners conform to the rules specific to the dataset and move the code and model to the server where data is stored for developing the DL system. The melanoma detector is still not production-ready.

**Case C — Wind power prediction:** Case C is a wind power predictor. Wind power is weather-dependent, and therefore it is irregular and fluctuates over different time scales. Thus, accurate forecasting of wind power is considered as a major contribution to reliable large-scale wind power integration. Here, the DL system is utilized to predict accurately the amount of electricity and power the wind turbines are going to produce within 24 to 48 h so that an accurate report can be furnished to the power companies for which energy is supplied. In this use case, a combination of wind and weather is predicted, from which the power generated by the wind turbines is calculated. The wind power is predicted based on the meteorological data obtained from the National meteorological agency. Deep Learning is used to forecast weather and thereby predicting the wind power that can be generated in the future. The power companies have strict requirements in terms of the amount of power they are going to deliver, and penalties should be paid if they cannot deliver the reported energy.

**Case D — Real-Estate Forecast:** Case D is a system used for real-estate forecasting. Predicting property values is of significant interest to various parties in an economy. Estimation of the house price is important to prospective homeowners, developers, investors, appraisers, tax assessors, and other real estates market participants, such as mortgage lenders and insurers. Real-estate investors and portfolio managers prepare and conduct their investment decisions grounded on periodic evaluations of their real-estate portfolios. Individuals are interested in knowing the values of their properties before determining their list prices. Tax authorities levy property taxes based on estimates of the value of the properties. Banks and mortgage providers conduct housing collateral valuations to qualify the borrowers for their mortgage applications. Initially, house price was predicted based on the comparison between cost and sale price and there were neither accepted standards nor certification processes. Therefore, the house price prediction system was used to fill up the existed information gap, and it also enhanced the efficiency of the real estate market. The house price prediction case was initially built on a traditional assorted database system where SQL queries and data pipeline scripts were used, whereas now it utilizes DL techniques where the model is trained with historical sales data about properties, geography, and demography in the Swedish market. The house price prediction system is a long-running DL system deployed in production and is used by many banks in Sweden.

**Case E — Manufacturing:** Case E is from the manufacturing domain, which is a paper mill industry that produces paper from pulp. The pulp is dried to form cartons and cardboard, which is further used for making milk cartons. The company produces a huge amount of paper board every year and wants to keep material costs as low as possible while retaining high quality. Paper mills gather large amounts of data, which provides them with ever-growing visibility into their processes, due to the rapidly increasing availability of instrumentation and the usage of centralized data historians. The quality of the paper board is predicted based on data from process sensors and images of wood fibers taken with the PulpEye technology. A DL component is incorporated in the system to predict the quality of the resulting product based on all the measurements in the machine and measurements on the pulp that goes in. Further, there are also images of what happens at the start of the machine, and microscope images of the fibers in the pulp. The DL system serve as a foundation to control the manufacturing process so that the same quality could be maintained with less input material and waste.

**Table 2**
Description of Use cases and Roles of the interviewees.

| Case | Use case | Interviewed Experts | |
|---|---|---|---|
| | | ID | Role |
| A | Recommending products to the users in a personalized fashion | P1 | Principal Data Scientist |
| B | Predicting the wind power using the historical weather data | P2<br>P3<br>P4<br>P5 | Data Scientist<br>Head of Data Analytics team<br>Data Scientist<br>AI Research Engineer |
| C | Estimating and predicting the price of houses | P2<br>P3 | Data Scientist<br>Head of Data Analytics team |
| D | Automated classification of skin lesions into benign and malignant | P2<br>P3<br>P4<br>P5 | Data Scientist<br>Head of Data Analytics team<br>Data Scientist<br>AI Research Engineer |
| D | Detecting the credit card frauds during gaming | P2<br>P3 | Data Scientist<br>Head of Data Analytics team |
| E | Predicting quality of paper boards | P4<br>P5 | Data Scientist<br>AI Research Engineer |

**Case F — Financial Systems:** Case F is a financial fraud detection system. Frauds in finance still amount to significant loss of money. Hackers and fraudsters all over the world are experimenting with new techniques to perpetrate financial fraud. Therefore, trusting financial fraud detection systems programmed based on the conventional rule-based method alone will not serve the purpose. This is where Deep Learning shines as a unique solution. The DL system utilizes customer details such as payment history, activity history and payment request data such as payment method, amount, location, etc. For fraud detection, post-payment signals of anomalous pay are also considered. In the current world, fraud detection requires a complete strategy that matches data points with activities to determine what is wrong. When it comes to modeling fraud detection as a classification problem, the main challenge is that in the real world, the majority of the transactions are not fraudulent. However, to train DL systems, counterexamples are also required.

Based on the study with the aforementioned use cases, data management challenges are identified. This study presents a set of clearly explained data management challenges faced by practitioners while integrating DL components in real-world software systems. We have also classified the challenges according to the data life cycle phase in which they are encountered.

### *3.2.2. Expert interviews*

The primary objective of our study was to explore data management challenges encountered while implementing DL systems in real-world settings. Each case in the study pertains to a software-intensive system that incorporates DL components developed at an organization. For the study, a sample pool of DL experts who has an experience of minimum 3 years working exclusively on DL systems and works in seven different domains were selected by their expertise in the area of study. The selected seven practitioners include two authors of this paper. From the acknowledgment in the literature (and our experiences when soliciting interviewees), it can be inferred that only a few experienced practitioners are skilled in the area of intersection between DL and SE, Table 2 illustrates the roles of our interviewees in implementing use cases across multiple domains.

### 3.2.3. Data collection

Semi-structured interviews were used to acquire qualitative data. Based on the objective of the research to explore data management challenges for DL systems, an interview guide with 40 questions categorized into four sections was formulated. The first and second sections focused on the background of the interviewee. The third section concentrated on the importance of data in various projects and the last section inquired in detail about data management, the challenges faced during every phase of the data processing pipeline. The interview guide was reviewed by the authors and some additional questions were added, a few similar questions were merged, and some less relevant questions were removed, finally forming an interview protocol with 36 questions spread across five different categories. All interviews were face-to-face except for one which was done via video conference and each interview lasted 45 to 55 min. All the interviews were recorded with the permission of respondents and were transcribed later for analysis.

### 3.2.4. Data analysis

The audio recordings were sent to be transcribed after the interviews, and the first author wrote a synopsis of each interview, summarizing the important points of discussion. After transcription, the analytical insights from the summary were cross-checked repeatedly with the audio recordings and interview transcripts. The results of data analysis were checked by second and third author of this paper to reduce the bias. A theoretical thematic data analysis approach was opted for coding (Maguire and Delahunt, 2017). First, the author coded each segment of the interview transcript that was relevant to or captured something interesting about data in NVivo. In the first iteration, the aim was to identify the phases of the data life cycle. After identifying the phases as shown in Fig. 2, a second iteration was performed to code the data management challenges encountered in each phase by setting high-level themes as *(i) Data Collection, (ii) Data Exploration, (iii) Data Preprocessing, (iv) Dataset Preparation, (v) Data Testing, (vi) Deployment, (vii) Post-deployment*. Further, the codes such as problem description, implications, empirical basis, examples etc. were formed. The results deduced from the analysis were tabulated and sent to the authors for comments, and then the final summary of the cases and mapping were sent to the interviewees for validating the inferred results.

## 3.3. Step 2: Systematic literature review

As the second step in our research, we conducted a systematic literature review (SLR) based on the guidelines proposed by Kitchenham (2004). In this study, the goal of the SLR was to identify the solutions for data management challenges identified through multiple case study. As we already identified the data management challenges through interview study, we have only searched for solutions to the identified challenges in the literature. We have not attempted to expand the list of data management challenges through literature review. According to Kitchenham, systematic literature reviews not only aggregate all existing evidence on a research question, but also intend to support the development of evidence-based guidelines for practitioners. Further, SLRs help to identify less explored areas and thus provide a framework to position new research activities. Moreover, systematic literature reviews aim to identify, analyze, and interpret all relevant studies on the topic of interest (Kitchenham, 2004). In this study, our topic of interest was data management challenges for DL and solutions. From the topic, the aim was to investigate the data management challenges experienced by practitioners while implementing DL components in software-intensive systems. The interview study was performed followed by a literature review to identify the data management challenges. Further, we identified proposed solutions from literature as well as from the practitioners. The SLR process is summarized as three phases: (1) planning the review, (2) conducting the review, and (3) reporting the review. Definition of research questions, identification of research, selection of primary studies, study quality assessment, data extraction, data analysis, and synthesis are the steps that are described in detail below.

### 3.3.1. Selection of relevant studies

To analyze all available empirical materials specific to the objective of this research, we started with the formulation of a formal search strategy after defining our research goals and questions.

*Search strategy.* Google Scholar, IEEE Xplore, Web of Science, and ACM Digital Library databases were searched since they include journals and conferences focusing on data management as well as DL. These databases allow us to perform keyword searches. The search was conducted in April 2020 and therefore this SLR includes studies that were published before the date. To collect the maximum number of relevant papers, we did not restrict ourselves to selected journals/conferences.

*Search string.* We formulated search strings that included searched the three important keywords in our four research questions. Further, we supplemented the keywords with their synonyms, resulting in the following search string: (((Data) AND (metadata OR labeling OR aggregation OR GDPR OR duplication OR redundant OR heterogeneous OR dirty OR categorical OR transformation OR sequences OR time-series OR extraction OR overfitting OR regularization OR cross-validation OR feedback) AND (challenges OR problems OR issues OR characteristics) OR (technique OR methods OR approaches)) AND (data OR data management OR data analytics OR machine learning OR data mining)) AND ("(industry OR company OR validation OR empirical)")

### 3.3.2. Study selection

We formulated the inclusion criteria and exclusion criteria to select the papers relevant to the study.

**Inclusion criteria (IC)**

1. A research paper that describes the data management for deep learning/machine learning in industrial settings,
2. A research paper that explicitly describes solution for a particular identified data management challenge with validation

**Exclusion criteria (EC)**

1. Non-scientific papers
2. Non-English and duplicates
3. Proposals without industrial/academic validation

We conducted a literature review to identify the data management methods for solving challenges associated with the data for DL. DL is a widely used technique among large-scale companies like Facebook, Google, Uber, etc. We were only interested in identifying the papers that discuss data management solutions for DL models. As data and challenges around it are discussed in various contexts, solutions for data management challenges can be adopted from other data domains as well. Therefore, we extended our search to data analytics, data management, and data mining papers to identify solutions for data management challenges. We have only included the papers that validate the solution with some datasets.

The paper titles were examined to filter studies that were unrelated to our search objective. We reviewed the abstracts and keywords in the remaining studies to select relevant studies. In many cases, an abstract and keywords were insufficient to establish whether a study was relevant. In such cases, we also went over the conclusions. Further, we filtered the remaining studies by applying the inclusion/exclusion criteria. The initial primary studies for the SLR are chosen from this final step. A complete list of included papers (primary studies as well as supporting studies) is provided in the Figshare.

#### *3.3.3. Data extraction*

We defined a data extraction process according to the guidelines provided in Kitchenham (2004) to identify relevant information from the 32 included primary studies that pertain to our research questions. Our data extraction process includes the following: First, we set up an Excel table to record ideas, concepts, and findings of each of the 58 papers.

#### *3.3.4. Data analysis and synthesis*

The solutions are classified according to the data life cycle phase in which they can be used to mitigate the challenge. The solution was analyzed to check if it was applicable in the context of deep learning. Furthermore, the solutions were again classified into expensive and cheap in terms of infrastructure and cost-effectiveness. Each of the identified solutions was thoroughly analyzed to interpret the reason for it being counted as a challenge in the deep learning context even with the solutions from other domains.

