Artificial Intelligence Review (2024) 57:66
https://doi.org/10.1007/s10462-023-10692-0

Unlabeled learning algorithms and operations: overview and future trends in defense sector

Eduardo e Oliveira[1,2] · Marco Rodrigues[1] · João Paulo Pereira[1] · António M. Lopes[1,2] · Ivana Ilic Mestric[3] · Sandro Bjelogrlic[3]

Accepted: 30 December 2023 / Published online: 20 February 2024

Abstract
In the defense sector, artificial intelligence (AI) and machine learning (ML) have been used to analyse and decipher massive volumes of data, namely for target recognition, surveillance, threat detection and cybersecurity, autonomous vehicles and drones guidance, and language translation. However, there are key points that have been identified as barriers or challenges, especially related to data curation. For this reason, and also due to the need for quick response, the defense sector is looking for AI technologies capable of successfully processing and extracting results from huge amounts of unlabelled or very poorly labelled data. This paper presents an in-depth review of AI/ML algorithms for unsupervised or poorly supervised data, and machine learning operations (MLOps) techniques that are suitable for the defense industry. The algorithms are divided according to their nature, meaning that they either focus on techniques, or on applications. Techniques can belong to the supervision spectrum, or focus on explainability. Applications are either focused on text processing or computer vision. MLOps techniques, tools and practices are then discussed, revealing approaches and reporting experiences with the objective of declaring how to make the operationalization of ML integrated systems more efficient. Despite many contributions from several researchers and industry, further efforts are required to construct substantially robust and reliable models and supporting infrastructures for AI systems, which are reliable and suitable for the defense sector. This review brings up-to-date information regarding AI algorithms and MLOps that will be helpful for future research in the field.

Keywords Machine learning algorithms · Machine learning operations · Artificial intelligence

Abbreviations

ActiveMetaL	Active Meta-Learning recommender system
AI	Artificial Intelligence
AL	Active Learning
AutoML	Auto Machine Learning
BERT	Bidirectional Encoder Representations from Transformer
BD	Big Data
BN	Bayesian Networks

Extended author information available on the last page of the article

CAM	Class Activation Map
CEGE	Centroid Estimation with Guaranteed Efficiency
CEM	Contrastive Explanations Method
CNN	Convolutional Neural Network
CPU	Central Processing Unit
DBN	Deep Belief Networks
CRF	Conditional Random Field
DeepLIFT	Deep Learning Deep Learning Important FeaTuresmportant FeaTures
DevOps	Development and Operations
DL	Deep Learning
DQN	Deep Q-learning Network
DT	Decision Trees
FEAC-Stream	Fast Evolutionary Algorithm for Clustering data streams
GAN	Generative Adversarial Network
GCN	Graph Convolutional Networks
GPT	Generative Pre-training
GPU	Graphics Processing Unit
HMM	Hidden Markov Models
HPC	High Performance Computing
IaaS	Infrastructure as a Service
ICE	Individual Conditional Expectation
QAI	Quantum Artificial Intelligence
QML	Quantum Machine Learning
LAL	Learning Active Learning
LAL-RL	Learning Active Learning Reinforcement Learning
LIME	Local Interpretable Model-agnostic Explanations
LSTM	Long Short Term Memory
ML	Machine Learning
MLaaS	Machine Learning as a Service
MLM	Masked Language Modeling
MLOps	Machine Learning Operations
MLP	Multi-Layer Perceptron
NLP	Natural Language Processing
NSP	Next Sentence Prediction
PaaS	Platform as a Service
RBFN	Radial Basis Function Network
RBM	Restricted Boltzmann Machines
RNN	Recurrent Neural Network
RoBERTa	Robustly Optimized BERT pre-training Approach
SHAP	Shapley Additive explanations
SimCLR	Simple framework for ContrastiveLearning of visual Representations
SoET	State of the Emerging Technologies
SOM	Self-Organizing Maps
SOTA	State-of-the-Art
SVAE	Structural Variational Autoencoders
SVM	Support Vector Machine
SwAV	Swapping Assignments between multiple Views of the same image
T5	Text-To-Text Transfer Transformer
TPU	Tensor Processing Unit

