machine learning & knowledge extraction

MDPI

Review

AI System Engineering—Key Challenges and Lessons Learned †

Lukas Fischer [1,*], Lisa Ehrlinger [1,2], Verena Geist [1], Rudolf Ramler [1], Florian Sobiezky [1], Werner Zellinger [1], David Brunner [1], Mohit Kumar [1] and Bernhard Moser [1]

[1] Software Competence Center Hagenberg GmbH (SCCH), 4232 Hagenberg, Austria; lisa.ehrlinger@scch.at (L.E.); verena.geist@scch.at (V.G.); rudolf.ramler@scch.at (R.R.); florian.sobieczky@scch.at (F.S.); werner.zellinger@scch.at (W.Z.); david.brunner@scch.at (D.B.); mohit.kumar@scch.at (M.K.); bernhard.moser@scch.at (B.M.)

[2] Institute for Application-Oriented Knowledge Processing, Johannes Kepler University, 4040 Linz, Austria

* Correspondence: lukas.fischer@scch.at; Tel.: +43-50-343-828

† This paper is an extended verision of our paper published in CD-MAKE Conference.

Abstract: The main challenges are discussed together with the lessons learned from past and ongoing research along the development cycle of machine learning systems. This will be done by taking into account intrinsic conditions of nowadays deep learning models, data and software quality issues and human-centered artificial intelligence (AI) postulates, including confidentiality and ethical aspects. The analysis outlines a fundamental theory-practice gap which superimposes the challenges of AI system engineering at the level of data quality assurance, model building, software engineering and deployment. The aim of this paper is to pinpoint research topics to explore approaches to address these challenges.

Keywords: AI system engineering; deep learning; embedded AI; federated learning; transfer learning; human centered AI

Citation: Fischer, L.; Ehrlinger, L.; Geist, V.; Ramler, R.; Sobieczky, F.; Zellinger, W.; Brunner, D.; Kumar, M.; Moser, B. AI System Engineering: Key Challenges and Lessons Learned. *Mach. Learn. Knowl. Extr.* **2021**, *3*, 56–83. https://dx.doi.org/10.3390/make3010004

Received: 3 December 2020
Accepted: 25 December 2020
Published: 31 December 2020

1. Introduction

Many real-world tasks are characterized by uncertainties and probabilistic data that is hard to understand and hard to process for humans. Machine learning (ML) and knowledge extraction [1] help turning this data into useful information for realizing a wide spectrum of applications such as image recognition, scene understanding, decision-support systems, and so forth, that enable new use cases across a broad range of domains.

The success of various machine learning methods, in particular Deep Neural Networks (DNNs), for challenging problems of computer vision and pattern recognition, has led to a Cambrian explosion in the field of Artificial Intelligence (AI). In many application areas, AI researchers have turned to deep learning as the solution of choice [2,3]. A characteristic of this development is the acceleration of progress in AI over the last decade, which has led to AI systems that are strong enough to raise serious ethical and societal acceptance questions. Another characteristic of this development is the way how such systems are engineered.

Above all, there is an increasing interconnection of traditionally separate disciplines such as data analysis, model building and software engineering. As outlined in Figure 1 AI system engineering encompasses all steps of building AI systems, from problem understanding, problem specification, AI model selection, data acquisition and data conditioning to deployment on target platforms and application environments.

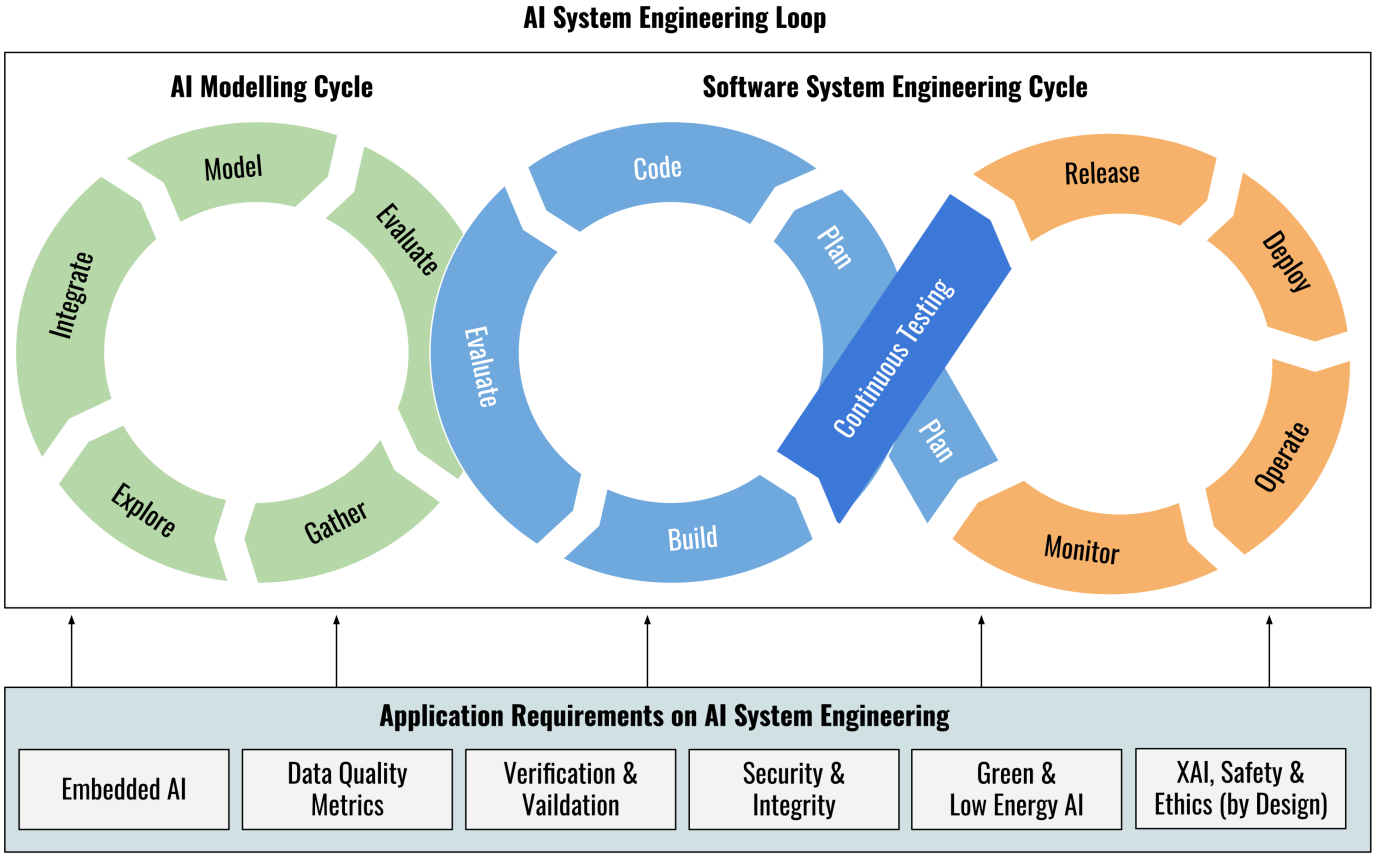

Figure 1. Artificial Intelligence (AI) System Engineering Lifecyle comprised of AI modelling cycle and software system development loop.

In particular, data-driven AI methods such as DNNs allow data to shape models and software systems that operate them. Hurdels, respectively, challenges for engineering AI systems can be split into the following three categories:

- **Hurdles from Current Machine Learning Paradigms**, see Section 2. These modelling and system development steps are made much more challenging by hurdles resulting from current machine learning paradigms. Such hurdles result from limitations of nowadays theoretical foundations in statistical learning theory and peculiarities or shortcomings of today's deep learning methods.
  - Theory-practice gap in machine learning with impact on reproducibilty and stability;
  - Lack of uniqueness of internal configuration of deep learning models with impact on reproducibility, transparancy and interpretability;
  - Lack of confidence measure of deep learning models with impact on trustworthiness and interpretability;
  - Lack of control of high-dimensionality effects of deep learning model with impact on stability, integrity and interpretability.
- **Key Challenges of AI Model Lifecycle**, see Section 3. The development of data-driven AI models and software systems therefore faces novel challenges at all stages of the AI model and AI system lifecycle, which arise along transforming data to learning models in the design and training phase, particularly
  - Data challenge to fuel the learning models with sufficiently representative data or to otherwise compensate for their lack, as for example by means of data conditioning techniques like data augmentation;

  - Information fusion challenge to incorporate constraints or knowledge available in different knowledge representation;
  - Model integrity and stability challenge due to unstable performance profiles triggered by small variations in the implementation or input data (adversarial noise);
  - Security and confidentiality to shield machine learning driven systems from espionage or adversarial interventions;
  - Interpretability and transparancy challenge to decode the ambiguities of hidden implicit knowledge representation of distributed neural parametrization;
  - Trust challenge to consider ethical aspects as a matter of principle, for example, to ensure correct behavior even in case of a possible malfunction or failure.
- **Key Challenges of AI System Lifecycle**, see Section 4. Once a proof of concept of a data-driven solution to a machine learning problem has been tackled by means of sufficient data and appropriate learning models, requirements beyond the proper machine learning performance criteria have to be taken into account to come up with a software system for a target computational platform intended to operate in a target operational environment. Key challenges arise from application specific requirements:
  - Deployment challenge and computational resource constraints, for example, on embedded systems or edge hardware;
  - Data and software quality;
  - Model validation and system verification including testing, debugging and documentation, for example, certification and regulation challenges resulting from highly regulated target domains such as in a bio-medical laboratory setting.

Outline and Structure

The outline of this paper follows the structure of our previous conference paper [4], which is now refined and extended by further use cases, details and diagrams. The paper is intended as a "Lessons learned" paper where we reflect on our experiences of past and ongoing research and development projects of machine learning systems for customers and research partners in such diverse fields as manufacturing, chemical industry, healthcare and mobility. In contrast to recent survey papers with focus on challenges of deploying machine learning systems [5], we also outline in-progress approaches from on-going research projects. the paper is composed in to an analysis section and a section of illustrative examples to demonstrate emerging approaches from selected ongoing research projects. The analysis part consists of three building blocks according to Figure 1: Block 1—Application Requirements raising hurdles (see Section 2), Block 2—AI Modeling Cycle (see Section 3) and Block 3—Software System Engineering Cycle (see Section 4). Selected approaches based on ongoing research are given in Section 5. An overview of the structure is given in the following:

- Overview of challenges and analysis
  (1) Application Requirements raising hurdles (see Section 2)
  (2) AI Modeling Cycle (see Section 3)
  (3) Software System Engineering Cycle (see Section 4)
- Outline of approaches from selected ongoing research projects
  (1) Automated and Continuous Data Quality Assurance (see Section 5.1)
  (2) Domain Adaptation Approach for Tackling Deviating Data Characteristics at Training and Test Time (see Section 5.2)
  (3) Hybrid Model Design for Improving Model Accuracy (see Section 5.3)
  (4) Interpretability by Correction Model Approach (see Section 5.4)
  (5) Software Quality by Automated Code Analysis and Documentation Generation (see Section 5.5)
  (6) the ALOHA Toolchain for Embedded Platforms (see Section 5.6)
  (7) Confidentiality-Preserving Transfer Learning (see Section 5.7)
  (8) Human AI Teaming as Key to Human Centered AI (see Section 5.8)

2. Hurdles from Current Machine Learning Paradigms

There are peculiarities of deep learning methods that affect the correct interpretation of the system's output and the transparency of the system's configuration.

2.1. Theory-Practice Gap in Machine Learning

The design and test principles of machine learning are underpinned by statistical learning theory and its fundamental theorems such as Vapnik's theorem [6]. The theoretical analysis relies on idealized assumptions such as that the data is drawn independent and identically distributed from the same probability distribution. As outlined in Reference [7]; however, this assumption may be violated in typical applications such as natural language processing [8] and computer vision [9,10].

This problem of data set shifting can result from the way input characteristics are used, from the way training and test sets are selected, from data sparsity, from shifts in data distribution due to non-stationary environments, and also from changes in activation patterns within layers of deep neural networks. Such a data set shift can cause misleading parameter tuning when performing test strategies such as cross-validation [11,12].

This is why engineering machine learning systems largely relies on the skill of the data scientist to examine and resolve such problems.

2.2. Lack of Uniqueness of Internal Configuration

First of all, in contrast to traditional engineering, there is a lack of uniqueness of internal configuration causing difficulties in model comparison. Systems based on machine learning, in particular deep learning models, are typically regarded as black boxes. However, it is not just simply the complex nested non-linear structure which matters as often pointed out in the literature, see Reference [13]. There are mathematical or physical systems which are also complex, nested and non-linear, and yet interpretable (e.g., wavelets, statistical mechanics). It is an amazing, unexpected phenomenon that such deep networks become easier to be optimized (trained) with an increasing number of layers, hence complexity, see References [14,15]. More precisely, to find a reasonable sub-optimum out of many equally good possibilities. As consequence, and in contrast to classical engineering, we lose uniqueness of the internal optimal state.

2.3. Lack of Confidence Measure

A further peculiarity of state of the art deep learning methods is the lack of confidence measure. In contrast to Bayesian based approaches to machine learning, most deep learning models do not offer a justified confidence measure of the model's uncertainties. For example, in classification models, the probability vector obtained in the top layer (predominantly softmax output) is often interpreted as model confidence, see, for example, Reference [16] or Reference [17]. However, functions like softmax can result in extrapolations with unjustified high confidence for points far from the training data, hence providing a false sense of safety [18]. Therefore, it seems natural to try to introduce the Bayesian approach also to DNN models. The resulting uncertainty measures (or, synonymously, confidence measures) rely on approximations of the posterior distribution regarding the weights given the data. As a promising approach in this context, variational techniques, for example, based on Monte Carlo dropout [19], allow to turn these Bayesian concepts into computationally tractable algorithms. The variational approach relies on the Kullback-Leibler divergence for measuring the dissimilarity between distributions. As a consequence, the resultant approximating distribution becomes concentrated around a single mode, underestimating the uncertainty beyond this mode. Thus, the resulting measure of confidence for a given instance remains unsatisfactory and there might be still regions with misinterpreted high confidence.

2.4. Lack of Control of High-Dimensionality Effects

Further, there is the still unsolved problem of lack of control of high-dimensionality effects. There are high dimensional effects, which are not yet fully understood in the context of deep learning, see References [20,21]. Such high-dimensional effects can cause instabilities as illustrated, for example, by the emergence of so-called adversarial examples, see for example, References [22,23].

3. Key Challenges of AI Model Lifecycle

AI model lifecycle refers to the development steps of data-driven modelling, starting from data conditioning as basis for model training to finding a solution configuration of a proposed machine learning model for the task at hand. Typically these steps aim at extracting higher level semantics and meaning from lower level representations as indicated in Figure 2.