### *3.4. Step3: Interpretive multiple case study (Validation)*

We have conducted a second round of study for validating the results obtained after performing the first two steps. The objective of this follow-up study is to validate the solutions we identified from the literature. We followed exactly the same procedure as step 1. The interviewees who participated in the first round of multiple case study were contacted through email. We shared the results of the first study in the form of a conference paper and the solutions from literature as a separate document, and requested them to go through those documents if they are interested in participating in the validation study. After 1.5 weeks, they were contacted again with an interview guide for validation study. Most of them showed interest in participating in the study. Therefore, we conducted the follow-up study with the interview guide we prepared for validating the solutions. All the interviews lasted for around 30–45 min. Interviewees were asked for the changes in the availability of solutions for the identified set of challenges. We also asked if they are familiar with the solutions from literature. Further, we inquired in detail about the solutions identified from literature and their reasoning for not using it in the industries. The interviews were recorded with the permission of interviewees and transcribed for performing data analysis. First author performed open coding and identified high level codes such as working solutions, used solutions, non-general solutions, partial solutions, reasons etc. The codes were reviewed by the second and third author.

### *3.5. Threats to validity*

The interview study can have some threats to validity such as construct, internal, and external threats, credibility and transferability. This section explains the mitigation strategies used to administer these threats.

#### *3.5.1. Construct validity*

A few cases were removed from the results to verify construct validity, as some interviewers did not have a thorough comprehension of the concepts covered. Our study includes certain limitations with multiple interviews as a result of the screening process. This limitation, on the other hand, can be viewed as an opportunity for future research. The results of the interviews were disclosed to the participants in order to reduce researcher bias. We also created a semi-structured interview guide and delivered it to the interviewees prior to the interviews. Before the interview, the interviewees were supplied a brief synopsis of the topic to be discussed. We rephrased the question of whenever the response becomes off-topic, or asked them to elaborate when we received ambiguous answers. Further, confusions or lack of clarity were resolved by contacting the participants.

#### *3.5.2. Internal validity*

As the researcher only had limited access to the descriptions of the strategies, it was not possible to investigate about the other influential factors that can affect the final results. To minimize internal validity threats, two of the co-authors, who has in-depth knowledge about the data processed in the company, were asked to validate the findings. Moreover, the paper was sent to the steering committee for review before the final submission.

#### *3.5.3. External validity*

The presented study is derived from the cases studied with different teams in the domains of manufacturing, banking, business and healthcare. Some parts of the work can be seen in parts of the company differently. All company terminologies are normalized, and implementation details are given at the appropriate degree of abstraction (Bickman and Rog, 2008). We do not claim that the opportunities and challenges will be the same for companies from different disciplines.

#### *3.5.4. Descriptive validity*

Descriptive Validity refers to the accuracy and objectivity of the data collected during the case study. In order to mitigate the problems with factual accuracy and objectivity, all the interviews were recorded with the permission of interviewees. Further, interviewees were contacted through email whenever lack of clarity is encountered while transcribing the recordings. Two co-authors from the company also helped with certain company specific terminologies. Thus, we rule out the chance of misinterpreting what participants say and do, as well as their perspective on what is going on, which is a crucial way of detecting biases and misconceptions of what is interpreted.

#### *3.5.5. Credibility*

To increase the credibility of the results obtained through literature review, we conducted a second round of interview study with the practitioners participated in the first round.

#### *3.5.6. Transferability*

The degree to which qualitative research findings can be generalized or transferred to different contexts or settings is referred to as transferability. Due to the non-disclosure agreement with the participants of the case study, we have limitations in disclosing the entire details gathered during data collection. We agree that this impacts the transferability and to reduce this, we have tried to include details wherever possible with the help of anonymization

A strength of our study is the use of both multi-case study approach and systematic literature review for framing the results. However, several factors present some potential threat to the validity of the systematic literature review.

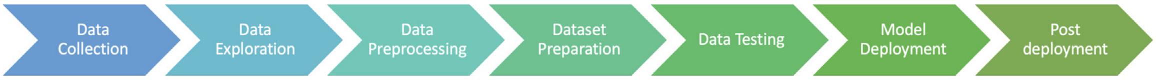


**Fig. 3.** Data lifecycle phases.

**Table 3**
Mapping between data management challenges and DL use cases.

| Phase | Challenge | Use cases of DL components | | | | | |
|---|---|---|---|---|---|---|---|
| | | RS[a] | WPP[b] | HPP[c] | MD[d] | FFD[e] | MS[f] |
| Data Collection | Lack of labeled data | X | X | X | X | X | X |
| | Data Granularity | X | X | | | | |
| | Shortage of diverse samples | | X | X | X | X | |
| | Need for sharing and tracking techniques | X | X | X | X | X | |
| | Data Storage complying to GDPR | X | | | | | |
| Data Exploration | Statistical Understanding | | X | | | X | X |
| | Deduplication Complexity | X | X | X | X | X | X |
| | Heterogeneity in data | X | X | X | X | X | |
| Data Preprocessing | Dirty data | X | X | X | X | X | X |
| | Managing sequences in data | | X | X | | X | |
| | Managing categorical data | | | X | | X | |
| Dataset Preparation | Data Dependency | X | X | X | X | X | X |
| | Data Quality | X | X | X | X | X | X |
| Data Testing | Tooling | X | X | X | X | X | X |
| | Expensive Testing | X | | | X | | X |
| Deployment | Data Extraction Methods | X | X | X | X | X | X |
| | Overfitting | | | | X | X | |
| Post Deployment | Data sources and Distribution | X | X | X | | | |
| | Data drifts | X | X | X | | | |
| | Feedback loops | X | | | | | |

[a] Recommender System.
[b] Wind Power Prediction.
[c] House Price Prediction.
[d] Melanoma Detection.
[e] Financial Fraud Detection.
[f] Manufacturing Systems.

### 3.5.7. Identification and selection of papers

Search string based approach was adopted for identifying and selecting the primary set of papers. This might have resulted in missing of relevant studies and in turn the solutions to the challenges. To mitigate this threat, we conducted a second round of interview study to validate the solutions and sought the help of experts to see if we missed any relevant solutions. Further, data collection period will have an impact on our findings. Many new papers would have been published after we did our search in various databases.

### 3.5.8. Exclusion of gray literature

Our study is also limited by the fact that we have not used gray literature or non-academic literature. According to Garousi et al. multi-vocal literature review with both academic and non-academic literature is used when there is limited studies on a specific topic. Since, deep learning is a mature field, we had 58 papers included in the study. Therefore, we did not incorporate the results from gray literature.

## 4. Data management challenges for DL

This section presents the findings from the first step of the three-step research process in Fig. 2 which is an exploratory interpretive multi-case study. First, we illustrate the phases of the data lifecycle. Then, we map the data management challenges to the corresponding phase as shown in Table 3 and describe the challenges. Data Collection, Data Exploration, Data Preprocessing, Dataset Preparation, Data Testing, Deployment, and Post-deployment are the 7 data lifecycle stages as shown in Fig. 3.

### 4.1. Data collection

The data lifecycle starts with the data collection phase. Data collection is the process of acquiring data from a wide range of sources that produces data. Lack of labeled data, data granularity, shortage of diverse samples, data sharing and tracking methods, data storage complying with GDPR are the challenges encountered during the phase of data collection.

### 4.1.1. Lack of labeled data

*Description:* Success of supervised DL algorithms is underpinned by labeled data with sufficient quality data labels that are required for training, testing, and validation. Data labels are obtained through the process of transcribing, tagging, and labeling significant features within the data. With high-quality, human-powered data labeling, companies can build and improve DL implementations. As a result of the disparate origins, unlabeled data and noisy labels increase the complexity. Further, to label data, metadata is required for the practitioners, as they might not be experts in the domain where they develop DL models. According to the practitioners that participated in the multiple case study, lack of metadata creates confusion and a poor understanding of the data. Further, the semantics of the data is often obscured due to poor organization, leading to ambiguities.

*Implications:* As most of the companies use supervised learning for training their DL models, lack of labeled data is considered as one of the biggest challenges. The impact of the absence of metadata challenge varies for different types of data. For instance, the dataset for building stock market price prediction will have opening price, closing price, quoted price, session price, etc. When the metadata is missing, it is hard for practitioners who develop DL models to identify and distinguish different prices. On the other hand, if the dataset consists of images, audio, or video, metadata extraction methods can be employed to extract the labels for that dataset. Another associated challenge is that without metadata, it is difficult to analyze if some pattern makes sense or not. For example, a particular column representing the temperature of a location cannot be always zero, while the column indicating the value of the on/off switch can have zero all the time.

*Empirical basis:* Recommender system, wind power prediction system, house price prediction, melanoma detection, financial fraud detection, manufacturing system has encountered both lack of labeled data problem and absence of metadata problem. For instance, practitioners sought help from a doctor to label the skin lesions while developing a melanoma detection system.

#### 4.1.2. Data granularity

*Description:* Data collected over a period is aggregated and stored in a database or data warehouse. Data aggregation may remove important data points that cannot be collected again. Thus, fine granularity or details in the data is lost through data aggregation techniques. Like in mobile networks, counter data is collected and aggregated to a value over 15 min, and it is stored. Because saving every second's data point is expensive.

*Implications:* As a result of data aggregation, a significant amount of information is lost, which could mean that even though a lot of data are in place while looking at the details, the granularity needed for a use case will not be there. Therefore, even if data is collected over ten years, the problem is still limited by data collection choices which are difficult to get around.

*Empirical Basis:* In our study, the recommender system case experience this data granularity problem. When the reviews from users are all logged for a long period and handed over for building recommender systems, but failed to log the user's identity, the data granularity is lost. Further, combining the data with another dataset on the user level becomes impossible. Wind power prediction systems also suffer from this challenge.

#### 4.1.3. Shortage of diverse data samples

*Description:* Upon training, the deep neural network should be given all possible instances and varieties of data so that it will not fail on inputting unseen data in production. However, during data collection, many companies collect numerous abnormal samples and fail to collect the counterexamples of data and vice versa, leading to a class imbalance problem. The DL model needs to be trained with counterexamples as well. For instance, the company which did data collection for financial fraud detection only collected fraudulent samples and failed to save the non-fraudulent transactions in the dataset.

*Implications:* Financial fraud detection cannot be developed only with the samples of fraudulent transactions, the model needs non-fraudulent samples to distinguish between normal and abnormal financial transactions. DL models cannot learn the normal cases by themselves when only the abnormal samples are fed during the training. The models that are trained on such input data will produce weird outputs once deployed in production.

*Empirical Basis:* In our study, wind power prediction system, house price prediction system, melanoma detection, and financial fraud detection system suffers from this challenge. Melanoma detection is the use case that has the highest impact due to this challenge. Collecting malignant samples from patients is a difficult process for the data collection team.

#### 4.1.4. Data sharing and tracking methods

*Description:* Sharing the collected data with the practitioners is required for implementing DL models. There is no defined channel or medium for sharing the collected data. According to the size of the data, companies choose different means of sharing. Some companies opt to share the data in the form of Excel files over email, FTP server, or even in the form of physical tapes. According to the practitioners, data for wind power prediction were given in the form of huge magnetic tapes, while the data for house price prediction was sent through email.

*Implications:* Two of the experienced practitioners identify data tracking as an important measure by which data quality can be assured. However, due to the tight limit on time and resources, often data tracking is not focused much or is kept at the least priority, leading to poor data quality.

*Empirical Basis:* All use cases except melanoma detection systems do not have data sharing and data tracking methods. Data pipelines are used for melanoma detection use cases, which enables sharing and tracking of data.

#### 4.1.5. Data storage complying to GDPR

*Description:* DL systems are powerful and have the potential to memorize every piece of information given to it. Thus, the amount of training data has the biggest impact on the performance of the model. General Data Protection Regulation (GDPR) is a regulation in EU law to protect online personal data. GDPR is a set of legislative rules which impose restrictions on the processing and storage of information. Major companies that focused on collecting and maintaining datasets can build better DL models to a certain extent. The problem with small-scale companies is that they do not have clear knowledge on how to collect and store data complying with the rules of GDPR, and there is no framework or protocol to help them to do data storage efficiently.

*Implications:* In such cases, a certain percentage of revenue needs to be paid as a penalty for not following the regulations of GDPR, which end up in the deletion of a huge portion of data they collected over time.