VAE	Variational AutoEncoders
XLM-R	Cross-lingual Language Modeling-RoBERTa

1 Introduction

Artificial intelligence (AI) has advanced quickly during the last ten years. The recent past has seen extraordinary advances in AI theories and applications, which have had a significant impact on our daily lives. AI is a critical research topic in academia, industry and business. Scientific production is proof of these advances. A study developed by Ahmad et al. (2021) found 9734 publications, only in the Web of Science database with the contribution of 31803 researchers, in a ten-year period, between 2008 and 2017 in the field of AI. Between each of the years on which the study was based there is always a consistent increase in research output. AI is a multidisciplinary field, involving computer science, mathematics, logic, biology, psychology, philosophy, and many other disciplines, which has produced amazing results in areas like speech recognition, image processing, natural language processing, and collaborative and intelligent robotics (Zhang and Lu 2021). Beneath decision makers there is a consensus that AI represents a turning point in world history and that it is progressively evolving into corporate strategy and even to populate nations policies.

AI has evolved to a state where people directly hand problems to computers and machines learn to solve them autonomously using algorithms. Some authors such as Deng (2018) understand that this evolution took place in three waves. The first one took place in the last century in the 60's and the focus was in knowledge reasoning, mainly knowledge-based programs relying on logical reasoning implemented in rules. The second wave appeared from the 80's onward and came up with the need to add the ability to learn and handle uncertainty to knowledge-based expert systems. Instead of creating precise rules as in expert systems, in this wave, systems started to be based on statistical models and simple neural networks. Algorithms and methods, tuned by parameters obtained through training data, began to deal with uncertainty and adapt to different environments and situations. At this time, algorithms and technologies such as Bayesian Networks (BN), Decision Trees (DT), Support Vector Machines (SVM), and Random Forest were developed, and the first steps were taken in neural networks, giving rise to algorithms such as Backpropagation and Recurrent Neural Network (RNN). The third and current wave of AI began with the appearance of deep learning (DL), a slightly more than ten years ago. In traditional machine learning (ML), before data is loaded into models for predictions, data scientists or data engineers develop explicit features. DL uses many layers between the input and the output and employs neural networks as their models, being very skilled at identifying important features without the need for human involvement. So, instead of requiring feature engineering manually, DL models discover usable representations and features from the data itself (Dong et al. 2021).

With regard specifically to operations, DL operations are typically simpler to undertake than ML operations. DL brought models of simpler application and design than traditional ML models (Shao et al. 2022). Feature extraction and classifier learning are done concurrently in an end-to-end way. DL also facilitates data engineering, as building blocks or layers are directly transposable between different analysis or tasks. This former point led to the emergence of toolkits that helped to widely disseminate the methods. Hardware developments and advances in technology with dramatically increased processing speeds also

facilitated the expansion of DL. GPU acceleration enabled by parallel processing eases the training overwhelming time of DL algorithms (Shao et al. 2022).

DL can currently be considered as the core or frontier technology for pushing AI further into building smarter and more intelligent systems. The growth of DL knowledge extended the application spectrum of AI to almost all domains. Some examples are autonomous driving in automotive industry, image captioning in social media, natural language processing in call automation bots, drug discovery in pharmaceuticals, cancer research in medicine, collaborative robotics in manufacturing, colourization of b&w video in movie industry and fraud detection in banking and finance. Between the most prevalent algorithms for DL there can be found: Convolutional Neural Networks (CNN), Generative Adversarial Networks (GAN), Long Short-Term Memory Networks (LSTM), Recurrent Neural Networks (RNN), Radial Basis Function Networks (RBFN), Multilayer Perceptron's (MLP), Self-Organizing Maps (SOM), Deep Belief Networks (DBN), Restricted Boltzmann Machines (RBM) and Autoencoders.

DL has been applied in a variety of fields, including defense, to solve a wide range of problems. DL has been used in the defense industry to analyse and decipher massive volumes of data, including images and videos, to help with activities like target recognition, surveillance, threat detection and cybersecurity. Additionally, it has been applied to the development of intelligent systems for autonomous vehicles and drones as well as the improvement of language translation accuracy for military communications (Svenmarck et al. 2018).