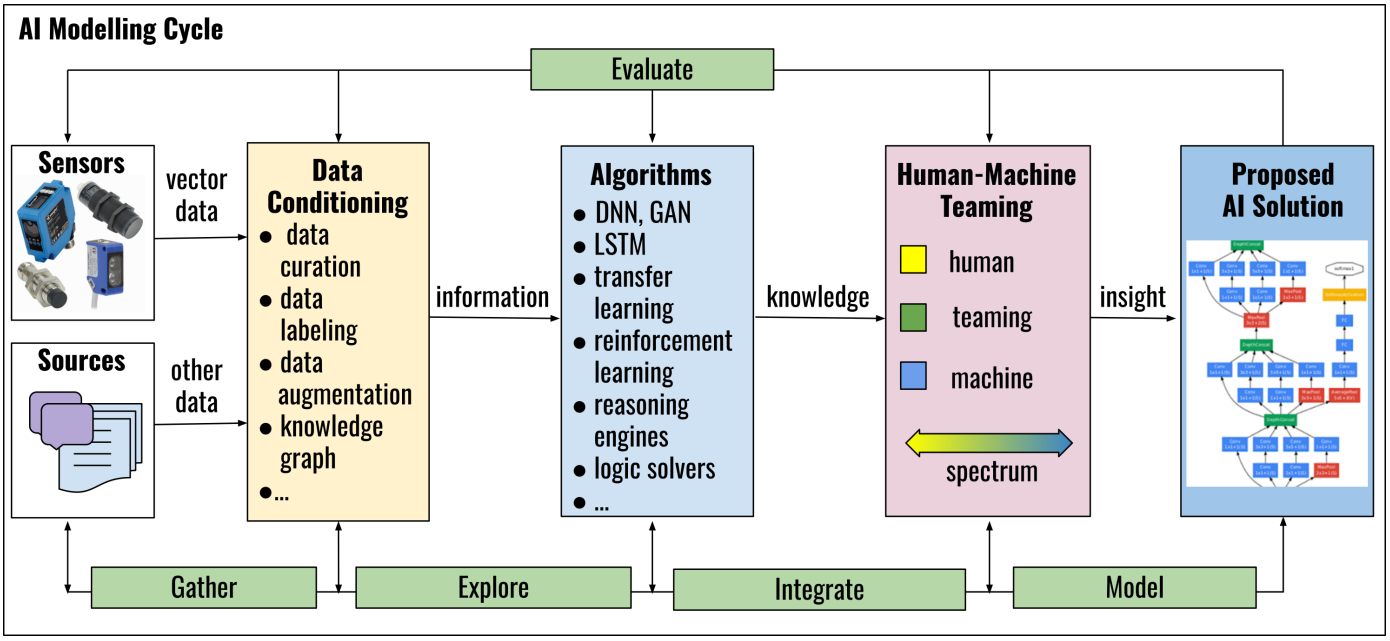

Figure 2. Steps of Developing AI Models.

3.1. Data Challenge: Data Augmentation with Pitfalls

Instead of using the raw sampled data directly, it has been become a standard technique in machine learning to apply additional data curation and data conditioning methods to enhance the expressiveness of data to be used for training. An example would be the imputation of missing values to increase the amount of training data [24]. This way, data augmentation techniques are used to improve the model's generalisation capabilities [25–28] and, also to mitigate adversarial vulnerability [25,29–31]. Some augmentation methods incorporate inductive bias into the model by (classical) invariance-preserving geometric transformations designed by domain experts, while others rely on statistical heuristics (e.g., Mixup [29]) or sampling from a learned data distribution (e.g., GANs [26,31]). But, by affecting a model's behavior beyond the given training data, any data augmentation strategy introduces a certain bias caused by its assumptions about the task at hand [32]. So, as in medicine, where a drug can have side effects, data augmentation might have side effects which have hardly been investigated yet [33].

3.2. Information Fusion Challenge

Often a single source of information does not provide the sufficient information that is required or is not reliable enough. Therefore, the exploitation of different sources and modalities having potentially complementing predictive power and noise topology is vital

for many applications to achieve the required robustness. There is ongoing progress in the integration of different modalities and representation variants of information. For example, generative adversarial networks (GAN) [34] successfully are applied to narrow the distribution deviation between modalities by an adversarial process; attention mechanisms [35] allow the localization of salient features from modalities such that they are similar or complementary to each other. Nevertheless there remain major challenges [36–39]:

- Current deep learning models cannot capture the fully semantic knowledge of the multimodal data. Although attention mechanisms can be used to mitigate these problems partly, they work implicitly and cannot be actively controlled. In this context the combination of deep learning with semantic fusion and reasoning strategies are promising approaches [39].
- In contrast to the widespread use of convenient and effective knowledge transfer strategies in the image and language domain, similar methods are not yet available for audio or video data, not to mention other fields of applications for example, in manufacturing.
- The situation is worsened when it comes to dynamically changing data with shifts in its distribution. The traditional method of deep learning for adopting to dynamic multimodal data is to train a new model when the data distribution changes. This, however, takes too much time and is therefore not feasible in many applications. A promising approach is the combination with transfer learning techniques, which aim to handle deviating distributions as outlined in References [40,41]. See also Section 2.1.

3.3. Model Integrity and Stability Challenge

Deep learning methods are known to be surprisingly prone to adversarial attacks in the form of small perturbations to the input data. This intriguing weakness was first discovered and demonstrated by Szegedy et al. [22] by means of images that remain almost imperceptible to humans. Such adversarial perturbations can be caused by targeted attacks to cause a neural network classifier to completely change its prediction, even with reported high confidence on the wrong prediction. This effect is both a security and a safety issue [22,42–44]. As pointed out in Section 2.4 the susceptibility to compromise model integrity and stability is closely related to some intrinsic of nowadays deep model architectures such as high-dimensional effects.

3.4. Security and Confidentiality Challenge

Neural networks are not just input-output functions, they also represent a form of memory mechanism via compressed representations of the training data stored within their weights. This can cause unintended memorization. It is therefore possible to partially reconstruct input data from the model parameters (weights) themselves [45]. Such model inversion attacks can cause severe data leakage [46]. The purpose of such attacks is often not to disrupt (poison) the learning mechanism, but to extract sensitive information in the process of or after the creation of the models. For example, membership inference attacks aim at determining whether a given sample is part of the training data or not [47–50]. Protection against membership inference attacks is of particular interest in GDPR-critical domains with sensitive personal data such as healthcare, e-government or customer data in trade and industry. In contrast, the goal of property inference attacks is not at the individual level (membership of a certain class), but rather at the level of aggregated properties of the training data such as amount of data [51]. Protection against property inference attacks are of particular interest in industry to keep secrets of underlying business models.

In the literature, the notions of *privacy* and *confidentiality* are used in this context. The former refers to personal data, for example, related to GDPR standards, while confidentiality is broader, taking also non-personal data such as company secrets into account. As this distinction is only a matter of application and not a conceptual one, we use them synonymously.

The necessity of techniques to protect privacy is emphasized by a growing number of attacks on machine learning models in potentially security-critical domains such as healthcare [52]. As a counter measure there is great interest in privacy-preserving AI that aims at allowing learning from data without disclosing the data itself. In this context, federated machine learning systems in combination with differential privacy have emerged as a promising approach [53]. Federated learning is based on the principle to keep the execution of data processing at the sites or devices where the data is kept. This way training iterations are performed locally and only results of the computation (e.g., updated neural network weights) are returned to a central repository to update the main model. Differential privacy is based on the idea to perturb the data in a way that allows statistical reasoning while reducing individually recognizable information [54]. This way differential privacy complicates considerably membership inference attacks. The main advantage is to maintain data sovereignty by keeping the data with its owner, while at the same time the training of algorithms on the data is made possible. But there is an unavoidable tradeoff between protection of confidentiality and other performance criteria such as accuracy or transparency [55,56]. Moreover, current federated learning systems rely on assumptions on the distribution of the data which often are not applicable for industrial applications [57,58].

3.5. Interpretability Challenge

Essential aspects of trusted AI are explainability and interpretability. While interpretability is about being able to discern the mechanics without necessarily knowing why. Explainability is being able to quite literally explain what is happening, for example, by referring to mechanical laws. It is well known that the great successes of machine learning in recent decades in terms of applicability and acceptance are relativized by the fact that they can be explained less easily with increasing complexity of the learning model [59–61]. Explainability of the solution is thus increasingly perceived as an inherent quality of the respective methods [61–64]. Particularly in the case of deep learning methods attempts to interpret the predictions made using parameters fail [64]. The necessity to obtain not only increasing prediction accuracy but also the interpretation of the solutions determined by ML or Deep Learning arises at the latest with the ethical [65,66], legal [67], psychological [68], medical [69,70], and sociological [71] questions tied to their application. The common element of these questions is the demand to clearly interpret the decisions proposed by AI. The complex of problems that derives from this aspect of artificial intelligence for explainability, transparency, trustworthiness, and so forth, is generally described with the term Explainable Artificial Intelligence, synonymously Explainable AI or XAI. Its broad relevance can be seen in the interdisciplinary nature of the scientific discussion that is currently taking place on such terms as interpretation, explanation and refined versions such as causability and causality in connection with AI methods [64,72–74].

3.6. Trust Challenge

In contrast to traditional computing, AI can now perform tasks that previously only humans were able to do. As such it contains the possibility to revolutionize every aspect of our society. The impact is far-reaching. First, with the increasing spread of AI systems, the interaction between humans and AI will increasingly become the dominant form of human-computer interaction [75]. Second, this development will shape the future workforce. PwC (https://www.pwc.com/gx/en/services/people-organisation/workforce-of-the-future/workforce-of-the-future-the-competing-forces-shaping-2030-pwc.pdf) predicts a relatively low displacement of jobs (around 3%) in the first wave of AI, but this could dramatically increase up to 30% by the mid-2030's. Therefore, human centered AI has started coming to the forefront of AI research based on postulated ethical principles for protecting human autonomy and preventing harm. Recent initiatives at national (https://www.whitehouse.gov/wp-content/uploads/2019/06/National-AI-Research-and-Development-Strategic-Plan-2019-Update-June-2019.pdf) and supra-national (https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai) level emphasize the

need for research in trusted AI. In contrast to interpretability, trust is a much more comprehensive concept. Trust is linked to the uncertainty about a possible malfunctioning or failure of the AI system as well as to circumstances of delegating control to a machine as a black box. Predictability and dependability of AI technology as well as the understanding of the technology's operations and the intentions of its creators are essential drivers of trust [76]. Particularly, in critical applications the user wants to understand the rationale behind a classification, and under which conditions the system is trustful and when not. Consequently, AI systems must make it possible to take these human needs of trust and social compatibility into account. On the other hand, we have to be aware of limitations and peculiarities of state of the art AI systems. Currently, the topic of trusted AI is discussed in different communities at different levels of abstraction:

- in terms of high level ethical guidelines (e.g., ethics boards such as algorithmwatch.org (https://algorithmwatch.org/en/project/ai-ethics-guidelines-global-inventory/), EU's Draft Ethics Guidelines (https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai));
- in terms of regulatory postulates for current AI systems regarding for example, transparency (working groups on standardization, for example, ISO/IEC JTC 1/SC 42 on artificial intelligence (https://www.iso.org/committee/6794475/x/catalogue/p/0/u/1/w/0/d/0));
- in terms of improved features of AI models (above all by explainable AI community [77,78]);
- in terms of trust modelling approaches (e.g., multi-agent systems community [76]).

In view of the model-intrinsic and system-technical challenges of AI that have been pointed out in the Sections 2 and 3, the gap between the envisioned high-level ethical guidelines of human-centered AI and the state of the art of AI systems becomes evident.

4. Key Challenges of AI System Lifecycle

In data-driven AI systems, there are two equally consequential components—software code and data. However, some input data are inherently volatile and may change over time. Therefore, it is important that these changes can be identified and tracked to fully understand the models and the final system [79]. To this end, the development of such data-driven systems has all the challenges of traditional software engineering combined with specific machine learning problems causing additional hidden technical debts [80].

4.1. Deployment Challenge and Computational Resource Constraints

The design and training of the learning algorithm and the inference of the resulting model are two different activities. The training is very computationally intensive and is usually conducted on a high performance platform [81]. It is an iterative process that leads to the selection of an optimal algorithm configuration, usually known as hyperparameter optimization, with accuracy as the only major goal of the design [82]. While the training process is usually conducted offline, inference very often has to deal with real-time constraints, tight power or energy budgets, and security threats. This dichotomy determines the need for multiple design re-spins (before a successful integration), potentially leading to long tuning phases, overloading the designers and producing results highly depending on their skills. Despite the variety of resources available, optimizing these heterogeneous computing architectures for performing low-latency and energy-efficient DL inference tasks without compromising performance is still a challenge [83].

4.2. Data and Software Quality

This section highlights quality assurance issues related to data and software maintenance.

4.2.1. Data Quality Assurance Challenge

While much of the research in machine learning and its theoretical foundation has focused on improving the accuracy and efficiency of training and inference algorithms, less

attention has been paid to the equally important practical problem of monitoring the quality of the data supplied to AI systems [84,85]. Due to the multi-dimensional nature of poor data quality, the gain of a comprehensive understanding is not trivial [86]. The culture and management of an organization is often a critical factor for data quality, which means that organizations with a high awareness for quality in general (e.g., production quality or manufacturing of high-quality products) are often more willing to deal with data quality [86,87].

Common causes for poor data quality are errors during data collection (e.g., by sensors or humans), complex and insufficiently defined data management processes, errors in the data integration pipeline, incorrect data usage, and the expiration of data (e.g., customer addresses or telephone numbers) [86]. With respect to data management, especially heterogeneous data sources and a large number of schema-free data pose additional challenges, which directly impact data extraction from multiple sources, data preparation, and data cleansing [88–90].