*Empirical Basis:* Recommender system's use case encountered this challenge and lost a considerable amount of data as they had to delete the data to comply with the GDPR. Even though this is a problem experienced by only one case in the entire study, it is important as it has significant legal and financial complications involved.

To summarize, data management challenges at the data collection step are lack of data labeling techniques, unavailability of fine-grained data, shortage of diverse samples, lack of data sharing and tracking methods, and data storage guidelines following GDPR.

### 4.2. Data exploration

Data exploration, or exploratory data analysis (EDA), is a process to analyze and understand the data with statistical and visualization methods. This step helps to identify patterns and problems in the dataset, as well as for deciding which model or algorithm to use in subsequent steps. Statistical understanding, data deduplication complexity, and data heterogeneity are the major challenges encountered at this phase of the data lifecycle.

#### 4.2.1. Statistical understanding

*Description:* When confronted with data that needs to be analyzed, the first step is to carefully identify the distribution of data. Statistical understanding is much required for determining the distribution of data. Even with sufficient knowledge in statistics, it is challenging to identify the distribution of data. The normal distribution or Gaussian distribution is that nice, familiar bell-shaped curve. But, data comes from a range of devices out in the wild, and there is no point in assuming an easy to handle normal distribution.

*Implications:* For instance, consider an image processing application, to model the pixel values efficiently, the assumption of Gaussian distribution is inaccurate as it violates the boundary properties. In such cases, models like BMM (Beta Mixture Model) should be used. Without clear knowledge of statistical distributions, it will become difficult to model the distribution.

*Empirical Basis:* Practitioners mentioned that they have encountered this challenge while developing Wind power prediction systems, financial fraud detection systems, and manufacturing systems.

### *4.2.2. Data deduplication complexity*

*Description:* A dataset often has a lot of duplicates, some with slight variations and some exact copies. Consequently, analyzing the dataset for duplicates and de-duplication is a complex task. For example, consider a song recommender system trained on a dataset of songs. If you take a random song, there can be 200 versions of the same song with slight variations in it, but it is more or less the same song. If the model is trained with such a dataset, the result may turn out horrible, such that it may recommend 50 copies that sound more or less the same. In such cases, de-duplication becomes complex. Because if the dataset has 100,000,000 songs, you need to compare a song with every other song in the dataset. Therefore, it is a quadratic complexity of that problem.

*Implications:* It is impossible to do from a time point of view in a single machine, and it is required to run it on hundreds and hundreds of machines.

*Empirical Basis:* According to the practitioners, they have encountered this challenge for all use cases presented in this study. Especially, with the melanoma detection model, it was almost impossible for them to deduplicate the images in the dataset.

### *4.2.3. Data heterogeneity*

*Description:* Different data sources may offer conflicting information. Moreover, rapidly growing multimedia data from the Web and mobile devices consists of a huge collection of still images, video and audio streams, graphics and animations, and unstructured text, each with different characteristics. The major challenge here is to find a method that can resolve the conflicts and fuse the data from different sources effectively and efficiently. The majority of current DL algorithms are evaluated on bi-modalities (i.e., data from two sources). However, format, size, and encoding techniques vary from data to data. A single dataset itself may have data in audio, video, and text formats. If a dataset containing only textual data is examined, some text will be in UTF-8, others in UTF-16, some in CSV, comma-separated format, others in tab-separated format, some with HTML code embedded in the actual text, and others with something strange like placeholders embedded inside the actual national language text.

*Implications:* It is required to invest a significant amount of time and effort in just transforming the text into a uniform format and coding for the data.

*Empirical Basis:* All six use cases studied here have encountered this challenge. Moreover, the system performance decreases for significantly enlarged modalities. Furthermore, practitioners lack the idea about levels in DL architectures appropriate for feature fusion with heterogeneous data.

Statistical understanding, data de-duplication, data heterogeneity are the main data management challenges encountered in the data exploration phase.

## *4.3. Data preprocessing*

Data preprocessing is an integral step in DL, as the quality of data and the useful information that can be derived from it has direct impact on the model's learning ability. Therefore, data must be preprocessed before feeding it into the model. Dirty data, managing sequences in data, and managing categorical data.

### *4.3.1. Dirty data*

*Description:* Raw data is typical with imperfections like missing values, wrong values, and ill-formatted values. These unclean or noisy data are known as dirty data. Deep neural networks are good at deriving patterns from the given input. So, it is dangerous to feed noisy data to the DL models. Also, the DL experts might not be experts in the domain, and so they are unaware of what needs to be filled when there are missing values and how to identify the wrong or ill-formatted values. For example, if there is a column for age and some values are missing. The system is supposed to make predictions based on each user, and you do not have the age for 10% of them. That column can be filled out with the average or minus one. To fill the column, it is required to know what the column is meant to be and what can be filled in to replace the missing/wrong/misformatted values. All practitioners mentioned the unclean data issue in all the cases they handled, and in most cases, discussion with the people who collected the data was the only practical solution.

*Implications:* Dirty data impacts are related to error type and error rate. Thus, the error rate of each error type in the provided data should be necessarily detected. Testing accuracy decreases with increased noise in data.

*Empirical Basis:* According to the input from practitioners, all datasets have the problem with dirty data, and it takes an enormous amount of time to fix the noise in the data.

### *4.3.2. Managing sequences in data*

*Description:* Metadata management should be considered with equal importance in managing the sequences in data. Storing the sequencing data alongside the contextual metadata is a bit challenging, especially when the data quantity is too large. For instance, for chronological data, there is a time series that needs to be divided chronologically somehow, so we do not end up predicting the past. Missing the time steps in the sequence is another associated challenge. i. e, data has variable-length sequences by definition. Those sequences with fewer time steps may be considered to have missing values.

*Implications:* Short gaps in the sequential data can be imputed using the before and after data present in the sequence. However, when the gap increases, it consists of more contiguous missing values, and consequently, the amount of information in the sequence is often not sufficient to impute the gaps.

*Empirical Basis:* House price prediction system and wind power prediction system have the problem with missing time steps.

### *4.3.3. Managing categorical data*

*Description:* Variables containing label values rather than numerical values are referred to as categorical data. Nominal, ordinal, interval and ratio are examples of categorical variables. County — Tuscaloosa, Mobile, Walker, and so on are examples of nominal variables with attribute values that have no natural order. The difference between values (e.g., Letter Grade — A, B, C, D, F) does not have a natural order in ordinal variables. The difference between the two values is meaningful (e.g., Age of Driver — 22–24, 25–34, 35–44, etc.). Interval variables are constructed from intervals on a continuous scale. A ratio variable has all the properties of an interval variable, but it also has a distinct definition of 0.0. (e.g., Weight). DL models cannot operate on label values as it requires all input variables in the numeric form. Even though one-hot encoding is used very frequently, it can be frustrating during implementation.

*Implications:* When there are thousands of categories, the complexity increases. For example, if there is text data that needs to be cleaned up and converted to numeric form, then it might not be possible to do it with a laptop or even a big server. There are core systems like Hadoop, Spark, or Google Data Flow where big data processing can be done. However, it is still very dependent on the person doing it, what they are comfortable with, and also the data, how big is it, how difficult is it, and what needs to be done with it. There are no predefined sets or standards to handle this.

*Empirical Basis:* House price prediction system, financial fraud detection system have more categorical data compared to the other use cases.

To summarize, dirty data is the major challenge in this phase, and reducing its effect needs an enormous amount of time and effort from the practitioners. Managing categorical data, as well as sequential data, are the other two challenges encountered in this phase.

### 4.4. Dataset preparation

After the data is preprocessed, the dataset is split into two parts: training and testing datasets. The training dataset is again split into training and validation datasets. The ratio of these two splits varies according to the size of the dataset, developer, and type of dataset. Nevertheless, the typical practice is to split the dataset in an 80:20 ratio for training and testing respectively.

#### 4.4.1. Data dependency

*Description:* Data leakage is the challenge of not splitting the training and validation/test dataset properly so that the training data for the model happens to have the data which needs to be predicted. For instance, data leakage happens when the same data instance occurs in both training and testing datasets.

*Implications:* Data leakage hides the actual performance of the model and when it is exposed to new and unseen data, the performance will not be as expected. So proper attention should be taken while splitting the dataset. Based on the study, we could infer that checking the data distribution is not always a solution to reduce data dependency.

*Empirical Basis:* All use cases discussed in this paper have encountered this challenge.

#### 4.4.2. Data quality

*Description:* Quality of data is crucial, as poor quality data can cause severe performance degradation and exaggerated results. Data consistency is one of the factors deciding the quality of data. However, consistency is hard to achieve the target in many applications. For example, based on our study, the images collected from the hospitals are all taken in different conditions with different lighting. Accuracy, completeness, validity, and timeliness are some other factors that ensure the quality of data. However, there is no exhaustive list of factors that should be checked to ensure the quality of data, which is challenging. Although the dirty data problems are fixed in the data exploration phase, DL experts mentioned that they recheck the data quality during the dataset preparation stage. Data exploration primarily looks for dirty data problems with the help of domain experts. However, in the dataset preparation phase, it is usually the DL expert looks if he/she has sufficient quality data before feeding it as input to the model.

*Implications:* DL models learn by adjusting their internal parameters with massive quantities of training data until they can consistently discern comparable patterns in data they have never seen before. Therefore, a deep learning model is acutely sensitive to the quality of the data. Because of the huge volume of data required, even relatively small errors in the training data can lead to large-scale errors in the system's output.

*Empirical Basis:* Data quality is a typical challenge faced by all practitioners, and all the datasets discussed here have a problem with data quality.

To summarize, data quality is the main challenge experienced in this phase. Identifying what all factors determine data is a challenging task, and it, in turn, creates trouble while establishing tests for checking quality.

### 4.5. Data testing

Data testing is the phase in which practitioners decide whether to retrain the model or not. When the test cases fail, it indicates that there is a possibility of change in the underlying data. According to the experts participated in the interview, data testing is done at least once before the model deployment. After the deployment, during the continuous monitoring if data drift is encountered, data testing is done again to check if retraining is required or not. Therefore, data testing is a step that is conducted multiple times in the data life cycle.

#### 4.5.1. Expensive testing

*Description:* Testing the data is a critical step that ensures the data quality and reduces the possible occurrence of defected data that affects the efficiency of the process. Absent, obsolete, or wrong test data may prevent the practitioners from executing the test cases or produce unreliable test results.

*Implications:* Data testing is highly expensive in the sense that it requires a lot of effort and time to define and automate test-cases specific to DL models. It is pretty hard to do regression testing on data as the data is collected from users out in the wild, where exerting control is impossible.

*Empirical Basis:* Practitioners who work with recommender systems, melanoma detection system and manufacturing systems has mentioned this challenge.

#### 4.5.2. Data management tools

*Description:* Tooling is a challenge in most of the phases of the data lifecycle. Although tooling is a challenge experienced at most of the data lifecycle stages, many of the tools are under development. For instance, there exists data cleansing tools such as TIBCO clarity, Data Wrangler etc. helps with cleaning the dirty data. However, tools for data testing is still an unexplored area. The major advantage of conventional software systems is that there exists a large variety of tools, especially for testing. As DL is a recently emerged approach, tools for testing such models are yet to be developed.

*Implications:* With the help of data management tools, practitioners can easily develop and deploy DL models.

*Empirical Basis:* Although cloud has good quality tools and services, none of the companies who have participated in the study can use that due to the policy restrictions. Therefore, all the cases included in our study experience tooling problems.

The main challenge in this phase is the lack of tools for creating test cases for DL systems. Existing methods are highly expensive.

### 4.6. Model deployment

The process of running an application on a server or device is known as model deployment. All the stages, processes, and activities required to make a DL available to its intended users are included in the model deployment. Data extraction methods and overfitting are the two challenges encountered in this phase.

#### 4.6.1. Data extraction methods

*Description:* Training-serving skew is a typical problem encountered when running DL models in production, where the data encountered at serving time is significantly different from the data used to train the model, leading to reduced prediction quality. *Implications:* For example, Google once built a system called quick access in Google Drive which recommends a list of documents to open. When the system was built, they first extracted data and made a training dataset, trained a model on it, and did the evaluation which looked great. So, they put it in production, and it did not work. When they investigated, they discovered that they had a specific pipeline for extracting data for training, but when they deployed it in production, they had data extracted from an API, which did not match the extraction they had for training. Thus, some additional transformation happening in the API prevented the model from working as expected.