However, there are key points that have been identified as barriers or challenges to the development of DL, especially in the defense sector, and that need to be addressed in order to guarantee a successful integration of the technology in its operations. The first challenge is related to the huge amount and variability of data to be collected and analysed. Military activities are already extremely efficient at collecting data and relying on this collection and analysis for their operations but it is necessary to ensure that this data is adapted to ML/DL processes. Simultaneously, and contrary to, for example, the industrial sector where the environment remains constant and developments can affect relatively stable topics, in the reality of the defense industry, there is a great variability of environments and situations, which makes it difficult to train algorithms and its adaptation to all scenarios.

With the exponential increase in the amount of data collected, and since DL technologies are avid of data for their training, it becomes increasingly difficult or even impossible to plan data curation work, namely the placement of labels in unlabelled databases. For this reason and also the need for quick response, the defense sector is looking for AI technologies capable of successfully processing and extracting results from huge amounts of unlabelled or very poorly labelled data.

When applying DL in defense applications, interpretability - the capacity to articulate and comprehend how a ML model makes decisions - must be taken into account. This is especially true when the model is being used to make judgements that have major implications, like in military or law enforcement operations. DL models are known to function as black boxes and have reduced ability or even inability to provide an explanation of its reasoning to the decision-maker or human operator. The majority of the present DL models are incapable of reasoning and explanation, leaving them open to catastrophic errors or attacks that they are unable to predict and so avoid.

ML and DL models in particular are vulnerable to adversarial attacks, either through manipulation of input signals or through cyber threats that can disrupt the functioning of the models and lead to inconsistent results. Although there are refined techniques to counteract this type of attack, namely in the context of cybersecurity (Nelson et al. 2022), the

development of explainable/interpretable models can help in detecting problems at this level.

Through the application of fundamental Artificial Intelligence (AI) technological enablers, defense organizations and military forces can achieve information dominance, maintain a competitive edge over adversaries, improve interoperability, and raise preparedness. But simultaneously, risk and security issues emerge that need to be addressed. Table 1 presents some of the risk and security topics arising from the use of AI technologies in the context of defense.

Taking into account the matters researched and analysed above, the Table 2 summarizes a set of techniques and applications that are expected to be the focus of research and development in a near future, in various sectors and areas, but in particular in the defense field.

The development of ML systems consists of algorithms, which, in order to provide some kind of applicability, need to be transposed to a computer infrastructure with hardware and software, thus, becoming a part of an ecosystem of integrated technologies (Ruf et al. 2021). The evolution of technologies in the field of Software Engineering over the last decade, has resulted in the emergence of different platforms and programming languages, increasing the complexity of designing information systems, creating redundancy and, as a consequence, making it difficult to automate processes related to the software lifecycle, from development to deployment. This scenario made clear the need for specialized tools to appear in the process of automating the various tasks involved in this life cycle, as well as adequate work methodologies, with the aim of making the product available more quickly, as well as reducing human intervention.

Table 1 Risk and security topics

Topic	Description
Lack of transparency in decision-making process	Alignment issues between AI system's objectives and actions and human intent could be fostered by AI complexity, aptitude for autonomous learning, and possible lack of transparency in its reasoning or explainability process
Vulnerability to adversarial attacks	Just as AI systems bring new applications in defense, on the other hand they expose a potential for AI-driven cyber deception, in which the technology might be used to carry out advanced cyberattacks
Overreliance on AI	If the outputs of AI systems are accepted without complete understanding and critical thinking, it can lead to a dependency on AI systems that, consequently, may lead to the potential degradation of traditional human knowledge systems
Backwardness of auditing procedures	All systems have their biases and distortions, and AI systems are no exception. Given the high degree of innovation and speed of implementation of these systems, they should be accompanied by the development of robust auditing and performance assessment processes
Lack of appropriate skills	If the human factor is to be present in managing AI systems in defense, it will be necessary to plan for reskilling and upskilling employees in order to get them ready for AI integration