To select a proper method to assure (i.e., to measure and improve) the quality of data, on the one hand the intrinsic data characteristics, and on the other hand, the purpose of the data needs to be taken into account [91]. In complex AI systems, data quality needs to be monitored over the entire lifecycle: from data preparation, training, testing, and validating computational models. In the following, we summarize key data quality challenges, which appeared throughout our projects:

- *Missing data* is a prevalent problem in data sets. In industrial use cases, faulty sensors or errors during data integration are common causes for systematically missing values. Historically, a lot of research into missing data comes from the social sciences, especially with respect to survey data, whereas little research work deals with industrial missing data [24]. In terms of missing data handling, it is distinguished between *deletion* methods (where records with missing values are simply not used), and *imputation* methods, where missing values are replaced with estimated values for a specific analysis [24]. Little & Rubin [92] state that "the idea of imputation is both seductive and dangerous", pointing out the fact that the imputed data is pretended to be truly complete, but might have substantial bias that impairs inference. For example, the common practice of replacing missing values with the mean of the respective variable (known as *mean substitution*) clearly disturbs the variance of the respective variable as well as correlations to other variables. A more sophisticated statistical approach as investigated in Reference [24] is multiple imputation, where each missing value is replaced with a set of plausible values to represent the uncertainty caused by the imputation and to decrease the bias in downstream prediction tasks. In a follow-up research, also the integration of knowledge about missing data pattern is investigated.
- *Semantic shift* (also: semantic change, semantic drift) is a term originally stemming from linguistics and describes the evolution of word meaning over time, which can have different triggers and development [93]. In the context of data quality, *semantic shift* is defined as the circumstance when "the meaning of data evolves depending on contextual factors" [94]. Consequently, when these factors are modeled accordingly (e.g., described with rules), it is possible to handle semantic shift even in very complex environments as outlined in Reference [94]. While the most common ways to overcome semantic shift are rule-based approaches, more sophisticated approaches take into account the semantics of the data to reach a higher degree of automation. Example information about contextual knowledge are the respective sensor or machine with which the data is collected [94].
- *Duplicate data* describes the issue that one real-world entity has more than one representation in an information system [95–98]. This subtopic of data quality is also commonly referred to as entity resolution, redundancy detection, record linkage, record matching, or data merging [96]. Specifically, the detection of approximate duplicates has been researched intensively over the last decades [99].

A further challenge is the detection of outlying values, which are considered abnormalities, discordants, or deviants when compared to the remaining data [100]. We explicitly want to distinguish invalid data from outlying data. Although there is a plethora of research on statistical outlier detection (cf. Reference [100]), there are little automated and statistical approaches that detect invalid data beyond pure rule-based solutions [101]. The detection and distinction between invalid and outlying data is therefore at the same a practical challenge for companies and a scientific challenge in terms of methodology.

4.2.2. Software Quality: Configuration Maintenance Challenge

ML system developers usually start from ready-made, pre-trained networks and try to optimize their execution on the target processing platform as much as possible. This practice is prone to the entanglement problem [80]: If changes are made to an input feature, the meaning, weighting, or use of the other features may also change. This means that machine learning systems must be designed so that feature engineering and selection changes are easily tracked. Especially when models are constantly revised and subtly changed, the tracking of configuration updates while maintaining the clarity and flexibility of the configuration become an additional burden.

Furthermore, developing data preparation pipelines and ML systems requires detailed knowledge and expertise in the correct and optimal use of the existing libraries and frameworks. The rapid evolution of these libraries and frameworks associated with often incompatible API changes and outdated documentation, increases the potential confusion and the risk of making mistakes. A recent study [102] on deep learning bugs and anti-patterns in using popular libraries such as Caffe, Keras, Tensorflow, theano, and Torch found that these mistakes can lead to poor performance in model construction, crashes and hangs, for example, due to running out of memory, underperforming models and even data corruption.

5. Approaches, In-Progress Research and Lessons Learned

In this section, we discuss ongoing research facing the outlined challenges in the previous section, comprising:

(1) Automated and Continuous Data Quality Assurance, see Section 5.1;
(2) Domain Adaptation Approach for Tackling Deviating Data Characteristics at Training and Test Time, see Section 5.2;
(3) Hybrid Model Design for Improving Model Accuracy, see Section 5.3;
(4) Interpretability by Correction Model Approach, see Section 5.4;
(5) Software Quality by Automated Code Analysis and Documentation Generation, see Section 5.5;
(6) the ALOHA Toolchain for Embedded Platforms, see Section 5.6;
(7) Confidentiality-Preserving Transfer Learning, see Section 5.7;
(8) Human AI Teaming as Key to Human Centered AI, see Section 5.8.

5.1. Approach 1 on Automated and Continuous Data Quality Assurance

In times of large and volatile amounts of data, which are often generated automatically by sensors (e.g., in smart home solutions of housing units or industrial settings), it is especially important to, (i), automatically, and, (ii), continuously monitor the quality of data [79,87]. A recent study [101] shows that the continuous monitoring of data quality is only supported by very few software tools. In the open-source area these are Apache Griffin (https://griffin.incubator.apache.org), MobyDQ (https://github.com/mobydq/mobydq), and QuaIIe [88]. Apache Griffin and QuaIIe implement data quality metrics from the reference literature (see References [88,103]), whereby most of them require a reference database (gold standard) for calculation. Two examples from our research, which can be used for complete automated data quality measurement, are a novel metric to measure minimality (i.e., deduplication) in Reference [98], and a novel metric to measure the readability in Reference [104].

MobyDQ, on the other hand, is rule-based, with the focus on data quality checks along a pipeline, where data is compared between two different databases. Since existing open-source tools were insufficient for the permanent measurement of data quality within a database or a data stream used for data analysis and machine learning, we developed the Data Quality Library (DaQL) depicted in Figure 3 and introduced in Reference [85]. DaQL allows the extensive definition of data quality rules, based on the newly developed DaQL language. These rules do not require reference data and DaQL has already been used for a ML application in an industrial setting [85]. However, to ensure their validity, the rules for DaQL are created manually by domain experts. Recently, DaQL has been extended with entity models, which supports a user in the definition of data quality rules since domain knowledge about the underlying data structure is not necessary any more [105].

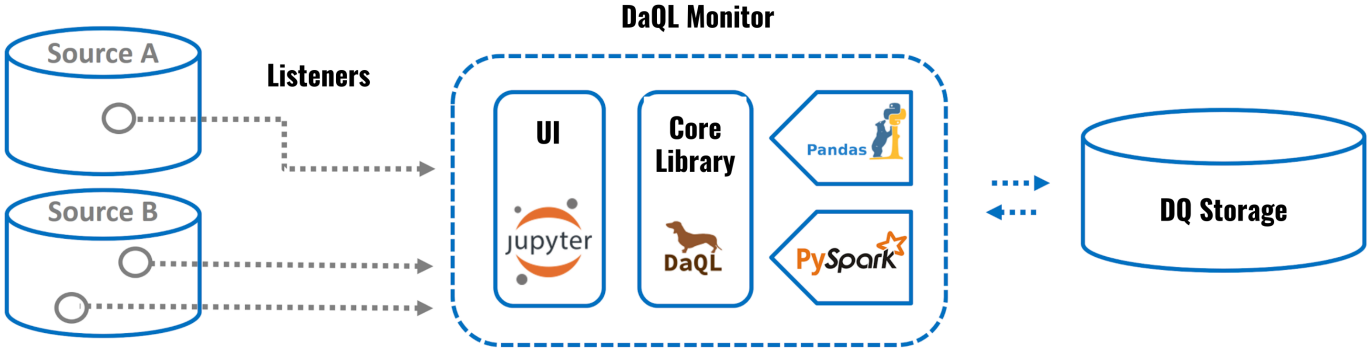

Figure 3. Architecture of Data Quality Library (DaQL) to Monitor Data Quality [85].

Lesson Learned: In the literature, data quality is typically defined with the fitness for use principle, which illustrates the high contextual dependency of the topic [91,106]. Thus, one important lesson learned is the need for more research into domain-specific approaches into data quality, which are at the same time suitable for automation [79]. An example from our ongoing research is the data quality tool DQ-MeeRKat (https://github.com/lisehr/dq-meerkat), which implements the novel concept of "reference data profiles" for automated data quality monitoring. Reference data profiles serve as quasi-gold-standard to automatically verify the quality of modified (i.e., inserted, updated, deleted) data. On the one hand, reference data profiles can be learned automatically and therefore require less human effort than rule-based approaches, and on the other hand (ii) they are adjusted to the respective data to be monitored and can therefore considered context-dependent.

As complement to the measurement (i.e., detection) of data quality issues, we consider research into the automated correction (i.e., cleansing) of sensor data as additional challenge [24]. Especially since automated data cleansing poses the risk to insert new errors in the data [95], which is specifically critical in enterprise settings.

In addition, the integration of contextual knowledge (e.g., the respective ML model using the data) needs to be considered. Here, knowledge graphs pose a promising solution (cf. Reference [107]), which indicates that knowledge about the quality of data is part of the bigger picture outlined in Section 5.8: the usage of knowledge graphs to interpret the quality of AI systems. However, also for data quality measurement, interpretability and explainability are considered a core requirement [101]. Therefore, we recommend to focus on clearly interpretable statistics and algorithms when measuring data quality since they prevent a user from deriving wrong conclusions from data quality measurement results [101].

5.2. Approach 2 on Domain Adaptation Approach for Tackling Deviating Data Characteristics at Training and Test Time

In References [9,10], we introduced a novel distance measure, the so-called Centralized Moment Discrepancy (CMD), for aligning probability distributions in the context of domain

adaption. Domain adaptation algorithms minimize the misclassification risk of a machine learning model for a *target* domain with little training data by adapting a model from a *source* domain with a large amount of training data. This is often done by mapping the domain-specific data samples in a new space where similarity is enforced by minimizing a probability metric, and, by subsequently learning a model on the mapped source data, see Figure 4.

In Reference [108] we can show that our CMD approach, refined by practice-oriented information-theoretic assumptions of the involved distributions, yields a generalization of the fundamental learning theoretic result of Vapnik [6]. As a result we obtain quantitative generalization bounds for recently proposed moment-based algorithms for unsupervised domain adaptation which perform particularly well in many applications such as object recognition [9,109], industrial manufacturing [110], analytical chemistry [111,112] and stereoscopic video analysis [113].

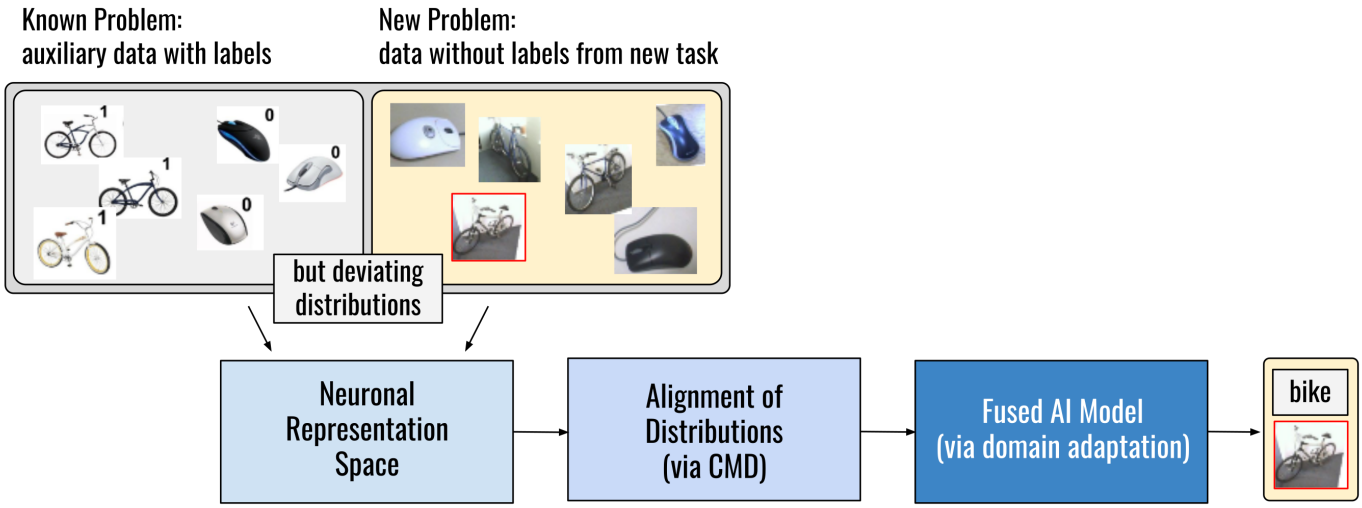

Figure 4. Illustration of domain adaptation: adapt learning capabilities from an auxiliary known problem with known labels to a new task with deviating distribution and unknown labels [114].

Lesson Learned: It is interesting that moment-based probability distance measure are The weakest among those utilized in the machine learning and, in particular, domain adaptation. Weak in this setting means that convergence by the stronger distance measures entails convergence of the weaker. Our lesson learned is that a weaker distance measure can be more robust than stronger distance measures. At the first glance, this observation might appear counter-intuitive. However, at a second look, it becomes intuitive that the minimization of stronger distance measures are more prone to the effect of negative transfer [115], that is, the adaptation of source-specific information not present in the target domain. Further evidence can be found in the area of generative adversarial networks where the alignment of distributions by strong probability metrics can cause problems of mode collapse which can be mitigated by choosing weaker similarity concepts [116]. Thus, it is better to abandon stronger concepts of similarity in favor of weaker ones and to use stronger concepts only if they can be justified.

5.3. Approach 3 on Hybrid Model Design for Improving Model Accuracy by Integrating Expert Hints in Biomedical Diagnostics

For diagnostics based on biomedical image analysis, image segmentation serves as a prerequisite step to extract quantitative information [117]. If, however, segmentation results are not accurate, quantitative analysis can lead to results that misrepresent the underlying biological conditions [118]. To extract features from biomedical images at a single cell level, robust automated segmentation algorithms have to be applied. In the Austrian FFG project

VISIOMICS (Platform supporting an integrated analysis of image and multiOMICs data based on liquid biopsies for tumor diagnostics—https://www.visiomics.at/), which is devoted to cell analysis, we tackle this problem by following a cell segmentation ensemble approach, consisting of several state-of-the-art deep neural networks [119,120]. In addition to overcome the lack of training data, which is very time consuming to prepare and annotate, we utilize a Generative Adversarial Network approach (GANs) for artificial training data generation [121] (Nuclear Segmentation Pipeline code available: https://github.com/SCCH-KVS/NuclearSegmentationPipeline). The underlying dataset was also published [122] and is available online (BioStudies: https://www.ebi.ac.uk/biostudies/studies/S-BSST265). The ensemble approach is depicted in Figure 5.