*Empirical Basis:* This challenge was encountered during the deployment of all the DL models discussed in this paper. Practitioners mentioned that the main reason for the challenge is the lack of tracking of data extraction methods. When an expert is replaced by another one, the new expert might not exactly know what data extraction method was used during the development phase and use a different extraction method during deployment, leading to training serving skew.

#### 4.6.2. Overfitting

*Description:* Overfitting is the situation when a deep neural network memorizes and fits itself so closely with the training set that it loses the capability to generalize and make predictions for new and unseen data.

*Implications:* An overfit model can cause the regression coefficients, p-values, and R-squared to be misleading. Because an overfitted model will be tailored to the specific data points that are included in the training sample and not generalizable outside the training sample. Therefore, it gives poor performance to unseen data.

*Empirical Basis:* For instance, in melanoma detection use case where tabular data is used along with images, it turned out that the model was just learning the ID number of a certain hospital, and that hospital was a popular hospital to which the more severe cases were sent. Thus, the model was not learning anything from the images, rather it was just learning that the patients in that hospital are more likely to be sick, which is because they were sent there.

Data Extraction methods and Overfitting are the two main problems encountered in this phase of the data pipeline. Usage of different data extraction methods while training and testing create problems. Overfitting is a common challenge with machine learning as well as DL systems. Overfitting challenge has solutions such as cross validation and regularization when detected during model training. However, during model deployment and post deployment, it is hard to solve.

### 4.7. Post-deployment

Monitoring the models that are deployed in production is an important task in the data lifecycle. Because, change in the data source, distribution, data drifts, etc. can cause serious performance degradation. Therefore, these are considered as challenges after the model deployment.

#### 4.7.1. Changes in data sources and distribution

*Description:* When a certain problem is modeled, a distribution is postulated based on the data available at that time. However, consistency in data distribution cannot be expected all the time. Consider the house price prediction system in our study, which is trained on historical real-estate data. When some sudden environmental disaster or society-wide effect takes place, the usual distribution will be disturbed, and the trend in data changes. Deep neural networks may not always be able to handle new data distributions when they appear. An abrupt change in the data source can sometimes result in unexpected and undesirable effects.

*Implications:* Change in data distribution has a far-reaching impact on any DL model. For example, when building the DL model, data that arrived before a change can bias the models towards characteristics that no longer hold.

*Empirical Basis:* For instance, the distribution of the price of houses shows abnormal deviation after a natural calamity. It is the same with data for wind power prediction systems as well. As a result, an already deployed well-performing model in production may not perform well after the change in data distribution.

#### 4.7.2. Data drifts

*Description:* Data drifts are also known as data shifts that happen over time. When data shifts happen, DL models may deliver weird and erroneous results. Consider systems, such as mobile interactions, sensor logs, and web clickstreams. Whenever the business tweaks or updates happen, the data those types of systems generate changes continuously. The sum of these changes is data drift. Other common examples of structural drift are fields being added, deleted, and re-ordered, or the type of field being changed.

*Implications:* When drift in a model is detected, the next step is identifying which features in the data are causing the drift. It is possible that some features have drifted but have not created a significant change in the model because these features are not very important. Identifying the feature that causes the drift and is of great importance to the model, is crucial to the performance of the model and should therefore receive better attention when retraining the model.

*Empirical Basis:* For example, data for case E, which is a financial fraud detection system, have this challenge. To support a growing customer base, a bank adds leading characters to its text-based account numbers. This kind of data change causes the bank's customer service system to combine data related to bank account 00-56789 with account 01-56789. All practitioners agree that most of the cases that they handle are subjected to this challenge.

#### 4.7.3. Feedback loops

*Description:* Feedback loops are sometimes beneficial and at times detrimental. For instance, feedback loops are inherent to the recommendation systems. Because, the data collected will be mostly from the customers and if good suggestions are given on what to buy, of course, the customers will buy more. As a result, the model sort of reinforces itself.

*Implications:* These feedback loops give rise to the "echo chambers" or "filter bubbles" that have user and societal implications. Systems that lead to a self-reinforcing pattern of narrowing exposure and shift in user's interest are often denoted as "echo chamber" and "filter bubble".

*Empirical Basis:* Feedback loops are not a typical challenge with regression models or classification models. Therefore, only case A has encountered this challenge.

To summarize, the data management challenges that can be found at the post-deployment stage include the change in data sources and distribution, feedback loops, and data drifts. A degenerative feedback loop is a problem specific to recommender systems.

**Table 4**
Mapping between Data Lifecycle phase, Challenges and Potential Solutions.

| Data lifecycle Phase | Challenge | Potential Solutions validated through case study |
|---|---|---|
| Data Collection | 1. Lack of labeled data | 1. Crowdsourcing<br>2. Active Learning<br>3. N-shot Learning<br>4. Unsupervised Learning<br>5. Semi-supervised learning<br>6. Self-supervised Learning<br>7. Help from Domain Experts |
| | 2. Data Granularity | 1. Lossless data aggregation<br>2. Synthetic data |
| | 3. Shortage of diverse samples | 1. N-shot Learning<br>2. Resampling<br>3. Synthetic data generation<br>4. Class weighting with data augmentation |
| | 4. Data sharing and tracking methods | 1. Data Pipelines |
| | 5. Data Storage complying to GDPR | |
| Data Exploration | 6. Statistical Understanding | |
| | 7. Deduplication Complexity | |
| | 8. Heterogeneity in data | 1. Data Pipelines |
| Data Preprocessing | 9. Dirty data | 1. Data Cleaning<br>2. Data Scrubbing<br>3. Data Imputation |
| | 10. Managing sequences in data | |
| | 11. Managing categorical data | 1. Learned Embeddings<br>2. One-hot encoding |
| Dataset Preparation | 12. Data Dependency | |
| | 13. Data Quality | 1. Data Validation Frameworks<br>2. Data Linter |
| Data Testing | 14. Tooling | 1. Cloud based services |
| | 15. Expensive Testing | |
| Deployment | 16. Data Extraction Methods | |
| | 17. Overfitting | 1. Regularization<br>2. Cross Validation<br>3. Model simplification<br>4. Data Augmentation<br>5. Dropouts<br>6. Transfer Learning |
| Post-deployment | 18. Data sources and Distribution | |
| | 19. Feedback Loops | |
| | 20. Data Drifts | |

## 5. Solutions for data management challenges

This section summarizes the solutions to the challenges identified and described in the previous section. The solutions presented were derived from a systematic literature review and validated in the second round of multi-case study research. Further, Table 4 shows the mapping between the data lifecycle phase, challenges, and the identified solutions. Data management existed even before the emergence of DL and machine learning. Therefore, many of the data management challenges were solved in the context of big data analytics, data mining, machine learning, etc. On the other hand, some challenges are very specific to DL. For instance, categorical data challenge, tooling, etc. are very specific to DL. Although there exist solutions for a subgroup of data management problems, industrial practitioners still list them as challenges. Therefore, we did a second round of multiple case study analysis to explore the validity of identified solutions.

### 5.1. Data collection

Challenges such as lack of labeled data, data granularity, shortage of diverse samples, need for sharing and tracking techniques are the challenges encountered in this data lifecycle phase. Crowdsourcing, active labeling, semi-supervised learning, and self-supervised learning are some solutions that can address the lack of labeled data challenges. Few-shot learning, zero-shot learning, less than one-shot learning can be used to address both lack of labeled data challenge and shortage of diverse sample problems. Resampling with data augmentation, synthetic data and data augmentation with class weighting are some other solutions that help with the shortage of diverse sample problems. Data granularity can be addressed with lossless data aggregation techniques and synthetic data. Data pipelines can partially solve the challenge of data sharing and tracking techniques.

#### 5.1.1. Crowdsourcing

*Description:* Crowdsourcing is a brilliant platform to outsource high-volume data labeling jobs to a flexible workforce (Malone et al., 2009). Crowdsourcing usually achieves its purpose by combining the responses of numerous annotators on the same sample. For instance, Nowak et al. (Nowak and Rüger, 2010) considered the problem of multi-label image annotation by experts and ordinary annotators from the MTurk crowdsourcing platform. After calculating various agreement statistics, the authors determined that if annotators are specialists, several labels for an object are redundant with high-quality recommendations for annotators. On the contrary instance, despite annotators' excellent accuracy, a dataset produced by multiple annotators doing a single task is of much greater quality.

Crowdsourcing is extensively utilized to solve problems that are not accessible to automatic calculations and necessitate human intervention. Three things hindrances affect the efficiency of crowdsourcing platforms. The first is quality, which refers to algorithms that accurately discriminate between true and false labels. The second factor is the labeling costs: it is not always feasible to fix an issue by increasing the number of annotators per sample. The third factor is that, in some cases, labeling speed is critical; in these cases, task execution latency must be kept to a minimum (Gilyazev and Turdakov, 2018).

*Challenges addressed:* Lack of labeled data

*Limitations:* According to the practitioners, crowdsourcing has the following limitations.

- Poor quality data labels
- Lack of agility with the workforce or tools
- Need for reviewing and evaluating the quality of labels
- Crowdsourcing cannot work in the absence of metadata

Since manually validating the quality of submitted results is laborious, unethical personnel frequently take advantage of this limitation and submit low-quality responses. As a result, practitioners have stated that they require systems that can reliably evaluate worker quality, allowing for the rejection and filtering of low-performing workers and spammers. Further, when the annotators in a pool work more, gradually their efficiency decrease, affecting the quality of annotations. Consequently, more number of workers in a pool demand management, which is an overhead.

#### 5.1.2. Active learning

*Description:* One method to make the data labeling process easier is to use active learning. The algorithm takes one example from the unlabeled set for each iteration, then passes it to the oracle (expert) for labeling, and the classifier is re-trained using the new set of training data. The labeled cases are chosen by an active learning algorithm and added to the training set. A learner typically starts with a small collection of labeled cases, selects a few informative instances from a pool of unlabeled data, and queries an oracle for labels (e.g., a human annotator). The goal is to reduce the total labeling cost to train a reliable model. (Settles et al., 2008). Active learning approaches seek out the most informative examples for the classifier, resulting in a substantial reduction in the amount of labeled data, as proven both theoretically and empirically. (Settles, 2009). Further, it helps the practitioners to identify the part of the data that needs to be labeled to achieve maximum benefit. For instance, more examples near the classification boundary help more compared to the labeled instances that lay far away from the decision boundary. Active learning assumes that labeling is performed by just one expert (oracle) at any given instance. However, it is often required to parallelize the load, thus allowing the expert to label more samples. To connect businesses and employees, crowdsourcing services such as Amazon Mechanical Turk (MTurk), CrowdFlower, and Toloka are used. Any platform user can create a job, such as labeling a set of numerous samples. Another user (annotator), having found an interesting task, performs it for a certain fee, which is much lower than the cost of involving an expert.

*Challenges addressed:* Lack of labeled data

*Limitations:* Based on the inputs from the practitioners, active learning has the following limitations:-

- Low-quality labeling due to data ambiguity, poor guidelines for annotators, and lack of motivation or knowledge
- A fixed cost cannot be assumed for acquiring each label. For example, if labels are acquired by executing a biological experiment, then the cost of a query might be the price of the materials used.

#### 5.1.3. N-shot learning

*Description:* A shot is simply a single sample that can be used for training. As a result, N-shot learning has $N$ training instances. The "few" in the term "few-shot learning" normally ranges from zero to five, so zero-shot learning refers to training a model with no instances, one-shot learning refers to training a model with one example, and so on. Few-shot learning is defined as learning new concepts from only a few labeled examples (Triantafillou et al., 2017). K-shot N-way classification is the job of categorizing a data point into one of $N$ classes when only K instances of each class are available to guide the decision. This is a challenging situation that needs methods that are different from those used when there is a lot of labeled data for each new concept. Deep learning algorithms rely on large datasets and are prone to overfitting due to a lack of data. However, expecting many examples for learning a new class or concept is unrealistic and undesirable, making few-shot learning a critical issue to handle. Few-shot learning aims to get as much information out of each training batch as possible, which is especially critical when the amount of data available for learning each class is restricted (Snell et al., 2017). The ability to classify instances of an unseen visual class, called zero-shot learning (Socher et al., 2013). Zero-shot learning can be used for products or activities that lack labeled data and new visual categories, such as the latest electronics or automobile models. The goal of zero-shot learning is to recognize items that were not seen during training (Xian et al., 2017). Some common zero-shot splits may regard feature learning as a separate stage from training, necessitating the creation of additional dataset splits. Furthermore, in a zero-shot learning context, separate training, and validation class split is a crucial component of parameter adjustment. Image classification systems in real-world applications do not have advanced knowledge of whether a new image corresponds to a seen or unseen class. As a result, generalized zero-shot learning is appealing from a practical standpoint (Romera-Paredes and Torr, 2015). The model must learn a new class from a single example in one-shot learning, which is an extreme variant of few-shot learning. Models learn $N$ new classes given just M less than $N$ examples in a 'less than one-shot learning assignment, which can be accomplished with the use of soft labels (Sucholutsky and Schonlau, 2020). The essential premise of less-than-one-shot learning is that after a few categories have been learned the hard way, some information from that process can be abstracted to make learning subsequent categories more efficient. In other words, rather than starting from scratch to learn a new or unfamiliar category, make use of existing information (Fei-Fei et al., 2006). However, n-shot learning is also not free from limitations.

*Challenges addressed:* Lack of labeled data

*Limitations:* Based on practitioners experience, n-shot learning has the following limitations:

- Lack of generalization
- Hinders the classification ability of the model when the images have noise
- Not a popular technique in industry

### *5.1.4. Unsupervised learning and semi-supervised learning*

*Description:* To address the issue of annotation, we need a method that reduces the requirement for annotation, as target annotation is typically costly. Unsupervised learning (UL) is a machine learning algorithm that works with unlabeled datasets. Cluster analysis is most typically used to identify hidden patterns in huge unlabeled datasets. UL can deduce the structure of data that has not been labeled (Celebi and Aydin, 2016). However, they necessitate some human interaction when it comes to validating output variables. An unsupervised learning model, for example, can detect that online buyers frequently purchase groups of products at the same time. A data analyst would need to confirm that grouping makes sense for a recommendation engine. Feature learning settings that are unsupervised, depends on the unlabeled data. The self-taught learning to set is a more general and powerful option, as it does not require your unlabeled data to come from the same distribution as your labeled data (Triguero et al., 2015). The semi-supervised learning (SSL) paradigm has received a lot of attention in a variety of domains, from biology to Web mining, where getting unlabeled data is easier than getting labeled data since it takes less effort, expertise, and time (Zhu and Goldberg, 2009). Self-labeling approaches are used in semi-supervised algorithms. Self-labeled techniques are a viable and promising category of strategies for utilizing both types of data, and they are strongly tied to other methods and difficulties. Single-classifier methods and multi-classifier methods are two types of self-labeled approaches. In semi-supervised learning, numerous labeled instances are typically required to train a weakly effective predictor, which is then utilized to exploit the unlabeled data (Zhou et al., 2007). However, there may be very few labeled training examples in many real-world applications, which makes the weakly useful predictor difficult to generate, and therefore these semi-supervised learning methods cannot be applied. Further, they have limitations in extracting the most confident predictions from the learner (Zhai et al., 2019).

*Challenges addressed:* Lack of labeled data

*Limitations:* According to the practitioners, these learning methods have limitations, such as:

- The spectral classes do not necessarily represent the features in UL.
- UL does not consider spatial relationships in the data.
- UL can take time to interpret the spectral classes.
- Iteration results are not stable for semi-supervised learning.
- Semi-supervised learning does not apply to network-level data.
- Semi-supervised learning has low accuracy.

### *5.1.5. Self-supervised learning*

*Description:* Self-supervised learning is a broad framework for learning that relies on surrogate (pretext) tasks that can be created using solely unsupervised data. A pretext assignment is created in such a way that completing it necessitates the acquisition of a useful visual representation (Zhai et al., 2019). Self-supervised learning attempts to overcome these restrictions by learning image representations directly from pixels, rather than depending on pre-defined semantic labels (Misra and Maaten, 2020). This is frequently accomplished using a pretext task that involves applying a transformation to the input image and requiring the learner to anticipate transformation properties from the produced image. Rotations, affine transformations, and jigsaw transformations are examples of transformations. Without the use of labels, self-supervision produces effective representations for downstream tasks. These methods outperform systems that merely learn visual representations from unsupervised images. Self-supervised learning models, on the other hand, perform poorly compared to fully supervised learning models, and they are frequently not considered advantageous beyond obviating or minimizing the need for annotations (Hendrycks et al., 2019).

*Challenges addressed:* Lack of labeled data

*Limitations:* Practitioners mentioned that self-supervised learning had the following limitations:

- The reason behind its limited usage is mostly due to its novelty
- Building models can be more computationally intense
- Inaccurate labels may cause inaccurate results

### *5.1.6. Help from domain experts*

*Description:* Domain experts to label data is one of the labeling techniques proposed in the literature (Shi et al., 2008). A domain expert possesses knowledge of the domain in addition to the ability to label instances. This knowledge can be used to identify areas of the feature space that are inaccessible, missing features that are required to divide classes, or features that are not required for the task at hand (Holmberg et al., 2020). Domain experts operate in the business domain. Their environment is made up of real-world business transactions and interactions, and a major portion of the knowledge they rely on has built up organically in the form of skill and experience, which they apply in a variety of ways (Viaene, 2013). Domain experts are especially useful in domains/use cases where label quality is critical. In reality, this frequently leads to a trade-off between cost and quality: one seeks to make the most of expensive domain specialists while also maximizing the use of low-cost crowd annotations (Nguyen et al., 2015).

*Challenges addressed:* Lack of labeled data

*Limitations:* Based on the case study, we understood that this solution has the following limitations:

- Not applicable for all sets of problems. For instance, hiring a doctor to annotate the inner body details from a surgery video might become a whole day process.
- Time-consuming and expensive

### *5.1.7. Lossless data aggregation*

*Description:* To combat the challenge of the data granularity problem, one of the naive techniques is lossless data compression. Data compression is a common practice (Lin and Kolcz, 2012), especially when data from multiple sources are collected. BZIP2 based on Burrow Wheelers Transform (BWT) (Burrows and Wheeler, 1994) and GZIP based on LempelZiv (LZ) (Ziv and Lempel, 1977) are two popular and successful lossless text compression schemes widely used for the compression of data files. These techniques can be directly applied to time-series data. Plane fitting is a method used in Google for aggregating data obtained through laser range scans which accurately measure the depth, the two sides, and the front of the vehicle (Anguelov et al., 2010). Another common technique is to have a separate raw data storage where data files are stored before aggregation (Raj et al., 2020). However, the data storage space limitation makes this approach less acceptable for many small-scale companies.

*Challenges addressed:* Data granularity

*Limitations:* Practitioners that participated in the study said that lossless data aggregation has the following limitation.

- Hard to retain the original amount of data/details without loss during aggregation

### 5.1.8. Resampling and synthetic data generation

*Description:* Up-sampling and down-sampling are two methods that can be used to re-balance the class distributions, thus solving challenges with the imbalanced dataset. With synthetic data generation, artificial objects similar to the minority class are introduced into the dataset. SMOTE (Chawla et al., 2002), improved alternatives such as ADASYN (Chen et al., 2010) or RAMO (He et al., 2008)are some oversampling techniques. Downsampling is the least preferred solution, as it may remove valuable samples in the dataset. However, running too many iterations of these leads to problems such as class distribution shift (Krawczyk et al., 2012). Besides the application of data level techniques, it is possible to choose algorithms such as decision trees that can perform well with the imbalanced dataset. Furthermore, performance evaluation metrics like precision, recall, and F1-measure can be used to reduce the impact of an imbalanced dataset problem. Upsampling with data augmentation is the most used industrial practice to mitigate the challenge of imbalanced datasets.

*Challenges addressed:* Shortage of diverse samples

*Limitations:* Based on the input from the practitioners, resampling and synthetic data generation has the following limitations.

- Oversampling increases the number of training examples, thus increasing the learning time.
- Oversampling makes exact copies of existing examples, likely leading to overfitting.
- Undersampling discards potentially useful data
- While synthetic data can imitate many properties of original data, it cannot exactly copy the original content.
- Synthetic data may not often cover the corner cases like authentic data
- Synthetic data needs a verification server

### 5.1.9. Class weighting with data augmentation

*Description:* Manipulation of data, such as weighing data examples or adding new instances, is becoming more popular as a solution to the problem of an imbalanced dataset. Data augmentation, for example, expands the data size by applying label-preserving changes to original data points; class weighting assigns significant weight to each instance to modify its effect on learning, and data synthesis generates entire synthetic cases. The use of data augmentation methods for time series data is fraught with challenges. To begin with, current data augmentation approaches do not completely use the intrinsic features of time series data. The so-called temporal dependency is a distinctive attribute of time series data (Wen et al., 2020). Time-series data, unlike image data, may be converted in the frequency and time–frequency domains, allowing effective data augmentation methods to be designed and implemented in the changed domain. When modeling multivariate time series, we must include the possibly complex dynamics of various variables across time, which makes things more complicated. As a result, just using image and audio processing data augmentation methods may not result in genuine synthetic data. Second, data augmentation techniques are task-specific. For example, data augmentation strategies that are appropriate for time series classification may not be appropriate for detecting time series anomalies (Hu et al., 2019). These two techniques of manipulation perform in diverse contexts: augmentation outperforms weighing when only a small quantity of data is available, but weighting outperforms augmentation when dealing with class imbalance issues. As a result, depending on the application parameters, the type of modification changes.

*Limitations:* Based on the input from the practitioners, class weighting with data augmentation is the most effective technique to combat a shortage of diverse samples and class imbalance. They did not mention any limitations for this solution.

*Challenges addressed:* Shortage of diverse samples and class imbalance

### 5.1.10. Data pipelines

*Description:* Data pipelines are complex chains of interconnected activities that start with a data source and end in a data sink. In a data pipeline, the output of one component becomes the input to the other (Van Alstyne et al., 2016) and allows smooth, automated flow of data from source to destination. Data pipelines enable data sharing between two companies or organizations within a company. The ultimate destination of a data pipeline need not be data storage. Instead, it can be any application such as a visualization tool (Matheus et al., 2018; Stadler et al., 2016), Machine Learning(ML) models (Gautam and Yadav, 2014; Tanzil et al., 2017) or Deep Learning(DL) models (Covington et al., 2016; Deng et al., 2013). The data pipeline's components can automate the operations of extracting, processing, integrating, validating, and loading data (Sun et al., 2018). Data pipelines can process different types of data such as continuous, intermittent, and batch data (Goodhope et al., 2012). Moreover, data pipelines eliminate errors and accelerate the end-to-end data processes, which in turn reduces the latency in the development of data products. Monitoring the data pipeline activities can solve the problem of data tracking (Munappy et al., 2019).

*Challenges addressed:* Need for data sharing and tracking methods, data heterogeneity

*Limitations:* According to the practitioners, data pipelines has the following issues.

- Building data pipelines for a use case demands an unreasonable amount of time and effort
- Need to identify the activities which consume data, the output of each activity, order of execution, monitoring methods, intermediate storages, where to place the storage in the pipeline, data collection method, etc. which varies between use cases

## 5.2. Data preprocessing

Dirty data, managing sequential data and categorical data are the main challenges in the data preprocessing phase. Dirty data is an umbrella term used for a collection of problems and there exist solutions such as data scrubbing, data cleansing, and data imputation. To manage categorical data, one-hot encoding, label encoding, and embedding vectors are the major solutions. However, managing sequences in data are not explored in the literature and practitioners are struggling with this challenge, although it is not common in all the use cases.