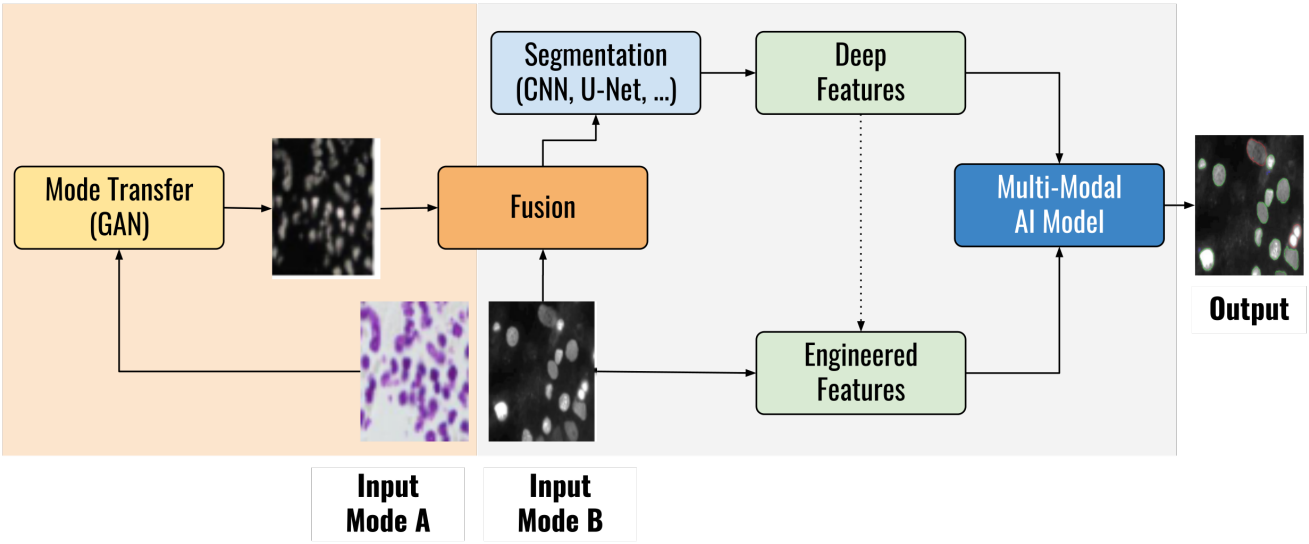

Figure 5. A cell segmentation ensemble approach in combination with Generative Adversarial Network approach (GANs) for multimodal data fusion.

Particularly for cancer diagnostics, clinical decision-making often relies on timely and cost-effective genome-wide testing. Similar to biomedical imaging, classical bioinformatic algorithms, often require manual data curation, which is error prone, extremely time-consuming, and thus has negative effects on time and cost efficiency. To overcome this problem, we developed the DeepSNP (DeepSNP code available: https://github.com/SCCH-KVS/deepsnp) network to learn from genome-wide single-nucleotide polymorphism array (SNPa) data and to classify the presence or absence of genomic breakpoints within large genomic windows with high precision and recall [123].

Lesson Learned: First, it is crucial to rely on expert knowledge when it comes to data augmentation strategies. This becomes more important the more complex the data is (high number of cores and overlapping cores). Less complex images do not necessarily benefit from data augmentation. Second, by introducing so-called localization units the network is able to gain the ability to exactly localize anomalies in terms of genomic breakpoints despite never experiencing their exact location during training. In this way we have learned that localization and attention units can be used to significantly ease the effort of annotating data.

5.4. Approach 4 on Interpretability by Correction Model Approach

Last year, at a symposium on predictive analytics in Vienna [124], we introduced an approach to the problem of formulating interpretability of AI models for classification or regression problems [125] with a given basis model, for example, in the context of model predictive control [126]. The basic idea is to root the problem of interpretability in the basic model by considering the contribution of the AI model as correction of this basis model and

is referred to as Before and After Correction Parameter Comparison (BAPC). The idea of small correction is a common approach in mathematics in the field of perturbation theory, for example of linear operators. In References [127,128] the idea of small-scale perturbation (in the sense of linear algebra) was used to give estimates of the probability of return of an odyssey on a percolation cluster. The notion of small influence appears here in a similar way via the measures of determination for the AI model compared to the basic model. Figure 6 visualizes the schema of BAPC.

According to BAPC, an AI-based correction of a solution of these problems, which is previously provided by a basic model, is interpretable in the sense of this basic model, if its effect can be described by its parameters. Since this effect refers to the estimated target variables of the data. In other words, an AI correction in the sense of a basic model is interpretable in the sense of this basic model exactly when the accompanying change of the target variable estimation can be characterized with the solution of the basic model under the corresponding parameter changes. The basic idea of the approach is thus to apply the explanatory power of the basic model to the correcting AI method in that their effect can be formulated with the help of the parameters of the basic model. BAPC's ability to use the basic model to predict the modified target variables makes it a so-called surrogate [62].

We have applied BAPC successfully to success-prediction of start-up companies with the AI-correction model trained on psychological profile data in the framework of the well-known newsvendor problem of econometrics [129] ("Best Service Innovation Award 2020" at ISM 2020 (http://www.msc-les.org/ism2020/)). The proposed solution for the interpretation of the AI correction is of course limited from the outset by the interpretation horizon of the basic model. In the case of our results using the psychometric data (such as 'risk-affinity'), it is desirable, however, to interpret their influence in terms of 'hard core' key performance indicators. Furthermore, it must be considered that the basic model is potentially too weak to describe the phenomena underlying the correction in accordance with the actual facts. We therefore distinguish between explainability and interpretability and, with the definition of interpretability in terms of the basic model introduced above, we do not claim to always be able to explain, but rather to be able to describe (i.e., interpret) the correction as a change of the solution using the basic model. This is achieved by means of the features used in the basic model and their modified parameters. As with most XAI approaches (e.g., feature importance vector [64]), the goal is to find the most significant changes in these parameters.

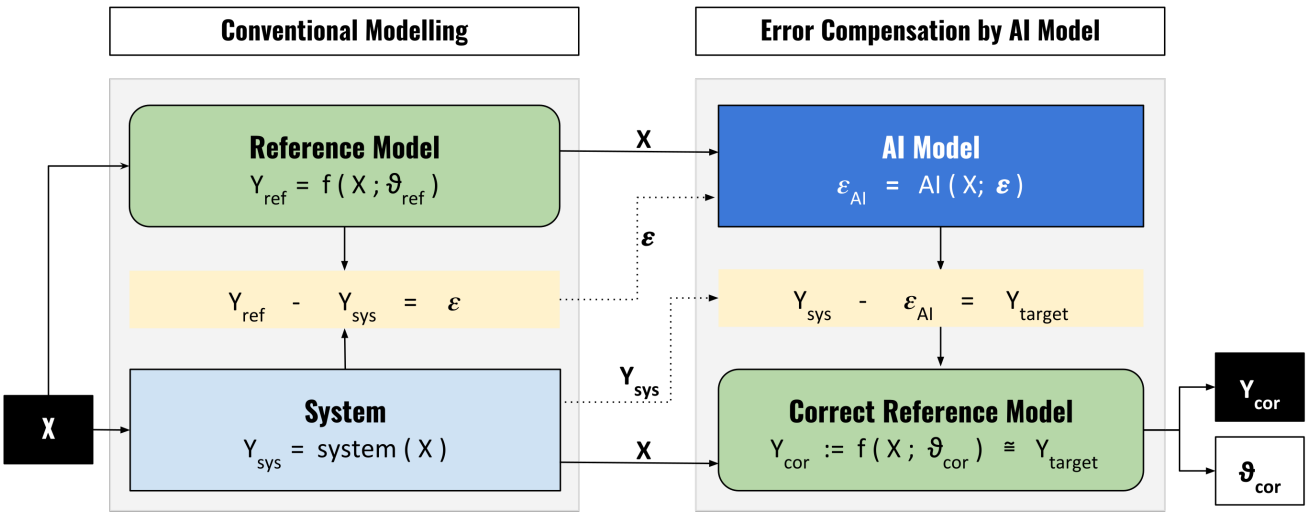

Figure 6. Schema of Before and After Correction Parameter Comparison (BAPC) [124]. Left: Reference Model produces prediction Y_{ref} by means of parameter ϑ_{ref} due to some conventional parameter identification method; Right: An AI Model is trained on $(X_i, \varepsilon_i)_i$ to compensate for the residuum of the reference model. The interpretation of the AI Model can be grounded on the meaning of the parameter of the reference model.

Lesson Learned: This approach is work in progress and will be tackled in detail in the upcoming Austrian FFG research project inAIco. As lesson learned we appreciate the BAPC approach as result of interdisciplinary research at the intersection of mathematics, machine learning and model predictive control. We expect that the approach generally only works for small AI corrections. It must be possible to formulate conditions about the size (i.e., smallness) of the AI correction under which the approach will work in any case. However, it is an advantage of our approach that interpretability does not depend on human understanding (see the discussion in References [62,64]). An important aspect is its mathematical rigidity, which avoids the accusation of quasi-scientificity (see Reference [130]).

5.5. Approach 5 on Software Quality by Code Analysis and Automated Documentation

Quality assurance measures in software engineering include, for example, automated testing [131], static code analysis [132], system redocumentation [133], or symbolic execution [134]. These measures need to be risk-based [135,136], exploiting knowledge about system and design dependencies, business requirements, or characteristics of the applied development process.

AI-based methods can be applied to extract knowledge from source code or test specifications to support this analysis. In contrast to manual approaches, which require extensive human annotation work, machine learning methods have been applied for various extraction and classification tasks, such as comment classification of software systems with promising results in References [137–139].

Software engineering approaches contribute to automate (i) AI-based system testing, for example, by means of predicting fault-prone parts of the software system that need particular attention [140], and (ii) system documentation to improve software maintainability [133,141,142] and to support re-engineering and migration activities [142]. In particular, we developed a feed-back directed testing approach to derive tests from interacting with a running system [143], which we successfully applied in various industry projects [144,145].

Also software redocumentation with the aim to recover outdated or non-existing documentation is becoming increasingly important in order to cope with raising complexity, to enhance human understanding, and to ensure compliance with company policies or legal regulations [146]. In an ongoing redocumentation project [147], we automatically generate parts of the functional documentation, containing business rules and domain concepts, and all the technical documentation. We also exploit source code comments, which provide key information about the underlying software, as valuable source of information (see Figure 7). We, therefore, apply classical machine learning techniques but also deep learning approaches using NLP, word embedding and novel approaches for character-to-image encoding [148]. By leveraging this ML/DL pipeline, it is possible to classify comments and thus transfer valuable information from the source code into documentation with less effort but the same quality than using a manual classification approach, for example, in the form of heuristics, which is usually time-consuming, error-prone and strongly dependent on programming languages or concrete software systems.

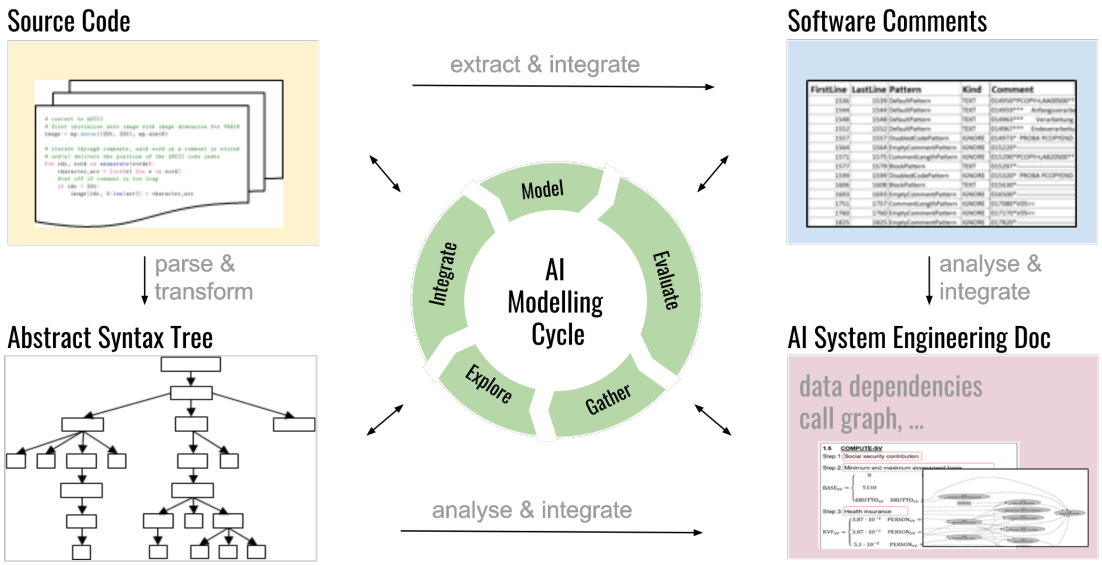

Figure 7. Automated software documentation for AI system engineering [148].

Lesson Learned: Keeping documentation up to date is essential for the maintainability of frequently updated software and to minimize the risk of technical debt due to the entanglement of data and sub-components of machine learning systems. The lesson learned is that for this problem also machine learning can be utilized when it comes to establishing rules for detecting and classifying comments (accuracy of >95%) and integrating them when generating readable documentation.

5.6. Approach 6 on the ALOHA Toolchain for Embedded AI Platforms

In References [149,150] we introduce ALOHA, an integrated tool flow that tries to make the design of deep learning (DL) applications and their porting on embedded heterogeneous architectures as simple and painless as possible. ALOHA is the result of interdisciplinary research funded by the EU (https://www.aloha-h2020.eu/). The proposed tool flow aims at automating different design steps and reducing development costs by bridging the gap between DL algorithm training and inference phases. The tool considers hardware-related variables and security, power efficiency, and adaptivity aspects during the whole development process, from pre-training hyperparameter optimization and algorithm configuration to deployment. According to Figure 8 the general architecture of the ALOHA software framework [151] consists of three major steps:

- (Step 1) algorithm selection,
- (Step 2) application partitioning and mapping, and
- (Step 3) deployment on target hardware.

Starting from a user-specified set of input definitions and data, including a description of the target architecture, the tool flow generates a partitioned and mapped neural network configuration, ready to the target processing architecture, which also optimizes predefined optimization criteria. The criteria for optimization include both application-level accuracy and the required security level, Inference execution time and power consumption. A RESTful microservices approach allows each step of the development process to be broken down into smaller, completely independent components that interact and influence each other through the exchange of HTTP calls [152]. The implementations of the various components are managed using a container orchestration platform. The standard ONNX (https://onnx.ai/) (Open Neural Network Exchange) is used to exchange deep learning models between the different components of the tool flow.