### 5.2.1. One-hot encoding

*Description:* Scaling converts categorical data to numerical data by turning one numerical data type into another numerical data form during coding (Zhang et al., 2003). The two most common scaling methods for converting categorical data into numerical form are one-hot encoding and label encoding. When categorical data has no link between categories, a one-hot encoding is appropriate. Each category variable is represented by a binary vector with one element for each unique label, with the class label set to 1 and all other elements set to 0. One of the advantages of one-hot encoding is that the result is binary rather than ordinal, and everything is stored in an orthogonal vector space. The drawback is that with large cardinality, the feature space can quickly expand, resulting in the curse of dimensionality. One-hot encoding produces billions of trailers and billions of one-hot vectors, which is challenging to manage as categorical data grows. Furthermore, the mapping is entirely uninformed: "similar" categories in embedding space are not placed closer to each other.

*Challenges addressed:* Managing categorical data

*Limitations:* According to the practitioners, this technique has the following limitations:

- Curse of dimensionality means that the error increases with the increase in the number of features
- Need for Principal Component Analysis
- Poor Scalability
- Natural order is lost and so is the relationship between each unique category

### *5.2.2. Learned embeddings*

*Description:* A learned embedding (vectors of numbers), typically known as an "embedding", is a distributed representation for categorical data. Each category is assigned to a unique vector, and the vector's attributes are changed or learned as the neural network is trained. The vector space acts as a projection of the categories, allowing closely related categories to naturally cluster together. This has the advantages of an ordinal relationship in that any such relationship can be learned from data, as well as a one-hot encoding in that each category has a vector representation. The input vectors are not sparse, unlike one hot encoding (do not have lots of zeros). The disadvantage is that it necessitates learning as part of the model and the creation of many more input variables (columns) (Guo and Berkhahn, 2016). Embeddings not only save memory and speed up neural networks as compared to one-hot encoding, but they also show the intrinsic features of categorical variables by mapping similar values near together in the embedding space. One of the solution's disadvantages is that the original data is replaced by a similarity matrix in the first phase, and dimensionality is decreased using an embedding in the second. The computation of a distance matrix can take a long time, depending on the data and distance function. An efficiently constructed distance function may be able to mitigate this. Furthermore, some embeddings are hypersensitive to noise in data. This may be mitigated by additional data cleaning. In addition, some embeddings are sensitive to the choice of their hyperparameters. This may be mitigated by careful analysis or hyperparameter tuning.

*Limitations:* According to the input from the practitioners, learned embeddings is the most effective technique to combat the categorical data challenge.

*Challenges addressed:* Managing categorical data

### *5.2.3. Solutions to dirty data problem*

*Description:* Dirty data problem is an umbrella term that includes various data problems such as incompleteness, uniqueness, incorrectness, inconsistency, inaccuracy, Kim et al. (2003) etc. Data scrubbing is the procedure to modify or removing incomplete, incorrect, inaccurately formatted, or repeated data in a dataset to improve consistency, accuracy, and reliability. Data cleaning is a process of tidying up the data, largely involving correcting or deleting obsolete, redundant, corrupt, poorly formatted, or inconsistent data. Data professionals do the actual cleaning, checking the dataset, and making corrections and edits as needed. Data scrubbing is a subset of data cleaning. Imputation is a powerful method used to solve the missing data problem. Imputation can be based on statistical methods as well as machine learning-based methods. K. Lakshminarayan et al. Jerez et al. (2010) has applied machine learning for implementing data imputation. Statistical methods include mean hot-deck and multiple imputation (Lakshminarayan et al., 1996). Data linter (Hynes et al., 2017) is a new class of tools that automatically inspects ML data sets to identify potential issues in the data.

*Challenges addressed:* Dirty data problems such as inconsistency, incompleteness, inaccuracy, redundancy, ill-formatted data.

*Limitations:* According to the practitioners, the solutions to the dirty data challenge is still not a well-explored area and have the following limitations.

- Hard to anticipate all potential data problems in a real-world context where data changes rapidly.
- Does not preserve the relationships among variables
- Leads to an underestimation of standard errors
- Time consuming and expensive

## *5.3. Data preparation*

Data dependency and data quality are the two challenges encountered in the data preparation phase. To combat the challenge of data quality, data validation frameworks and data linter are used in practice. However, these are solutions are not applicable for image data.

### *5.3.1. Tools for testing data quality*

*Description:* Data linter (Hynes et al., 2017) is an ML-based tool used in Google for detecting lints which can be classified into three high-level categories: outliers, packaging errors and miscodings of data. Data validation at the major steps of the data processing pipeline can be a possible option to detect and catch the errors affecting the data quality. However, complete prevention of data quality problems is not possible as the data generation happens in distributed data sources where exerting control is not possible. Therefore, the only possible option is to detect the quality issues in the early stages and implement mitigation strategies to overcome the issues (Lwakatare et al., 2021). To detect the data quality issues, test cases should be written and executed, which is an expensive task.

*Challenges addressed:* Data quality

*Limitations:* According to the practitioners, the data quality testing tools have the following limitations.

- Incompatible with image data
- Impossible to predict all potential problems with data
- Expensive and time-consuming

## *5.4. Deployment*

The deployment phase faces challenges like data extraction methods and overfitting. Overfitting can be solved using regularization, data augmentation cross-validation, model simplification, transfer learning, and dropouts.

### *5.4.1. Regularization and cross-validation*

*Description:* Regularization is a strategy for improving model generalization by making minor changes to the learning procedure. As a result, the model's performance on previously unseen data improves as well. Google shows that L1 regularization can increase performance for a few kernels, but degrade performance in larger-scale instances. L2 regularization, on the other hand, never affects performance and improves it significantly with many kernels (Cortes et al., 2012). Zhang et al. (2018) and Cliche (2017) describes the application of regularizer in Facebook and Bloomberg, respectively. DeCov is a specific strategy used in Microsoft and Facebook to avoid the overfitting of their ML/DL models (Cogswell et al., 2015). Cross-Validation is a powerful preventative measure against overfitting (Whittaker et al., 2010). Google, Facebook, Uber, and many more large-scale enterprises employ K-fold cross-validation as their most widely used techniques. Cross-validation permits hyperparameter adjustment with the sole original training set, while keeping the test set as a completely unknown dataset for final model selection. Regularization, on the other hand, aims to lower the estimator's variance by simplifying it, which increases the bias and reduces the anticipated error. When cross-validation is used, the processing power required is also considerable. As a result, in the case of huge data sets, the time it takes to acquire feedback on the model's performance increases.

*Challenges addressed:* Overfitting of DL models

*Limitations:* According to the practitioners, regularization and cross-validation has the following limitations:

- Curse of dimensionality, which means that the error increases proportional to the increase in number of features.
- Cross Validation is computationally costly, requiring a lot of processing power.
- Cross-Validation drastically increases the training time

### 5.4.2. Model simplification

*Description:* When dealing with overfitting, the first step is to reduce the model's complexity. We can lower the network's complexity by simply removing layers or reducing the number of neurons (Fahland and Van Der Aalst, 2013). While doing this, it is critical to calculate the input and output dimensions of the various layers involved in the neural network. Without a general rule on the size of the neural network, re-architecture is difficult.

*Challenges addressed:* Model simplification is used to address the overfitting challenge

*Limitations:* According to the practitioners, this solution has the following limitations:

- No guidelines on how to perform model simplification
- Expertise on Neural Network architecture is required

### 5.4.3. Dropout

*Description:* Regularization approaches like L1 and L2 change the cost function to reduce overfitting. Dropout, on the other hand, affects the network as a whole. It removes neurons from the neural network at random throughout each iteration of training (Srivastava et al., 2014). When different sets of neurons are dropped, it is equivalent to training different neural networks. The different networks will overfit in different ways, so the overall effect of dropout will be to reduce overfitting. Based on the case study for validation, dropout can sometimes decrease the performance of the model.

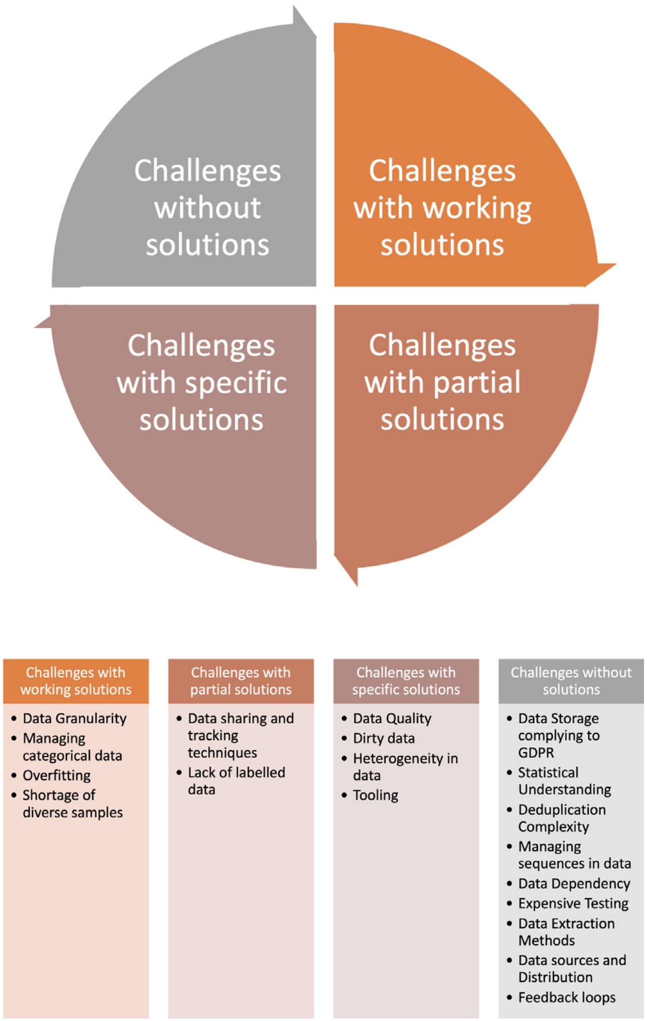
*Challenges addressed:* Dropout is a regularization technique that prevents neural networks from overfitting.

*Limitations:* According to the practitioners, dropout has the following limitations

- Dropout can hurt the model performance when training time is limited or when the neural network is small relative to the dataset

## 6. Summarizing challenges and solutions

We conducted a multiple case study to validate the industrial applicability of the identified solutions. Based on the limitations of the existing solutions and availability, we have classified the challenges into four, as shown in Fig. 4. Challenges with working solutions is a category of challenges that needs improvements in solutions in terms of time and cost. The second category is the challenges with partial solutions, where the solutions can be used to solve a part of the problem. For example, the lack of labeled data can be solved by using crowdsourcing, active learning, N-short learning, etc. However, if the metadata itself is missing, it cannot be solved by using any of the listed techniques. Challenges with specific solutions is a third category where the existing solutions are not applicable for all use cases. i.e. the solutions are applicable only for specific DL use cases. For instance, a data linter and data validation framework can solve the problem of data quality. However, they are not applicable for the use cases that use image datasets like Melanoma detection. Finally, the fourth category of challenges contributes to the open research challenges, as the listed challenges have currently no available solutions.


Challenges without solutions
Challenges with working solutions
Challenges with specific solutions
Challenges with partial solutions

| Challenges with working solutions | Challenges with partial solutions | Challenges with specific solutions | Challenges without solutions |
|---|---|---|---|
| • Data Granularity<br>• Managing categorical data<br>• Overfitting<br>• Shortage of diverse samples | • Data sharing and tracking techniques<br>• Lack of labelled data | • Data Quality<br>• Dirty data<br>• Heterogeneity in data<br>• Tooling | • Data Storage complying to GDPR<br>• Statistical Understanding<br>• Deduplication Complexity<br>• Managing sequences in data<br>• Data Dependency<br>• Expensive Testing<br>• Data Extraction Methods<br>• Data sources and Distribution<br>• Feedback loops |

**Fig. 4.** Classification of challenges based on the availability of solutions.

### 6.1. Challenges with working solutions

Data granularity is a challenge for which we have identified solutions such as lossless data aggregation techniques and synthetic data. Practitioners mentioned that they need more efficient solutions in terms of performance, as the existing solutions often affect the performance of the model. Managing categorical data is also a challenge for which learned embeddings are an effective solution. Practitioners did not mention any limitations for the technique. Similarly, overfitting is another challenge in this category, for which there exist a variety of solutions and regularization is the most widely used technique by the practitioners. The shortage of diverse samples can be solved using class weighting with data augmentation.