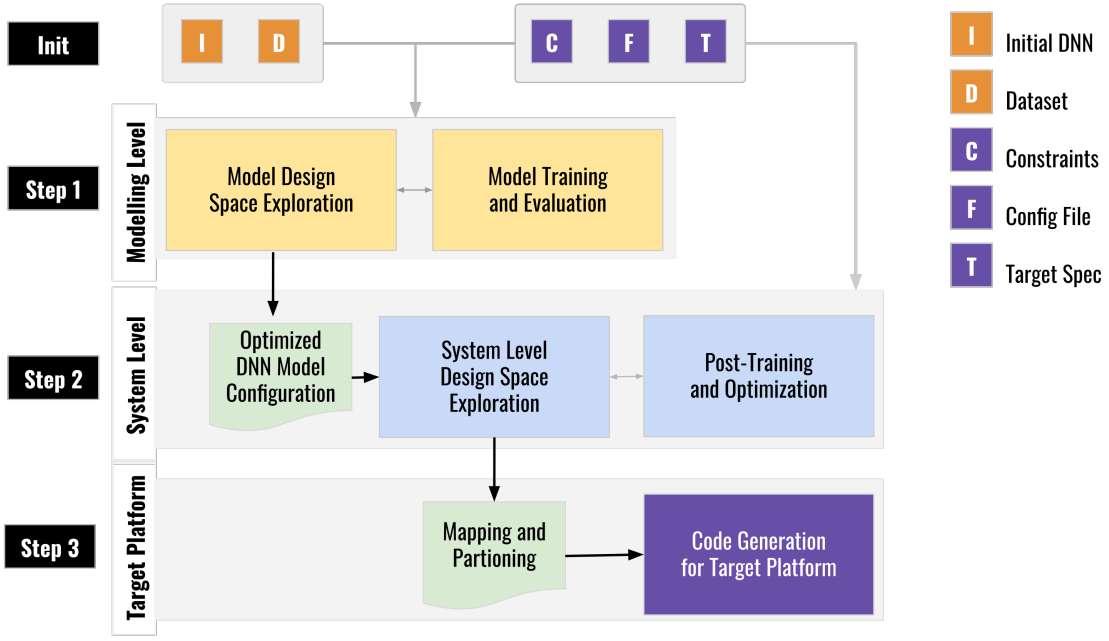

Figure 8. General architecture of the ALOHA software framework for Edge AI taking computational resource constraints at training time into account. Nodes in the upper part of the figure represent the key inputs of the tool flow specified by the users, for details see Reference [151].

In Step 1 a Design Space comprising admissible model architectures for hyperparamerter tuning is defined. This Design Space is configured via satellite tools that evaluate the fitness in terms of the predefined optimization criteria such as accuracy (by the Training Engine), robustness against adversarial attacks (by the Security evaluation tool) and power (by the Power evaluation tool). The optimization is based on (a) hyperparameter tuning based on a non-stochastic infinite-armed bandit approach [153], and (b) a parsimonious inference strategy that aims to reduce the bit depth of the activation values from initially 8bit to 4bit by a iterative quantization and retraining steps [154]. The optimization in Step 2 exploits genetic algorithm for surfing the design space and requiring evaluation of the candidate partitioning and mapping scheme to the satellite tools Sesame [155] and Architecture Optimization Workbench (AOW) [156].

The gain in performance was evaluated in terms of inference time needed to execute the modified model on NEURAghe [157], a Zynq-based processing platform that contains both a dual ARM Cortex A9 processor (667 MHz) and a CNN accelerator implemented in the programmable logic. The statistical analysis on the switching activity of our reference models showed that, on average, only about 65% of the kernels are active in the layers of the network throughout the target validation data set. The resulting model loses only 2% accuracy (baseline 70%) while achieving an impressive 48.31% reduction in terms of FLOPs.

Lesson Learned: Following the standard training procedure deep models tend to be oversized. This research shows that some of the CNN layers are operating in a static or close-to-static mode, enabling the permanent pruning of the redundant kernels from the model. But, the second optimization strategy dedicated to parsimonious inference turns out to more effective on pure software execution, since it more directly deactivates operations in the convolution process. All in all, this study shows that there is a lot of potential for optimisation and improvement compared to standard deep learning engineering approaches.

5.7. Approach 7 on Confidentiality-Preserving Transfer Learning

In our approach we, above all, tackle the following questions in the context of privacy-preserving federated learning settings:

(1) How to design a noise adding mechanism that achieves a given differential privacy-loss bound with the minimum loss in accuracy?
(2) How to quantify the privacy-leakage? How to determine the noise model with optimal tradeoff between privacy-leakage and the loss of accuracy?
(3) What is the scope of applicability in terms of assumptions on the distribution of the input data and, what is about model fusion in a transfer learning setting?

Questions (1) is dealt with in the ongoing H2020 project SERUMS (https://www.serums-h2020.org) and Austrian FFG research project PRIMAL and question (2) is addressed in the bi-national Germany-Austrian project KI-SIGS (Austrian sub-project PetAI) (https://ki-sigs.de/). SERUMS and KI-SIGS are motivated by privacy issues in healthcare systems while PRIMAL focuses on industrial applications. Question (1) is addressed in References [158,159] where first sufficient conditions for $(\epsilon, \delta)-$ differential privacy are derived and then using entropy as design parameter, the optimal noise distribution that minimizes the expected noise magnitude together with satisfying the sufficient conditions for $(\epsilon, \delta)-$differential privacy is derived. The optimal differentially private noise adding mechanism could be applied for distributed deep learning [159,160] where a privacy wall separates the private local training data from the globally shared data, and fuzzy sets and fuzzy rules are used to aggregate robustly the local deep fuzzy models for building the global model. For addressing Question (2), a conceptual and theoretical framework could be established which we will outline next in its main features. For details, see Reference [161]. Question (3) is above all of interest in industrial settings where transfer learning techniques become more and more important to overcome the limitations and costs of data acquisition in flexible production with more personalized products, thus less mass production and less big data per product specification. It is the central research topic in the ongoing project S3AI (https://www.s3ai.at). In Reference [58] we propose a software platform for this purpose. See Reference [57] for a similar approach.

Now let us outline the approach of Reference [161] related to (2) where privacy-leakage is quantified in-terms of mutual information between private data and publicly released data. There we introduce an information theoretic approach for analyzing the privacy-utility tradeoff for multivariate data. First, we conceptualize and specify the problem setting in terms of a data release mechanism that relies on source data which are partially private and marked as such. Given this data we propose a mathematical framework that allows us to express the tradeoff between privacy-leakage and loss of accuracy of to be learned features of interest. The situation is illustrated in Figure 9, where x denotes private data, $y(x)$ corresponding features. Only data are released after adding some perturbing noise v according to the differential privacy paradigm.

The resulting released noisy data $(\widetilde{x}, \widetilde{y}(\widetilde{x}))$ will deviate from the original data $(x, y(x))$. Now, the tradeoff problem (2) means to design a noise model that keeps the mutual information $I(x; \widetilde{x})$ below some specified bound while minimizing the expected distortion between the original, $y(x)$, and the resulting distorted features, $\widetilde{y}(\widetilde{x})$. This optimization problem can be solved by specifying a level of noise entropy, and then solving for the optimal noise model by means of a variational optimization method. This way the noise entropy becomes the key design parameter to control the tradeoff problem, which provides an approach to tackle question (2). It is shown in Reference [159] that the noise model optimization improves the tradeoff substantially, up to factor 4 compared to standard configuration with a Gaussian noise model.

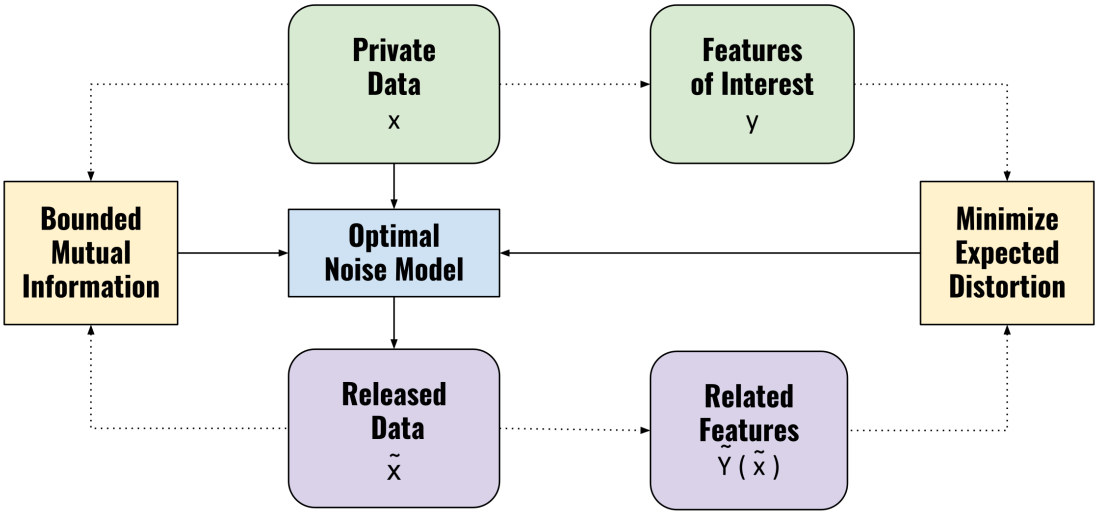

Figure 9. Design of optimal noise model for tackling the tradeoff between privacy-leakage (in terms of bounded mutual information between private data and perturbed released data) and feature distortion (loss of accuracy); for details see Reference [161].

Lesson Learned: Federated learning offers an infrastructural approach to privacy (and confidentiality, respectively), but further functionalities are required to enhance its privacy-preserving capabilities and scope of applicability. Most important, privacy-preservation of data-driven AI turns out to be a matter of trade-off between privacy-leakage, on the one hand, and loss of accuracy of the target AI model, on the other hand. In this context the concept of differential privacy provides a powerful means of system design. But, the standard design based on a Gaussian noise model is only sub-optimal. The improvement of this trade-off requires refined analysis, as for example, based on exploiting information-theoretic concepts that allow to turn this problem into a feasible optimization problem. However, particularly for industrial settings, when it comes to deviating statistical data characteristics of its sources, respectively, the target application, further research is required to enhance the scope of applicability of privacy-preserving federated learning towards transfer learning.

5.8. Approach 8 on Human AI Teaming as Key to Human Centered AI

In Reference [162], we introduce an approach for human-centered AI in working environments utilizing knowledge graphs and relational machine learning ([163,164]). This approach is currently being refined in the ongoing Austrian project *Human-centered AI in digitized working environments (AI@Work)*. The discussion starts with a critical analysis of the limitations of current AI systems whose learning/training is restricted to predefined structured data, most vector-based with a pre-defined format. Therefore, we need an approach that overcomes this restriction by utilizing a relational structures by means of a knowledge graph (KG) that allows to represent relevant context data for linking ongoing AI-based and human-based actions on the one hand and process knowledge and policies on the other hand. Figure 10 outlines this general approach where the knowledge graph is used as an intermediate representation of linked data to be exploited for improvement of the machine learning system, respectively AI system.

Methods applied in this context will include knowledge graph completion techniques that aim at filling missing facts within a knowledge graph [165]. The KG flexibly will allow tying together contextual knowledge about the team of involved human and AI based actors including interdependence relations, skills and tasks together with application and system process and organizational knowledge [166]. Relational machine learning will be developed in combination with an updatable knowledge graph embedding [167,168]. This relational ML will be exploited for analyzing and mining the knowledge graph for

the purpose of detecting inconsistencies, curating, refinement, providing recommendations for improvements and detecting compliance conflicts with predefined behavioral policies (e.g., ethic or safety policies). The system will learn from the environment, user feedback, changes in the application or deviations from committed behavioral patterns in order to react by providing updated recommendations or triggering actions in case of compliance conflicts. But, the construction of the knowledge graph and keeping it up-to-date is a critical step as it usually includes laborious efforts for knowledge extraction, knowledge fusion, knowledge verification and knowledge updates. In order to address this challenge, our approach pursues bootstrapping strategies for knowledge extraction by recent advances in deep learning and embedding representations as promising methods for matching knowledge items represented in diverse formats.

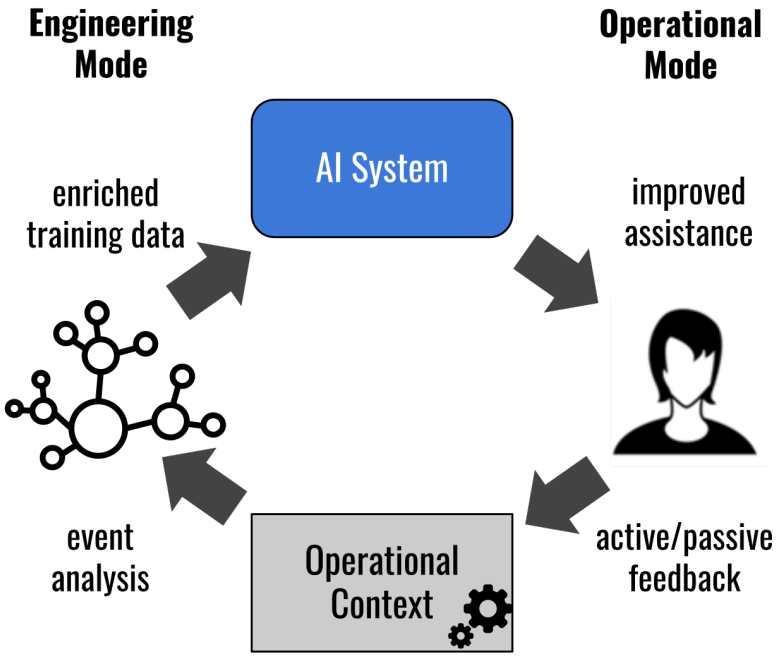

Figure 10. A knowledge-graph approach to enhance vector-based machine learning in order to support human AI teaming by taking context and process knowledge into account. A knowledge graph is used as an intermediate representation of data enriched with static and dynamic context information.

Lesson Learned: As pointed out in Section 3 there is a substantial gap between current state-of-the-art research of AI systems and the requirements posed by ethical guidelines. Future research will rely much more on machine learning on graph structures. Fast updatable knowledge graphs and related knowledge graph embeddings might be a key towards ethics by design enabling human centered AI.

6. Discussion and Conclusions

This paper can only give a small grasp of the broad field of AI research in connection with the application of machine learning in practice. The associated research is indeed inter- and even trans-disciplinary [169]. Nonetheless, we come to the conclusion that a discussion on AI System Engineering needs to start with its theoretical foundations and a critical discussion about the limitations of current data-driven AI systems as outlined in Sections 2–4. Approach 1, Section 5.1, and Approach 2, Section 5.2, help to stick to the theoretical prerequisites. Approach 1 contributes by reducing errors in the data and Approach 2 by extending the theory by relaxing its preconditions, bringing statistical learning theory closer to the needs of practice. However, building such systems and addressing the related challenges as outlined in Sections 3 and 4 requires a bunch of skills from different fields, predominantly model building and software engineering know-how. Approach 3, Section 5.3, and Approach 4, Section 5.4,

contribute to model building: Approach 3 by creatively adopting novel hybrid machine learning model architectures and Approach 4 by means of system theory that investigates AI as addendum to a basis model in order to be able to establish a notion of interpretability in a strict mathematical sense. Every model applied in practice must be coded in software. Approach 5, Section 5.5, outlines helpful state-of-the-art approaches in software engineering for maintaining the engineered software in good traceable and reusable quality which becomes more and more important with increasing complexity. Approach 6, Section 5.6, is an integrative approach that takes all the aspects discussed so far into account by proposing a software framework that supports the developer in all these steps when optimizing an AI system for embedded platforms. Approach 7 on confidentiality, Section 5.7, leads to fundamental questions of modeling and quantifying the trade-off between privacy-leakage and loss of accuracy of the target AI model. Finally, the challenge for human centered AI as outlined in Section 3.6 is somehow beyond of the state of the art. While most of the challenges described in this work require, above all, progress in the respective disciplines, the challenge for human centered AI addressing trust in the end will require a mathematical theory of trust, that is a trust modeling approach at the level of system engineering that takes the psychological and cognitive aspects of human trust into account as well. Approach 8, Section 5.8, contributes to this endeavor by its conceptional approach for human AI teaming and its analysis of its prerequisites from relational machine learning.

Author Contributions: Conceptualization, B.M. and L.F.; methodology, B.M., L.F., L.E., F.S., M.K. and W.Z.; software, V.G. and R.R.; validation, L.E., V.G., R.R., F.S., D.B. and W.Z.; formal analysis, F.S., W.Z., M.K. and B.M.; investigation, L.F., L.E., V.G., R.R., F.S., W.Z., D.B., M.K. and B.M.; resources, L.F. and R.R.; data curation, L.E., F.S., W.Z., D.B.; writing—original draft preparation, L.F. and B.M.; writing—review and editing, L.F., L.E., V.G., R.R., F.S., W.Z., D.B., M.K. and B.M.; visualization, B.M. and R.R.; supervision, B.M.; project administration, L.F. and R.R.; funding acquisition, L.F., R.R. and B.M. All authors have read and agreed to the published version of the manuscript.

Funding: The research reported in this paper has been funded by the Federal Ministry for Climate Action, Environment, Energy, Mobility, Innovation and Technology (BMK), the Federal Ministry for Digital and Economic Affairs (BMDW), and the Province of Upper Austria in the frame of the COMET—Competence Centers for Excellent Technologies Programme managed by Austrian Research Promotion Agency FFG.

Acknowledgments: Special thanks go to A Min Tjoa, former Scientific Director of SCCH, for his encouraging support in bringing together data and software science to tackle the research problems discussed in this paper.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
AOW	Architecture Optimization Workbench
BAPC	Before and After Correction Parameter Comparison
CMD	Centralized Moment Discrepancy
DaQL	Data Quality Library
DL	Deep Learning
DNN	Deep Neural Networks
GAN	Generative Adversarial Network
KG	Knowledge Graph
MDPI	Multidisciplinary Digital Publishing Institute
ML	Machine Learning
NLP	Natural Language Processing
ONNX	Open Neural Network Exchange
SNPa	Single-Nucleotide Polymorphism array
XAI	Explainable AI

References

1. Holzinger, A. Introduction to machine learning and knowledge extraction (MAKE). *Mach. Learn. Knowl. Extr.* **2017**, *1*, 1. [CrossRef]
2. Lecun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
3. Sünderhauf, N.; Brock, O.; Scheirer, W.; Hadsell, R.; Fox, D.; Leitner, J.; Upcroft, B.; Abbeel, P.; Burgard, W.; Milford, M.; et al. the limits and potentials of deep learning for robotics. *Int. J. Robot. Res.* **2018**, *37*, 405–420. [CrossRef]
4. Fischer, L.; Ehrlinger, L.; Geist, V.; Ramler, R.; Sobieczky, F.; Zellinger, W.; Moser, B. Applying AI in Practice: Key Challenges and Lessons Learned. In Proceedings of the Machine Learning and Knowledge Extraction—4th IFIP TC 5, TC 12, WG 8.4, WG 8.9, WG 12.9 International Cross-Domain Conference (CD-MAKE 2020), Dublin, Ireland, 25–28 August 2020; Lecture Notes in Computer Science; Holzinger, A., Kieseberg, P., Tjoa, A.M., Weippl, E.R., Eds.; Springer: Berlin/Heidelberg, Germany, 2020; Volume 12279, pp. 451–471.
5. Paleyes, A.; Urma, R.G.; Lawrence, N.D. Challenges in Deploying Machine Learning: A Survey of Case Studies. *arXiv* **2020**, arXiv:2011.09926.
6. Vapnik, V.N. *Statistical Learning Theory*; Wiley-Interscience: Hoboken, NJ, USA, 1998.
7. Quionero-Candela, J.; Sugiyama, M.; Schwaighofer, A.; Lawrence, N.D. *Dataset Shift in Machine Learning*; The MIT Press: Cambridge, MA, USA, 2009.
8. Jiang, J.; Zhai, C. Instance weighting for domain adaptation in NLP. In Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, Prague, Czech Republic, 25–27 June 2007; pp. 264–271.
9. Zellinger, W.; Grubinger, T.; Lughofer, E.; Natschläger, T.; Saminger-Platz, S. Central Moment Discrepancy (CMD) for Domain-Invariant Representation Learning. In Proceedings of the 5th International Conference on Learning Representations (ICLR 2017), Toulon, France, 24–26 April 2017.
10. Zellinger, W.; Moser, B.A.; Grubinger, T.; Lughofer, E.; Natschläger, T.; Saminger-Platz, S. Robust unsupervised domain adaptation for neural networks via moment alignment. *Inf. Sci.* **2019**, *483*, 174–191. [CrossRef]
11. Xu, G.; Huang, J.Z. Asymptotic optimality and efficient computation of the leave-subject-out cross-validation. *Ann. Stat.* **2012**, *40*, 3003–3030. [CrossRef]
12. Little, M.A.; Varoquaux, G.; Saeb, S.; Lonini, L.; Jayaraman, A.; Mohr, D.C.; Kording, K.P. Using and understanding cross-validation strategies. Perspectives on Saeb et al. *GigaScience* **2017**, *6*, gix020. [CrossRef]
13. Samek, W.; Wiegand, T.; Müller, K.R. Explainable Artificial Intelligence: Understanding, Visualizing and Interpreting Deep Learning Models. *arXiv* **2017**, arXiv:1708.08296.
14. Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; Vinyals, O. Understanding deep learning requires rethinking generalization. In Proceedings of the 5th International Conference on Learning Representations (ICLR 2017), Toulon, France, 24–26 April 2017.
15. Vidal, R.; Bruna, J.; Giryes, R.; Soatto, S. Mathematics of Deep Learning. *arXiv* **2017**, arXiv:1712.04741.
16. Gal, Y. Uncertainty in Deep Learning. Ph.D. Thesis, University of Cambridge, Cambridge, UK, 2016.
17. Guo, C.; Pleiss, G.; Sun, Y.; Weinberger, K.Q. On Calibration of Modern Neural Networks. In Proceedings of the 34th International Conference on Machine Learning (ICML'17), Sydney, Australia, 6–11 August 2017; Volume 70, pp. 1321–1330.
18. Hein, M.; Andriushchenko, M.; Bitterwolf, J. Why ReLU networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 41–50.
19. Gal, Y.; Ghahramani, Z. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In Proceedings of the 33rd International Conference on International Conference on Machine Learning (ICML'16), New York City, NY, USA, 19–24 June 2016; Volume 48, pp. 1050–1059.
20. Gorban, A.N.; Tyukin, I.Y. Blessing of dimensionality: Mathematical foundations of the statistical physics of data. *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* **2018**, *376*, 20170237. [CrossRef]
21. Galloway, A.; Taylor, G.W.; Moussa, M. Predicting Adversarial Examples with High Confidence. *arXiv* **2018**, arXiv:1802.04457.
22. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. *arXiv* **2014**, arXiv:1312.6199.
23. Athalye, A.; Engstrom, L.; Ilyas, A.; Kwok, K. Synthesizing Robust Adversarial Examples. *arXiv* **2017**, arXiv:1707.07397.
24. Ehrlinger, L.; Grubinger, T.; Varga, B.; Pichler, M.; Natschläger, T.; Zeindl, J. Treating Missing Data in Industrial Data Analytics. In Proceedings of the 2018 IEEE Thirteenth International Conference on Digital Information Management (ICDIM), Berlin, Germany, 24–26 September 2018; pp. 148–155.
25. Perez, L.; Wang, J. The effectiveness of data augmentation in image classification using deep learning. *arXiv* **2017**, arXiv:1712.04621.
26. Antoniou, A.; Storkey, A.J.; Edwards, H. Data Augmentation Generative Adversarial Networks. *arXiv* **2017**, arXiv:1711.04340.
27. Yun, S.; Han, D.; Oh, S.J.; Chun, S.; Choe, J.; Yoo, Y. Cutmix: Regularization strategy to train strong classifiers with localizable features. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 6023–6032.
28. Berthelot, D.; Carlini, N.; Goodfellow, I.; Papernot, N.; Oliver, A.; Raffel, C.A. Mixmatch: A holistic approach to semi-supervised learning. In Proceedings of the 33rd Annual Conference on Neural Information Processing Systems, Vancouver, Canada, 8–14 December 2019; pp. 5050–5060.
29. Zhang, H.; Cisse, M.; Dauphin, Y.N.; Lopez-Paz, D. mixup: Beyond empirical risk minimization. *arXiv* **2018**, arXiv:1710.09412.

30. Verma, V.; Lamb, A.; Beckham, C.; Najafi, A.; Mitliagkas, I.; Lopez-Paz, D.; Bengio, Y. Manifold Mixup: Better Representations by Interpolating Hidden States. In Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA, 10–15 June 2019; Chaudhuri, K., Salakhutdinov, R., Eds.; PMLR: Long Beach, CA, USA, 2019; Volume 97, pp. 6438–6447.
31. Xiao, C.; Li, B.; yan Zhu, J.; He, W.; Liu, M.; Song, D. Generating Adversarial Examples with Adversarial Networks. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18), Stockholm, Sweden, 13–19 July 2018; pp. 3905–3911.
32. Battaglia, P.W.; Hamrick, J.B.; Bapst, V.; Sanchez-Gonzalez, A.; Zambaldi, V.; Malinowski, M.; Tacchetti, A.; Raposo, D.; Santoro, A.; Faulkner, R.; et al. Relational inductive biases, deep learning, and graph networks. *arXiv* **2018**, arXiv:1806.01261.
33. Eghbal-zadeh, H.; Koutini, K.; Primus, P.; Haunschmid, V.; Lewandowski, M.; Zellinger, W.; Moser, B.A.; Widmer, G. On Data Augmentation and Adversarial Risk: An Empirical Analysis. *arXiv* **2020**, arXiv:2007.02650.
34. Wang, Y.; Wu, C.; Herranz, L.; van de Weijer, J.; Gonzalez-Garcia, A.; Raducanu, B. Transferring GANs: Generating Images from Limited Data. In *Computer Vision—ECCV 2018*; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 220–236.
35. Xu, K.; Ba, J.L.; Kiros, R.; Cho, K.; Courville, A.; Salakhutdinov, R.; Zemel, R.S.; Bengio, Y. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In Proceedings of the 32nd International Conference on International Conference on Machine Learning (ICML'15), Lille, France, 7–9 July 2015; Volume 37, pp. 2048–2057.
36. Chen, X.; Lin, X. Big Data Deep Learning: Challenges and Perspectives. *IEEE Access* **2014**, *2*, 514–525. [CrossRef]
37. Lahat, D.; Adali, T.; Jutten, C. Multimodal Data Fusion: An Overview of Methods, Challenges, and Prospects. *Proc. IEEE* **2015**, *103*, 1449–1477. [CrossRef]
38. Lv, Z.; Song, H.; Basanta-Val, P.; Steed, A.; Jo, M. Next-Generation Big Data Analytics: State of the Art, Challenges, and Future Research Topics. *IEEE Trans. Ind. Inform.* **2017**, *13*, 1891–1899. [CrossRef]
39. Guo, W.; Wang, J.; Wang, S. Deep Multimodal Representation Learning: A Survey. *IEEE Access* **2019**, *7*, 63373–63394. [CrossRef]
40. Thangarajah, A.; Wu, Q.; Yang, Y.; Safaei, A. Fusion of transfer learning features and its application in image classification. In Proceedings of the 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), Windsor, ON, Canada, 30 April–3 May 2017; pp. 1–5.
41. Tzeng, E.; Hoffman, J.; Darrell, T.; Saenko, K. Simultaneous Deep Transfer Across Domains and Tasks. In Proceedings of the IEEE International Conference on Computer Vision 2015, Santiago, Chile, 7–13 December 2015; pp. 4068–4076.
42. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. *arXiv* **2015**, arXiv:1412.6572.
43. Kurakin, A.; Goodfellow, I.; Bengio, S. Adversarial examples in the physical world. *arXiv* **2017**, arXiv:1607.02533.
44. Biggio, B.; Roli, F. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognit.* **2018**, *84*, 317–331. [CrossRef]
45. Carlini, N.; Liu, C.; Erlingsson, U.; Kos, J.; Song, D. the Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks. In Proceedings of the 28th USENIX Conference on Security Symposium (SEC'19), Santa Clara, CA, USA, 14–16 August 2019; USENIX Association: Berkeley, CA, USA, 2019; pp. 267–284.
46. Fredrikson, M.; Jha, S.; Ristenpart, T. Model Inversion Attacks That Exploit Confidence Information and Basic Countermeasures. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS '15), Denver, CO, USA, 12–16 October 2015; Association for Computing Machinery: New York, NY, USA, 2015; pp. 1322–1333.
47. Shokri, R.; Stronati, M.; Song, C.; Shmatikov, V. Membership Inference Attacks Against Machine Learning Models. In Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–24 May 2017; pp. 3–18.
48. Nasr, M.; Shokri, R.; Houmansadr, A. Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning. In Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 19–23 May 2019; pp. 739–753.
49. Truex, S.; Liu, L.; Gursoy, M.; Yu, L.; Wei, W. Demystifying Membership Inference Attacks in Machine Learning as a Service. *IEEE Trans. Serv. Comput.* **2019**. [CrossRef]
50. Long, Y.; Bindschaedler, V.; Wang, L.; Bu, D.; Wang, X.; Tang, H.; Gunter, C.A.; Chen, K. Understanding membership inferences on well-generalized learning models. *arXiv* **2018**, arXiv:1802.04889.
51. Ganju, K.; Wang, Q.; Yang, W.; Gunter, C.A.; Borisov, N. Property Inference Attacks on Fully Connected Neural Networks Using Permutation Invariant Representations. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, Toronto, ON, Canada, 15–19 October 2018; Association for Computing Machinery: New York, NY, USA, 2018; CCS '18; pp. 619–633.
52. Rigaki, M.; Garcia, S. A Survey of Privacy Attacks in Machine Learning. *arXiv* **2020**, arXiv:2007.07646.
53. Konečný, J.; McMahan, B.; Ramage, D. Federated Optimization:Distributed Optimization Beyond the Datacenter. *arXiv* **2015**, arXiv:1511.03575.
54. Dwork, C.; Roth, A. The Algorithmic Foundations of Differential Privacy. *Found. Trends theor. Comput. Sci.* **2014**, *9*, 211–407. [CrossRef]
55. Yang, M.; Song, L.; Xu, J.; Li, C.; Tan, G. the Tradeoff Between Privacy and Accuracy in Anomaly Detection Using Federated XGBoost. *arXiv* **2019**, arXiv:1907.07157.
56. Ah-Fat, P.; Huth, M. Optimal accuracy privacy trade-off for secure computations. *IEEE Trans. Inf. Theory* **2019**, *65*, 3165–3182. [CrossRef]