### 6.2. Challenges with partial solutions

The need for data sharing and tracking methods and the lack of labeled data are the challenges that fall in this category. Lack of labeled data can be addressed using crowdsourcing, active labeling, etc. However, if the metadata is missing, it is often not possible to use any of the existing solution approaches. Similarly, data sharing and tracking can be partly solved by data pipelines. However, implementing data pipelines for each use case itself is identified as an expensive task.

### 6.3. Challenges with specific solutions

Data quality, dirty data, data heterogeneity, tooling are the challenges for which the solutions are not general. The tooling problem is to an extent solved for use cases that can avail cloud services. Similarly, data heterogeneity can be solved by defining data ingestion modules in the data pipelines. However, the practical difficulties of establishing secure data pipelines between two parties make this solution not applicable for a set of use cases like wind power prediction.

### 6.4. Challenges without solutions

Challenges in this category are more compared to the other three categories. None of these challenges has any available working solutions. Therefore, this category needs much attention.

The results of our study confirms the data collection and data quality challenges described by Whang et al. However, our study is a more extensive version with more challenges identified and categorized according to the data lifecycle stages. Moreover, our study identifies the solutions for the identified challenges and explains the reason for listing those challenges even with the existence of solutions in the literature. We also try to identify the least explored areas, such as data testing challenge, solutions to feedback loop challenge, statistical understanding etc. Another closely related study is software engineering challenges for deep learning, where the authors describe and categorize the challenges into development, production, and organizational challenges. Some challenges such as unintended feedback loops, monitoring and logging are also described in the study (Arpteg et al., 2018).

## 7. Conclusion and research implications

Data management challenges are an important part of the DL model development. However, the current body of literature neither discuss the challenges nor the solutions. The inability to formalize the solution for data management leads to a need for manual enforcement of them. The research presented here shows that this is an error-prone and time-consuming task that takes most of the effort of the data scientists and other practitioners working with data during the construction-intensive phases of a project. This problem exists in traditional Machine Learning-based development, as well, but it is more apparent and acute in DL. Because Deep neural networks has been able to learn the minute details from the input data. However, industries applying DL have not been able to completely utilize the potential of DL due to several reasons, and the inability to solve the data management challenges is predominant among them. The cases presented in this study shows that the data management, is a bottleneck in DL projects and demands a lot of human intervention which leads to operational errors, training-serving skew and performance degradation. This paper presents 20 data management challenges for DL and their categorization according to the data lifecycle phase. Further, we have identified solutions for these challenges through a systematic literature review. We have also conducted a second round of interview with the same practitioners to understand the reasons for not using these solutions in practice. Based on the availability of solutions, we further categorized the challenges into four, namely challenges with working solutions, challenges with partial solutions, challenges with specific solutions and challenges without solutions. We believe that results of this study can be used by researchers to identify and solve the challenges that are unsolved, partially solved and specifically solved. Further, it can help the practitioners to explore the solutions that are not familiar, such as N-shot learning. The implications for researchers are that there is a need for further research to find solutions in such a way that they are both amenable to automatic enforcement on the detailed design and easy to understand and use by both data scientists and developers and work in practice. The implications for practitioners are that there are solutions in the literature which remains totally unexplored, and this needs to be taken into account during the development of DL systems. Although the data management is inherently a task for a relatively small group, it should be possible to delegate the existing solutions to a larger group. This would give time for the data scientists to concentrate on the core tasks: data exploring, learn new tools, and maintaining the DL systems.

## CRediT authorship contribution statement

**Aiswarya Raj Munappy:** Conceptualization, Methodology, Validation/Verification, Investigation, Writing. **Jan Bosch:** Project administration, Validation/Verification, Investigation (partly). **Helena Holmström Olsson:** Project administration, Validation/Verification, Methodology. **Anders Arpteg:** Project administration, Validation/Verification. **Björn Brinne:** Project administration, Validation/Verification.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

This work is in part supported by Vinnova, Sweden and by the Software Center. The authors would also like to express their gratitude for all the support provided by Peltarion.

## References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., et al., 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467.

Anguelov, D., Dulong, C., Filip, D., Frueh, C., Lafon, S., Lyon, R., Ogale, A., Vincent, L., Weaver, J., 2010. Google street view: Capturing the world at street level. Computer 43 (6), 32–38.

Arpteg, A., Brinne, B., Crnkovic-Friis, L., Bosch, J., 2018. Software engineering challenges of deep learning. In: 2018 44th Euromicro Conference on Software Engineering and Advanced Applications. SEAA, IEEE, pp. 50–59.

Bahdanau, D., Cho, K., Bengio, Y., 2014. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

Baxter, P., Jack, S., et al., 2008. Qualitative case study methodology: Study design and implementation for novice researchers. Qual. Rep. 13 (4), 544–559.

Beaufays, F., 2015. Google AI blog: The neural networks behind google voice transcription. https://ai.googleblog.com/2015/08/the-neural-networks-behind-google-voice.html. (Accessed 05 January 2020).

Bengio, S., 2000. Modeling high-dimensional discrete data with. In: Advances in Neural Information Processing Systems 12: Proceedings of the 1999 Conference, Vol. 1. MIT Press, p. 400.

Bickman, L., Rog, D.J., 2008. The SAGE Handbook of Applied Social Research Methods. Sage publications.

Burrows, M., Wheeler, D.J., 1994. A block-sorting lossless data compression algorithm. Digit. SRC Res. Rep..

Celebi, M.E., Aydin, K., 2016. Unsupervised Learning Algorithms. Springer.

Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P., 2002. SMOTE: Synthetic minority over-sampling technique. J. Artificial Intelligence Res. 16, 321–357.

Chen, S., He, H., Garcia, E.A., 2010. RAMOBoost: Ranked minority oversampling in boosting. IEEE Trans. Neural Netw. 21 (10), 1624–1642.

Chen, X.-W., Lin, X., 2014. Big data deep learning: Challenges and perspectives. IEEE Access 2, 514–525.

Cliche, M., 2017. Bb_twtr at semeval-2017 task 4: Twitter sentiment analysis with cnns and lstms. arXiv preprint arXiv:1704.06125.

Cogswell, M., Ahmed, F., Girshick, R., Zitnick, L., Batra, D., 2015. Reducing overfitting in deep networks by decorrelating representations. arXiv preprint arXiv:1511.06068.

Cortes, C., Mohri, M., Rostamizadeh, A., 2012. L2 regularization for learning kernels. arXiv preprint arXiv:1205.2653.

Covington, P., Adams, J., Sargin, E., 2016. Deep neural networks for youtube recommendations. In: Proceedings of the 10th ACM Conference on Recommender Systems. pp. 191–198.

Daily, M., Medasani, S., Behringer, R., Trivedi, M., 2017. Self-driving cars. Computer 50 (12), 18–23.

Deng, L., Li, J., Huang, J.-T., Yao, K., Yu, D., Seide, F., Seltzer, M., Zweig, G., He, X., Williams, J., et al., 2013. Recent advances in deep learning for speech research at microsoft. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, pp. 8604–8608.

Deng, L., Yu, D., et al., 2014. Deep learning: Methods and applications. Found. Trends® Signal Process. 7 (3–4), 197–387.

Dong, X.L., Rekatsinas, T., 2018. Data integration and machine learning: A natural synergy. In: Proceedings of the 2018 International Conference on Management of Data. pp. 1645–1650.

Eisenhardt, K.M., 1989. Building theories from case study research. Acad. Manag. Rev. 14 (4), 532–550.

Fahland, D., Van Der Aalst, W.M., 2013. Simplifying discovered process models in a controlled manner. Inf. Syst. 38 (4), 585–605.

Fei-Fei, L., Fergus, R., Perona, P., 2006. One-shot learning of object categories. IEEE Trans. Pattern Anal. Mach. Intell. 28 (4), 594–611.

Gautam, G., Yadav, D., 2014. Sentiment analysis of twitter data using machine learning approaches and semantic analysis. In: 2014 Seventh International Conference on Contemporary Computing. IC3, IEEE, pp. 437–442.

Gilyazev, R., Turdakov, D.Y., 2018. Active learning and crowdsourcing: A survey of optimization methods for data labeling. Program. Comput. Softw. 44 (6), 476–491.

Goodhope, K., Koshy, J., Kreps, J., Narkhede, N., Park, R., Rao, J., Ye, V.Y., 2012. Building linkedin's real-time activity data pipeline. IEEE Data Eng. Bull. 35 (2), 33–45.

Gruener, R., Cheng, O., Litvin, Y., 2018. Introducing petastorm: Uber ATG's data access library for deep learning | uber engineering blog. https://eng.uber.com/petastorm/. (Accessed 05 January 2020).

Guo, C., Berkhahn, F., 2016. Entity embeddings of categorical variables. arXiv preprint arXiv:1604.06737.

Halevy, A., Korn, F., Noy, N.F., Olston, C., Polyzotis, N., Roy, S., Whang, S.E., 2016. Goods: Organizing google's datasets. In: Proceedings of the 2016 International Conference on Management of Data. pp. 795–806.

He, H., Bai, Y., Garcia, E.A., Li, S., 2008. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence). IEEE, pp. 1322–1328.

Hendrycks, D., Mazeika, M., Kadavath, S., Song, D., 2019. Using self-supervised learning can improve model robustness and uncertainty. arXiv preprint arXiv:1906.12340.

Holmberg, L., Davidsson, P., Linde, P., 2020. A feature space focus in machine teaching. In: 2020 IEEE International Conference on Pervasive Computing and Communications Workshops. PerCom Workshops, IEEE, pp. 1–2.

Hu, Z., Tan, B., Salakhutdinov, R., Mitchell, T., Xing, E.P., 2019. Learning data manipulation for augmentation and weighting. arXiv preprint arXiv:1910.12795.

Hynes, N., Sculley, D., Terry, M., 2017. The data linter: Lightweight, automated sanity checking for ml data sets. In: NIPS MLSys Workshop. pp. 1–7.

Jerez, J.M., Molina, I., García-Laencina, P.J., Alba, E., Ribelles, N., Martín, M., Franco, L., 2010. Missing data imputation using statistical and machine learning methods in a real breast cancer problem. Artif. Intell. Med. 50 (2), 105–115.

Jones, N., 2014. Computer science: The learning machines. Nat. News 505 (7482), 146.

Kahng, M., Andrews, P.Y., Kalro, A., Chau, D.H.P., 2017. A cti v is: Visual exploration of industry-scale deep neural network models. IEEE Trans. Vis. Comput. Graphics 24 (1), 88–97.

Kepuska, V., Bohouta, G., 2018. Next-generation of virtual personal assistants (microsoft cortana, apple siri, amazon alexa and google home). In: 2018 IEEE 8th Annual Computing and Communication Workshop and Conference. CCWC, IEEE, pp. 99–103.

Kim, W., Choi, B.-J., Hong, E.-K., Kim, S.-K., Lee, D., 2003. A taxonomy of dirty data. Data Min. Knowl. Discov. 7 (1), 81–99.

Kitchenham, B., 2004. Procedures for performing systematic reviews. Keele, UK, Keele University.

Krawczyk, B., Schaefer, G., Wozniak, M., 2012. Breast thermogram analysis using a cost-sensitive multiple classifier system. In: Proceedings of 2012 IEEE-EMBS International Conference on Biomedical and Health Informatics. IEEE, pp. 507–510.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems. pp. 1097–1105.

Labrinidis, A., Jagadish, H.V., 2012. Challenges and opportunities with big data. Proc. VLDB Endow. 5 (12), 2032–2033.