57. Hiessl, T.; Schall, D.; Kemnitz, J.; Schulte, S. Industrial Federated Learning—Requirements and System Design. In *Highlights in Practical Applications of Agents, Multi-Agent Systems, and Trust-Worthiness*; The PAAMS Collection; De La Prieta, F., Mathieu, P., Rincón Arango, J.A., El Bolock, A., Del Val, E., Jordán Prunera, J., Carneiro, J., Fuentes, R., Lopes, F., Julian, V., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 42–53.
58. Zellinger, W.; Wieser, V.; Kumar, M.; Brunner, D.; Shepeleva, N.; Gálvez, R.; Langer, J.; Fischer, L.; Moser, B. Beyond Federated Learning: On Confidentiality-Critical Machine Learning Applications in Industry. In Proceedings of the International Conference on Industry 4.0 and Smart Manufacturing (ISM), Dublin, Ireland, 7–11 June 2020; in press.
59. London, A. Artificial Intelligence and Black-Box Medical Decisions: Accuracy versus Explainability. *Hastings Cent. Rep.* **2019**, *49*, 15–21. [CrossRef]
60. Holzinger, A.; Kieseberg, P.; Weippl, E.; Tjoa, A. *Current Advances, Trends and Challenges of Machine Learning and Knowledge Extraction: From Machine Learning to Explainable AI*; CD-MAKE; Lecture Notes in Computer Science; Springer International: Cham, Switzerland, 2018; pp. 1–8.
61. Skala, K. (Ed.) Explainable Artificial Intelligence: A Survey. In Proceedings of the Croatian Society for Information and Communication Technology, Electronics and Microelectronics—MIPRO 2018, Opatija, Croatia, 21–25 May 2018; pp. 210–215.
62. Carvalho, D.V.; Pereira, E.M.; Cardoso, J.S. Machine Learning Interpretability: A Survey on Methods and Metrics. *Electronics* **2019**, *8*, 832. [CrossRef]
63. Doshi-Velez, F.; Kim, B. Towards A Rigorous Science of Interpretable Machine Learning. *arXiv* **2017**, arXiv:1702.08608.
64. Guidotti, R.; Monreale, A.; Ruggieri, S.; Turini, F.; Giannotti, F.; Pedreschi, D. A Survey of Methods for Explaining Black Box Models. *ACM Comput. Surv.* **2018**, *51*, 1–42. [CrossRef]
65. Obermeyer, Z.; Powers, B.; Vogeli, C.; Mullainathan, S. Dissecting racial bias in an algorithm used to manage the health of populations. *Science* **2019**, *366*, 447–453. [CrossRef]
66. Char, D.S.; Shah, N.H.; Magnus, D. Implementing Machine Learning in Health Care—Addressing Ethical Challenges. *N. Engl. J. Med.* **2018**, *378*, 981–983. [CrossRef]
67. Deeks, A. the Judicial Demand for Explainable Artificial Intelligence. *Columbia Law Rev.* **2019**, *119*, 1829–1850.
68. Lombrozo, T. Explanatory preferences shape learning and inference. *Trends Cogn. Sci.* **2016**, *20*, 748–759. [CrossRef] [PubMed]
69. Forcier, M.B.; Gallois, H.; Mullan, S.; Joly, Y. Integrating artificial intelligence into health care through data access: Can the GDPR act as a beacon for policymakers? *J. Law Biosci.* **2019**, *6*, 317–335. [CrossRef] [PubMed]
70. Holzinger, A.; Langs, G.; Denk, H.; Zatloukal, K.; Müller, H. Causability and explainability of artificial intelligence in medicine. *WIREs Data Min. Knowl. Discov.* **2019**, *9*, e1312. [CrossRef] [PubMed]
71. Zou, J.; Schiebinger, L. AI can be sexist and racist—It's time to make it fair. *Nature* **2018**, *559*, 324–326. [CrossRef] [PubMed]
72. Gilpin, L.H.; Bau, D.; Yuan, B.Z.; Bajwa, A.; Specter, M.; Kagal, L. Explaining Explanations: An Overview of Interpretability of Machine Learning. In Proceedings of the 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA), Turin, Italy, 1–3 October 2018; pp. 80–89.
73. Holzinger, A. Interactive Machine Learning for Health Informatics: When do we need the human-in-the-loop? *Brain Inform.* **2016**, *3*, 119–131. [CrossRef] [PubMed]
74. Holzinger, A.; Carrington, A.; Müller, H. Measuring the Quality of Explanations: The System Causability Scale (SCS). In *KI-Künstliche Intelligenz (German J. Artif. Intell.)*; Special Issue on Interactive Machine Learning; Kersting, K., Ed.; TU Darmstadt, Darmstadt, Germany, 2020; Volume 34, pp. 193–198.
75. Amershi, S.; Weld, D.; Vorvoreanu, M.; Fourney, A.; Nushi, B.; Collisson, P.; Suh, J.; Iqbal, S.; Bennett, P.N.; Inkpen, K.; et al. Guidelines for Human-AI Interaction. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19), Glasgow, UK, 4–9 May 2019.
76. Cohen, R.; Schaekermann, M.; Liu, S.; Cormier, M. Trusted AI and the Contribution of Trust Modeling in Multiagent Systems. In Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS '19), Montreal, QC, Canada, 13–17 July 2019; pp. 1644–1648.
77. Gunning, D. DARPA's Explainable Artificial Intelligence (XAI) Program. In Proceedings of the 24th International Conference on Intelligent User Interfaces (IUI '19), Marina del Ray, CA, USA, 16–20 March 2019; Association for Computing Machinery: New York, NY, USA, 2019; p. ii.
78. Hoffman, R.R.; Mueller, S.T.; Klein, G.; Litman, J. Metrics for Explainable AI: Challenges and Prospects. *arXiv* **2019**, arXiv:1812.04608.
79. Ehrlinger, L.; Wöß, W. Automated Data Quality Monitoring. In Proceedings of the 22nd MIT International Conference on Information Quality (ICIQ 2017), Little Rock, AR, USA, 6–7 October 2017; pp. 15.1–15.9.
80. Sculley, D.; Holt, G.; Golovin, D.; Davydov, E.; Phillips, T.; Ebner, D.; Chaudhary, V.; Young, M.; Crespo, J.F.; Dennison, D. Hidden Technical Debt in Machine Learning Systems. In Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS), Montreal, QC, Canada, 8–13 December 2014.; pp. 2503–2511.
81. Wang, Y.E.; Wei, G.Y.; Brooks, D. Benchmarking TPU, GPU, and CPU Platforms for Deep Learning. *arXiv* **2019**, arXiv:1907.10701.
82. Yu, T.; Zhu, H. Hyper-Parameter Optimization: A Review of Algorithms and Applications. *arXiv* **2020**, arXiv:2003.05689.
83. Bensalem, M.; Dizdarević, J.; Jukan, A. Modeling of Deep Neural Network (DNN) Placement and Inference in Edge Computing. In Proceedings of the 2020 IEEE International Conference on Communications Workshops (ICC Workshops), Dublin, Ireland, 7–11 June 2020; pp. 1–6.

84. Breck, E.; Zinkevich, M.; Polyzotis, N.; Whang, S.; Roy, S. Data Validation for Machine Learning. *Proc. Mach. Learn. Syst.* **2019**, *1*, 334–347.
85. Ehrlinger, L.; Haunschmid, V.; Palazzini, D.; Lettner, C. A DaQL to Monitor Data Quality in Machine Learning Applications. In Proceedings of the International Conference on Database and Expert Systems Applications (DEXA), Linz, Austria, 26–29 August 2019; *Lecture Notes in Computer Science*; Springer: Cham, Switzerland, 2019; Volume 11706, pp. 227–237.
86. Apel, D.; Behme, W.; Eberlein, R.; Merighi, C. *Datenqualität erfolgreich steuern: Praxislösungen für Business-Intelligence-Projekte [Successfully Governing Data Quality: Practical Solutions for Business-Intelligence Projects]*; TDWI, Ed.; dpunkt.verlag GmbH: Heidelberg, Germany, 2015.
87. Sebastian-Coleman, L. *Measuring Data Quality for Ongoing Improvement*; Elsevier: Amsterdam, The Netherlands, 2013.
88. Ehrlinger, L.; Werth, B.; Wöß, W. Automated Continuous Data Quality Measurement with QuaIIe. *Int. J. Adv. Softw.* **2018**, *11*, 400–417.
89. Cagala, T. Improving data quality and closing data gaps with machine learning. In *Data Needs and Statistics Compilation for Macroprudential Analysis*; Settlements, B.F.I., Ed.; Bank for International Settlements: Basel, Switzerland, 2017; Volume 46.
90. Ramler, R.; Wolfmaier, K. Issues and effort in integrating data from heterogeneous software repositories and corporate databases. In Proceedings of the Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, Kaiserslautern, Germany, October 2008; pp. 330–332.
91. Wang, R.Y.; Strong, D.M. Beyond Accuracy: What Data Quality Means to Data Consumers. *J. Manag. Inf. Syst.* **1996**, *12*, 5–33. [CrossRef]
92. Little, R.J.A.; Rubin, D.B. *Statistical Analysis with Missing Data*, 2nd ed.; John Wiley & Sons, Inc.: New York, NY, USA, 2002.
93. Bloomfield, L. *Language*; Allen & Unwin: Crows Nest, Australia, 1933.
94. Ehrlinger, L.; Lettner, C.; Himmelbauer, J. Tackling Semantic Shift in Industrial Streaming Data Over Time. In Proceedings of the Twelfth International Conference on Advances in Databases, Knowledge, and Data Applications (DBKDA 2020), Lisbon, Portugal, 27 September–1 October 2020; pp. 36–39.
95. Maydanchik, A. *Data Quality Assessment*; Technics Publications, LLC.: Bradley Beach, NJ, USA, 2007.
96. Talburt, J.R. *Entity Resolution and Information Quality*; Elsevier: Amsterdam, The Netherlands, 2011.
97. Talburt, J.R.; Zhou, Y. A Practical Guide to Entity Resolution with OYSTER. In *Handbook of Data Quality*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 235–270.
98. Ehrlinger, L.; Wöß, W. A Novel Data Quality Metric for Minimality. In *Data Quality and Trust in Big Data*; Lecture Notes in Computer Science; Hacid, H., Sheng, Q.Z., Yoshida, T., Sarkheyli, A., Zhou, R., Eds.; Springer International Publishing: Cham, Switzerland, 2019; Volume 11235, pp. 1–15.
99. Stonebraker, M.; Bruckner, D.; Ilyas, I.F.; Beskales, G.; Cherniack, M.; Zdonik, S.B.; Pagan, A.; Xu, S. Data Curation at Scale: The Data Tamer System. In Proceedings of the 6th Biennial Conference on Innovative Data Systems Research (CDIR'13), Asilomar, CA, USA, 6–9 January 2013.
100. Aggarwal, C.C. *Outlier Analysis*, 2nd ed.; Springer International Publishing: New York, NY, USA, 2017; p. 446.
101. Ehrlinger, L.; Rusz, E.; Wöß, W. A Survey of Data Quality Measurement and Monitoring Tools. *arXiv* **2019**, arXiv:1907.08138.
102. Islam, M.J.; Nguyen, G.; Pan, R.; Rajan, H. A comprehensive study on deep learning bug characteristics. In Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Tallinn, Estonia, August 2019; pp. 510–520.
103. Heinrich, B.; Hristova, D.; Klier, M.; Schiller, A.; Szubartowicz, M. Requirements for Data Quality Metrics. *J. Data Inf. Qual.* **2018**, *9*, 1–32. [CrossRef]
104. Ehrlinger, L.; Huszar, G.; Wöß, W. A Schema Readability Metric for Automated Data Quality Measurement. In Proceedings of the Eleventh International Conference on Advances in Databases, Knowledge, and Data Applications (DBKDA 2019), Athens, Greece, 2–6 June 2019; pp. 4–10.
105. Lettner, C.; Stumptner, R.; Fragner, W.; Rauchenzauner, F.; Ehrlinger, L. DaQL 2.0: Measure Data Quality Based on Entity Models. In *Proceedings of the International Conference on Industry 4.0 and Smart Manufacturing (ISM 2020)*; Elsevier: Amsterdam, The Netherlands, 2020.
106. Chrisman, N. The role of quality information in the long-term functioning of a Geographic Information System. *Cartogr. Int. J. Geogr. Inf. Geovis.* **1983**, *21*, 79–88.
107. Ehrlinger, L.; Wöß, W. Towards a Definition of Knowledge Graphs. In Proceedings of the 12th International Conference on Semantic Systems—SEMANTiCS2016 and 1st International Workshop on Semantic Change & Evolving Semantics (SuCCESS16), Sun SITE Central Europe (CEUR), Technical University of Aachen (RWTH), Leipzig Germany, 12–15 September 2016; Volume 1695, pp. 13–16.
108. Zellinger, W.; Moser, B.A.; Saminger-Platz, S. On generalization in moment-based domain adaptation. *Ann. Math. Artif. Intell.* **2020**, 1–37. [CrossRef]
109. Sun, B.; Saenko, K. Deep coral: Correlation alignment for deep domain adaptation. In *Workshop of the European Conference on Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 443–450.
110. Zellinger, W.; Grubinger, T.; Zwick, M.; Lughofer, E.; Schöner, H.; Natschläger, T.; Saminger-Platz, S. Multi-source transfer learning of time series in cyclical manufacturing. *J. Intell. Manuf.* **2020**, *31*, 777–787. [CrossRef]

111. Nikzad-Langerodi, R.; Zellinger, W.; Lughofer, E.; Saminger-Platz, S. Domain-Invariant Partial-Least-Squares Regression. *Anal. Chem.* **2018**, *90*, 6693–6701. [CrossRef]
112. Nikzad-Langerodi, R.; Zellinger, W.; Saminger-Platz, S.; Moser, B.A. Domain adaptation for regression under Beer–Lambert's law. *Knowl.-Based Syst.* **2020**, *210*, 106447. [CrossRef]
113. Zellinger, W.; Moser, B.A.; Chouikhi, A.; Seitner, F.; Nezveda, M.; Gelautz, M. Linear optimization approach for depth range adaption of stereoscopic videos. *Electron. Imaging* **2016**, *2016*, 1–6. [CrossRef]
114. Zellinger, W. Moment-Based Domain Adaptation: Learning Bounds and Algorithms. Ph.D. Thesis, JKU, Linz, Austria, 2020.
115. Pan, S.J.; Yang, Q. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **2009**, *22*, 1345–1359. [CrossRef]
116. Eghbal-zadeh, H.; Zellinger, W.; Widmer, G. Mixture density generative adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 5820–5829.
117. Méhes, G.; Luegmayr, A.; Kornmüller, R.; Ambros, I.M.; Ladenstein, R.; Gadner, H.; Ambros, P.F. Detection of disseminated tumor cells in neuroblastoma: 3 log improvement in sensitivity by automatic immunofluorescence plus FISH (AIPF) analysis compared with classical bone marrow cytology. *Am. J. Pathol.* **2003**, *163*, 393–399. [CrossRef]
118. Jung, C.; Kim, C. Impact of the accuracy of automatic segmentation of cell nuclei clusters on classification of thyroid follicular lesions. *Cytom. Part A J. Int. Soc. Anal. Cytol.* **2014**, *85*, 709–718. [CrossRef] [PubMed]
119. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*; Medical Image Computing and Computer-Assisted Intervention—MICCAI; Navab, N.; Hornegger, J.; Wells, W.M.; Frangi, A.F., Eds.; Springer: Cham, Switzerland, 2015; pp. 234–241.
120. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
121. Kromp, F.; Fischer, L.; Bozsaky, E.; Ambros, I.; Doerr, W.; Taschner-Mandl, S.; Ambros, P.; Hanbury, A. Deep Learning architectures for generalized immunofluorescence based nuclear image segmentation. *arXiv* **2019**, arXiv:1907.12975.
122. Kromp, F.; Bozsaky, E.; Rifatbegovic, F.; Fischer, L.; Ambros, M.; Berneder, M.; Weiss, T.; Lazic, D.; Dörr, W.; Hanbury, A.; et al. An annotated fluorescence image dataset for training nuclear segmentation methods. *Sci. Data* **2020**, *7*, 1–8. [CrossRef]
123. Eghbal-Zadeh, H.; Fischer, L.; Popitsch, N.; Kromp, F.; Taschner-Mandl, S.; Gerber, T.; Bozsaky, E.; Ambros, P.F.; Ambros, I.M.; Widmer, G.; et al. DeepSNP: An End-to-End Deep Neural Network with Attention-Based Localization for Breakpoint Detection in Single-Nucleotide Polymorphism Array Genomic Data. *J. Comput. Biol.* **2018**, *26*, 572–596. [CrossRef]
124. Sobieczky, F. Explainability of models with an interpretable base model: explainability vs. accuracy. In Proceedings of Symposium on Predictive Analytics, Austin, TX, USA, 24–25 September 2020.
125. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2009.
126. Grancharova, A.; Johansen, T.A., Nonlinear Model Predictive Control. In *Explicit Nonlinear Model Predictive Control: Theory and Applications*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 39–69.
127. Sobieczky, F. An Interlacing Technique for Spectra of Random Walks and Its Application to Finite Percolation Clusters. *J. Theor. Probab.* **2010**, *23*, 639—670. [CrossRef]
128. Sobieczky, F. Bounds for the annealed return probability on large finite percolation graphs. *Electron. J. Probab.* **2012**, *17*, 17. [CrossRef]
129. Neugebauer, S.; Rippitsch, L.; Sobieczky, F.; Geiß, M. Explainability of AI-predictions based on psychological profiling. *Procedia Comput. Sci.* **2021**, unpublished work.
130. Lipton, Z.C. the Mythos of Model Interpretability. *Queue* **2018**, *16*, 31–57. [CrossRef]
131. Anand, S.; Burke, E.K.; Chen, T.Y.; Clark, J.; Cohen, M.B.; Grieskamp, W.; Harman, M.; Harrold, M.J.; Mcminn, P.; Bertolino, A. An orchestrated survey of methodologies for automated software test case generation. *J. Syst. Softw.* **2013**, *86*, 1978–2001. [CrossRef]
132. Nielson, F.; Nielson, H.R.; Hankin, C. *Principles of Program Analysis*; Springer: Berlin/Heidelberg, Germany, 2015.
133. Moser, M.; Pichler, J.; Fleck, G.; Witlatschil, M. Rbg: A documentation generator for scientific and engineering software. In Proceedings of the 2015 IEEE 22nd International Conference on Software Analysis, Evolution, and Reengineering (SANER), Montreal, QC, Canada, 2–6 March 2015; pp. 464–468.
134. Baldoni, R.; Coppa, E.; D'elia, D.C.; Demetrescu, C.; Finocchi, I. A survey of symbolic execution techniques. *ACM Comput. Surv. (CSUR)* **2018**, *51*, 1–39. [CrossRef]
135. Felderer, M.; Ramler, R. Integrating risk-based testing in industrial test processes. *Softw. Qual. J.* **2014**, *22*, 543–575. [CrossRef]
136. Ramler, R.; Felderer, M. A process for risk-based test strategy development and its industrial evaluation. In *International Conference on Product-Focused Software Process Improvement*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 355–371.
137. Pascarella, L.; Bacchelli, A. Classifying code comments in Java open-source software systems. In Proceedings of the 2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR), Buenos Aires, Argentina, 20–21 May 2017; pp. 227–237.
138. Shinyama, Y.; Arahori, Y.; Gondow, K. Analyzing code comments to boost program comprehension. In Proceedings of the 2018 IEEE 25th Asia-Pacific Software Engineering Conference (APSEC), Nara, Japan, 4–7 December 2018; pp. 325–334.
139. Steidl, D.; Hummel, B.; Juergens, E. Quality analysis of source code comments. In Proceedings of the 2013 IEEE 21st International Conference on Program Comprehension (ICPC), San Francisco, CA, USA, 20–21 May 2013; pp. 83–92.

140. Menzies, T.; Milton, Z.; Turhan, B.; Cukic, B.; Jiang, Y.; Bener, A. Defect prediction from static code features: Current results, limitations, new approaches. *Autom. Softw. Eng.* **2010**, *17*, 375–407. [CrossRef]
141. Van Geet, J.; Ebraert, P.; Demeyer, S. Redocumentation of a legacy banking system: An experience report. In Proceedings of the Joint ERCIM Workshop on Software Evolution (EVOL) and International Workshop on Principles of Software Evolution (IWPSE), Antwerp, Belgium, 20–21 September 2010; pp. 33–41.
142. Dorninger, B.; Moser, M.; Pichler, J. Multi-language re-documentation to support a COBOL to Java migration project. In Proceedings of the 2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER), Klagenfurt, Austria, 20–24 Feburary 2017; pp. 536–540.
143. Ma, L.; Artho, C.; Zhang, C.; Sato, H.; Gmeiner, J.; Ramler, R. Grt: Program-analysis-guided random testing (t). In Proceedings of the 2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE), Lincoln, NE, USA, 9–13 November 2015; pp. 212–223.
144. Ramler, R.; Buchgeher, G.; Klammer, C. Adapting automated test generation to GUI testing of industry applications. *Inf. Softw. Technol.* **2018**, *93*, 248–263. [CrossRef]
145. Fischer, S.; Ramler, R.; Linsbauer, L.; Egyed, A. Automating test reuse for highly configurable software. In Proceedings of the 23rd International Systems and Software Product Line Conference-Volume A, Paris, France, 9–13 September 2019; pp. 1–11.
146. Hübscher, G.; Geist, V.; Auer, D.; Hübscher, N.; Küng, J. Integration of Knowledge and Task Management in an Evolving, Communication-intensive Environment. In Proceedings of the 22nd International Conference on Information Integration and Web-based Applications & Services (iiWAS2020), Chiang Mai, Thailand, 30 November–2 December 2020; ACM: New York, NY, USA, 2020; pp. 407–416.
147. Geist, V.; Moser, M.; Pichler, J.; Beyer, S.; Pinzger, M. Leveraging Machine Learning for Software Redocumentation. In Proceedings of the 2020 IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER), London, ON, Canada, 18–21 February 2020; pp. 622–626.
148. Geist, V.; Moser, M.; Pichler, J.; Santos, R.; Wieser, V. Leveraging machine learning for software redocumentation—A comprehensive comparison of methods in practice. *Softw. Pract. Exp.* **2020**, 1–26. [CrossRef]
149. Meloni, P.; Loi, D.; Deriu, G.; Pimentel, A.D.; Saprat, D.; Pintort, M.; Biggio, B.; Ripolles, O.; Solans, D.; Conti, F.; et al. Architecture-aware design and implementation of CNN algorithms for embedded inference: The ALOHA project. In Proceedings of the 2018 30th International Conference on Microelectronics (ICM), Sousse, Tunisia, 16–19 December 2018; pp. 52–55.
150. Meloni, P.; Loi, D.; Deriu, G.; Pimentel, A.D.; Sapra, D.; Moser, B.; Shepeleva, N.; Conti, F.; Benini, L.; Ripolles, O.; et al. ALOHA: An architectural-aware framework for deep learning at the edge. In Proceedings of the Workshop on INTelligent Embedded Systems Architectures and Applications—INTESA, Turin, Italy, 4 October 2018; ACM Press: New York, NY, USA, 2018; pp. 19–26.
151. Meloni, P.; Loi, D.; Busia, P.; Deriu, G.; Pimentel, A.D.; Sapra, D.; Stefanov, T.; Minakova, S.; Conti, F.; Benini, L.; et al. Optimization and Deployment of CNNs at the Edge: The ALOHA Experience. In Proceedings of the 16th ACM International Conference on Computing Frontiers (CF '19), Alghero, Italy, April 2019; pp. 326–332.
152. Newman, S. *Building Microservices*, 1st ed.; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2015.
153. Li, L.; Jamieson, K.; DeSalvo, G.; Rostamizadeh, A.; Talwalkar, A. Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization. *J. Mach. Learn. Res.* **2017**, *18*, 6765–6816.
154. Jacob, B.; Kligys, S.; Chen, B.; Zhu, M.; Tang, M.; Howard, A.; Adam, H.; Kalenichenko, D. Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2704–2713.
155. Pimentel, A.D.; Erbas, C.; Polstra, S. A Systematic Approach to Exploring Embedded System Architectures at Multiple Abstraction Levels. *IEEE Trans. Comput.* **2006**, *55*, 99–112. [CrossRef]
156. Masin, M.; Limonad, L.; Sela, A.; Boaz, D.; Greenberg, L.; Mashkif, N.; Rinat, R. Pluggable Analysis Viewpoints for Design Space Exploration. *Procedia Comput. Sci.* **2013**, *16*, 226 – 235. [CrossRef]
157. Meloni, P.; Capotondi, A.; Deriu, G.; Brian, M.; Conti, F.; Rossi, D.; Raffo, L.; Benini, L. NEURAghe: Exploiting CPU-FPGA Synergies for Efficient and Flexible CNN Inference Acceleration on Zynq SoCs. *CoRR* **2017**, *11*, 1–24.
158. Kumar, M.; Rossbory, M.; Moser, B.A.; Freudenthaler, B. Deriving An Optimal Noise Adding Mechanism for Privacy-Preserving Machine Learning. In Proceedings of the 3rd International Workshop on Cyber-Security and Functional Safety in Cyber-Physical (IWCFS 2019), Linz, Austria, 26–29 August 2019; Anderst-Kotsis, G., Tjoa, A.M., Khalil, I., Elloumi, M., Mashkoor, A., Sametinger, J., Larrucea, X., Fensel, A., Martinez-Gil, J., Moser, B., et al., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 108–118.
159. Kumar, M.; Rossbory, M.; Moser, B.A.; Freudenthaler, B. An optimal $(\epsilon, \delta)-$differentially private learning of distributed deep fuzzy models. *Inf. Sci.* **2021**, *546*, 87–120. [CrossRef]
160. Kumar, M.; Rossbory, M.; Moser, B.A.; Freudenthaler, B. Differentially Private Learning of Distributed Deep Models. In Proceedings of the Adjunct Publication of the 28th ACM Conference on User Modeling, Adaptation and Personalization (UMAP '20 Adjunct), Genoa, Italy, 12–18 July 2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 193–200.
161. Kumar, M.; Brunner, D.; Moser, B.A.; Freudenthaler, B. Variational Optimization of Informational Privacy. In *Database and Expert Systems Applications*; Springer International Publishing: Cham, Switzerland, 2020; pp. 32–47.

162. Gusenleitner, N.; Siedl, S.; Stübl, G.; Polleres, A.; Recski, G.; Sommer, R.; Leva, M.C.; Pichler, M.; Kopetzky, T.; Moser, B.A. Facing mental workload in AI-transformed working environments. In Proceedings of the H-WORKLOAD 2019: 3rd International Symposium on Human Mental Workload: Models and Applications, Rome, Italy, 14–15 November 2019.
163. Nickel, M.; Murphy, K.; Tresp, V.; Gabrilovich, E. A Review of Relational Machine Learning for Knowledge Graphs. *Proc. IEEE* **2016**, *104*, 11–33. [CrossRef]
164. Paulheim, H. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semant. Web* **2017**, *8*, 489–508. [CrossRef]
165. Noy, N.; Gao, Y.; Jain, A.; Narayanan, A.; Patterson, A.; Taylor, J. Industry-scale Knowledge Graphs: Lessons and Challenges. *Commun. ACM* **2019**, *62*, 36–43. [CrossRef]
166. Johnson, M.; Vera, A. No AI Is an Island: The Case for Teaming Intelligence. *AI Mag.* **2019**, *40*, 16–28. [CrossRef]
167. Cai, H.; Zheng, V.W.; Chang, K.C.C. A Comprehensive Survey of Graph Embedding: Problems, Techniques and Applications. *IEEE Trans. Knowl. Data Eng.* **2017**, *30*, 1616–1637. [CrossRef]
168. Wang, Q.; Mao, Z.; Wang, B.; Guo, L. Knowledge Graph Embedding: A Survey of Approaches and Applications. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 2724–2743. [CrossRef]
169. Li, S.; Wang, Y. Research on Interdisciplinary Characteristics: A Case Study in the Field of Artificial Intelligence. *IOP Conf. Ser. Mater. Sci. Eng.* **2019**, *677*, 052023. [CrossRef]