Lakshminarayan, K., Harp, S.A., Goldman, R.P., Samad, T., et al., 1996. Imputation of missing data using machine learning techniques.. In: KDD. pp. 140–145.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521 (7553), 436–444.

LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D., 1989. Backpropagation applied to handwritten zip code recognition. Neural Comput. 1 (4), 541–551.

Lin, J., Kolcz, A., 2012. Large-scale machine learning at twitter. In: Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data. pp. 793–804.

Litjens, G., Kooi, T., Bejnordi, B.E., Setio, A.A.A., Ciompi, F., Ghafoorian, M., Van Der Laak, J.A., Van Ginneken, B., Sánchez, C.I., 2017. A survey on deep learning in medical image analysis. Med. Image Anal. 42, 60–88.

Lwakatare, L.E., Rånge, E., Crnkovic, I., Bosch, J., 2021. On the experiences of adopting automated data validation in an industrial machine learning project. In: 2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Practice. ICSE-SEIP, IEEE, pp. 248–257.

Maguire, M., Delahunt, B., 2017. Doing a thematic analysis: A practical, step-by-step guide for learning and teaching scholars.. All Ireland J. Higher Educ. 9 (3).

Malone, T.W., Laubacher, R., Dellarocas, C., 2009. Harnessing crowds: Mapping the genome of collective intelligence.

Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., Hung Byers, A., et al., 2011. Big Data: The Next Frontier for Innovation, Competition, and Productivity. McKinsey Global Institute.

Matheus, R., Janssen, M., Maheshwari, D., 2018. Data science empowering the public: Data-driven dashboards for transparent and accountable decision-making in smart cities. Gov. Inf. Q. 101284.

Meredith, J., 1998. Building operations management theory through case and field research. J. Oper. Manage. 16 (4), 441–454.

Misra, I., Maaten, L.v.d., 2020. Self-supervised learning of pretext-invariant representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6707–6717.

Munappy, A., Bosch, J., Olsson, H.H., Arpteg, A., Brinne, B., 2019. Data management challenges for deep learning. In: 2019 45th Euromicro Conference on Software Engineering and Advanced Applications. SEAA, IEEE, pp. 140–147.

Munappy, A.R., Mattos, D.I., Bosch, J., Olsson, H.H., Dakkak, A., 2020. From ad-hoc data analytics to dataops. In: Proceedings of the International Conference on Software and System Processes. pp. 165–174.

Najafabadi, M.M., Villanustre, F., Khoshgoftaar, T.M., Seliya, N., Wald, R., Muharemagic, E., 2015. Deep learning applications and challenges in big data analytics. J. Big Data 2 (1), 1.

National Security Agency, 2013. The National Security Agency: Missions, Authorities, Oversight and Partnerships. National Security Agency Ft. Meade, MD.

Nguyen, A., Wallace, B., Lease, M., 2015. Combining crowd and expert labels using decision theoretic active learning. In: Proceedings of the AAAI Conference on Human Computation and Crowdsourcing, Vol. 3, no. 1.

Nowak, S., Rüger, S., 2010. How reliable are annotations via crowdsourcing: A study about inter-annotator agreement for multi-label image annotation. In: Proceedings of the International Conference on Multimedia Information Retrieval. pp. 557–566.

Oguntimilehin, A., Ademola, E., 2014. A review of big data management, benefits and challenges. Rev. Big Data Manag., Benefits Chall. 5 (6), 1–7.

Raj, A., Bosch, J., Olsson, H.H., Wang, T.J., 2020. Modelling data pipelines. In: 2020 46th Euromicro Conference on Software Engineering and Advanced Applications. SEAA, IEEE, pp. 13–20.

Ranzato, M., Boureau, Y.-L., LeCun, Y., et al., 2007. Sparse feature learning for deep belief networks. Adv. Neural Inf. Process. Syst. 20, 1185–1192.

Rao, Q., Frtunikj, J., 2018. Deep learning for self-driving cars: Chances and challenges. In: Proceedings of the 1st International Workshop on Software Engineering for AI in Autonomous Systems. pp. 35–38.

Rastogi, R., 2018. Machine learning@ amazon. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. pp. 1337–1338.
Romera-Paredes, B., Torr, P., 2015. An embarrassingly simple approach to zero-shot learning. In: International Conference on Machine Learning. PMLR, pp. 2152–2161.
Roy, A., Sun, J., Mahoney, R., Alonzi, L., Adams, S., Beling, P., 2018. Deep learning detecting fraud in credit card transactions. In: 2018 Systems and Information Engineering Design Symposium. SIEDS, IEEE, pp. 129–134.
Runeson, P., Höst, M., 2009. Guidelines for conducting and reporting case study research in software engineering. Empir. Softw. Eng. 14 (2), 131–164.
Sergeev, A., Del Balso, M., 2018. Horovod: Fast and easy distributed deep learning in TensorFlow. arXiv preprint arXiv:1802.05799.
Settles, B., 2009. Active learning literature survey.
Settles, B., Craven, M., Friedland, L., 2008. Active learning with real annotation costs. In: Proceedings of the NIPS Workshop on Cost-Sensitive Learning, Vol. 1. Vancouver, CA.
Shi, X., Fan, W., Ren, J., 2008. Actively transfer domain knowledge. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, pp. 342–357.
Snell, J., Swersky, K., Zemel, R.S., 2017. Prototypical networks for few-shot learning. arXiv preprint arXiv:1703.05175.
Socher, R., Ganjoo, M., Sridhar, H., Bastani, O., Manning, C.D., Ng, A.Y., 2013. Zero-shot learning through cross-modal transfer. arXiv preprint arXiv:1301.3666.
Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: A simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15 (1), 1929–1958.
Stadler, J.G., Donlon, K., Siewert, J.D., Franken, T., Lewis, N.E., 2016. Improving the efficiency and ease of healthcare analysis through use of data visualization dashboards. Big Data 4 (2), 129–135.
Stuart, I., McCutcheon, D., Handfield, R., McLachlin, R., Samson, D., 2002. Effective case research in operations management: A process perspective. J. Oper. Manage. 20 (5), 419–433.
Sucholutsky, I., Schonlau, M., 2020. 'Less than one'-shot learning: Learning n classes from m< n samples. arXiv preprint arXiv:2009.08449.
Sun, H., Hu, S., McIntosh, S., Cao, Y., 2018. Big data trip classification on the new york city taxi and uber sensor network. J. Internet Technol. 19 (2), 591–598.
Tanzil, S.M.S., Hoiles, W., Krishnamurthy, V., 2017. Adaptive scheme for caching YouTube content in a cellular network: Machine learning approach. IEEE Access 5, 5870–5881.
Tole, A.A., et al., 2013. Big data challenges. Database Syst. J. 4 (3), 31–40.
Triantafillou, E., Zemel, R., Urtasun, R., 2017. Few-shot learning through an information retrieval lens. arXiv preprint arXiv:1707.02610.
Triguero, I., García, S., Herrera, F., 2015. Self-labeled techniques for semi-supervised learning: Taxonomy, software and empirical study. Knowl. Inf. Syst. 42 (2), 245–284.
Tur, G., De Mori, R., 2011. Spoken Language Understanding: Systems for Extracting Semantic Information from Speech. John Wiley & Sons.
Van Alstyne, M.W., Parker, G.G., Choudary, S.P., 2016. Pipelines, platforms, and the new rules of strategy. Harv. Bus. Rev. 94 (4), 54–62.
Verner, J.M., Sampson, J., Tosic, V., Bakar, N.A., Kitchenham, B.A., 2009. Guidelines for industrially-based multiple case studies in software engineering. In: 2009 Third International Conference on Research Challenges in Information Science. IEEE, pp. 313–324.
Viaene, S., 2013. Data scientists aren't domain experts. IT Prof. 15 (6), 12–17.
Wang, W., Zhang, M., Chen, G., Jagadish, H., Ooi, B.C., Tan, K.-L., 2016. Database meets deep learning: Challenges and opportunities. ACM SIGMOD Rec. 45 (2), 17–22.
Wen, Q., Sun, L., Yang, F., Song, X., Gao, J., Wang, X., Xu, H., 2020. Time series data augmentation for deep learning: A survey. arXiv preprint arXiv:2002.12478.
Whang, S.E., Lee, J.-G., 2020. Data collection and quality challenges for deep learning. Proc. VLDB Endow. 13 (12), 3429–3432.
Whittaker, C., Ryner, B., Nazif, M., 2010. Large-scale automatic classification of phishing pages. NDSS '10.
Xian, Y., Schiele, B., Akata, Z., 2017. Zero-shot learning-the good, the bad and the ugly. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4582–4591.
Yin, R.K., 2003. Case study research design and methods third edition. Appl. Soc. Res. Methods Ser. 5.
Yin, R.K., 2013. Validity and generalization in future case study evaluations. Evaluation 19 (3), 321–332.
Zhai, X., Oliver, A., Kolesnikov, A., Beyer, L., 2019. S4l: Self-supervised semi-supervised learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1476–1485.
Zhang, A., Ballas, N., Pineau, J., 2018. A dissection of overfitting and generalization in continuous reinforcement learning. arXiv preprint arXiv:1806.07937.
Zhang, C., Kumar, A., Ré, C., 2016. Materialization optimizations for feature selection workloads. ACM Trans. Database Syst. 41 (1), 1–32.
Zhang, S., Yao, L., Sun, A., Tay, Y., 2019. Deep learning based recommender system: A survey and new perspectives. ACM Comput. Surv. 52 (1), 1–38.
Zhang, S., Zhang, C., Yang, Q., 2003. Data preparation for data mining. Appl. Artif. Intell. 17 (5–6), 375–381.
Zhou, Z.-H., Zhan, D.-C., Yang, Q., 2007. Semi-supervised learning with very few labeled training examples. In: AAAI, Vol. 675680.
Zhu, X., Goldberg, A.B., 2009. Introduction to semi-supervised learning. Synth. Lect. Artif. Intell. Mach. Learn. 3 (1), 1–130.
Ziv, J., Lempel, A., 1977. A universal algorithm for sequential data compression. IEEE Trans. Inform. Theory 23 (3), 337–343.

**Aiswarya Raj Munappy** received the Licentiate degree in software engineering from the Chalmers University of Technology, Sweden, in 2021. She currently works as a Ph.D. student at the Chalmers University of Technology, Department of Computer Science. Her current research interests include the AI engineering, data management, deep learning, data pipelines and DataOps.

**Jan Bosch** is a Professor in the Software Engineering at the UniversityofGroningen and at ChalmersUniversityofTechnology. He received his M.Sc. in computer science in 1991 from the UniversityofTwente, and in 1995 his Ph.D. degree in computer science from LundUniversity. He has pioneered research in software architecture, digitalization and AI Engineering, and has published more than 500 research articles. Jan Bosch is the director of Software Center organization which aim to develop methods and tools for enhancing the productivity of collaborating organizations in various development phases.

**Helena Holmström Olsson** is a professor at the Department of Computer Science and Media Technology at Malmö University, Sweden. She received her Ph.D. from University of Gothenburg in 2004. She is a senior researcher in Software Center. Her research interest is in Software Engineering focused on AI engineering, data-driven development, software and business ecosystems, business agility and the overall digital transformation of the embedded systems domain.

**Anders Arpteg** has been working with AI for 20 years in both academia and industry and holds a Ph.D. in AI from Linköping University. Previously, he headed up a research group at Spotify and has headed up the research team at Peltarion. Anders is a member of the AI Innovation of Sweden steering committee, an AI adviser for the Swedish government, a member of the Swedish AI Agenda, Chairman of the Machine Learning Stockholm meetup group and a member of several other advisory boards.

**Björn Brinne** is an experienced Data Science leader with a demonstrated history of working in the computer software industry. He is skilled in data science, machine learning, and analytics. He is a strong engineering professional with a PhD in theoretical physics from Stockholm University and has contributed to many research papers across a range of academic fields, including computer science, string theory and computational biology. Björn Brinne has over a decade of experience working in data science at companies such as Truecaller, King and Electronic Arts.