Table 2 AI techniques and applications identified

Topic	Description
Active learning	Process of prioritising the data to be labelled to have the impact on training a supervised model
Weak supervision	Lower quality labels more efficiently and/or at higher abstraction level
Semi-supervised learning	Method in which we have input data, and a fraction of input data is labelled as the output
Zero-shot learning	Using classes that were not observed during training -> predict the class that they belong
Self-supervised learning	The model trains itself to learn one part of the input from another part of the input
Clustering	Task of grouping a set of objects in such a way that objects in the same group are more similar
Dimensionality reduction	Transformation of data from a high-dimensional space into a low-dimensional space so that the low-dimensional representation retains some meaningful properties of the original data
Explainability/interpretability	The ability to explain and understand the decision-making processes of a DL model
Text processing	Process of transforming unstructured text into structured to identify meaningful patterns and new insights
Computer vision	Derive meaningful information from digital images, videos and other visual inputs

This concept, which involves software tools, but also processes and mindset, is called Development and Operations (DevOps) (Azad 2022), and, being a ML system composed of hardware and software, but whose integration with the rest of the ecosystem is done at the software level, the need to automate the development and availability of ML models integrated into information systems naturally appeared with the designation of machine learning operations (MLOps) (ML + DevOps) (Mäkinen et al. 2021), thus creating an extra layer of complexity.

A MLOps architecture is based on a workflow, which, not being a fixed set of steps, is typically characterized by engineering areas, tasks and agents that are generally common to most of the scenarios. In Subramanya et al. (2022a), a case study of an MLOps pipeline with the aim of automating the energy price forecasting process using artificial neural networks is demonstrated, as well as a case study to automate energy consumption forecasting. In both, the forecast result is also integrated into an independent software tool. The process of automating the availability of ML models integrated in a software ecosystem represents an advance in the synergies between several fields, Data Engineering, ML and Software Engineering, however, evolution comes with challenges (John et al. 2021), such as lack of appropriate tools to ensure data quality, high computational costs, or the necessary synergy between development and operations teams ensuring technological compatibility between the entire stack used, and the future of MLOps involves solving them, thus smoothing the entire process.

At the same time that AI and ML activities are gaining importance and preponderance in institutions, so are their management and automation needs. In the same way that the evolution of DevOps took place in past years, an accelerated evolution in MLOps is also expected. At the same time that MLOps investigation is being carried out in the research community, tools are appearing on the market that target workflows for model deployment

and administration, ensuring that they are completely within the control of the team and that bottlenecks are removed.

Finally, there has been a trend towards the evolution of AI Cloud services which is expected to become more pronounced in the near future. These services bring promises of access to high performance systems without the necessary investments and maintenance costs, associated with an offer of advanced tools, easy parametrization and constant evolution. However, reservations are raised about rising costs and confidentiality issues.

Taking into account the aforementioned topics, in the short and medium term, future trends and technology moves will continue to focus on the development of DL, sustaining the current wave, while awaiting, in a longer term, more disruptive developments, in models and support hardware that can constitute a base for the generation of a new, not yet foreseen, (fourth) wave of AI. The Fig. 1 shows the evolution of AI in its three main waves and presents the central topics of focus in each of them. At the same time, it summarizes the main future trends for the upcoming years.

Searching quality scientific sources of information was the main tool used to gather relevant technical and technological information to carry on this research work. Taking into account the objective of looking at future technologies in AI, the initial searches were limited to the years 2020 onwards, with references with earlier dates only considered after being indirectly identified in initial searches and considered relevant by the team. When searches resulted in large numbers of articles, which was expected given the relevance of the topic in the international scientific community, these were ranked by number of citations, or by number of citations per year, since more recent articles have fewer citations.

Information, data and data science play an important role in modern military operations. Defense organizations are on their way to adopting Data Science and Artificial Intelligence activities and technologies. The work developed involved a research to investigate and evaluate current and future relevant AI technological enablers to be deployed in the short and medium term. Briefly, the analysis was based on three main areas:

a. Future AI/ML platform development (including public cloud computing but also including on-premise solutions);

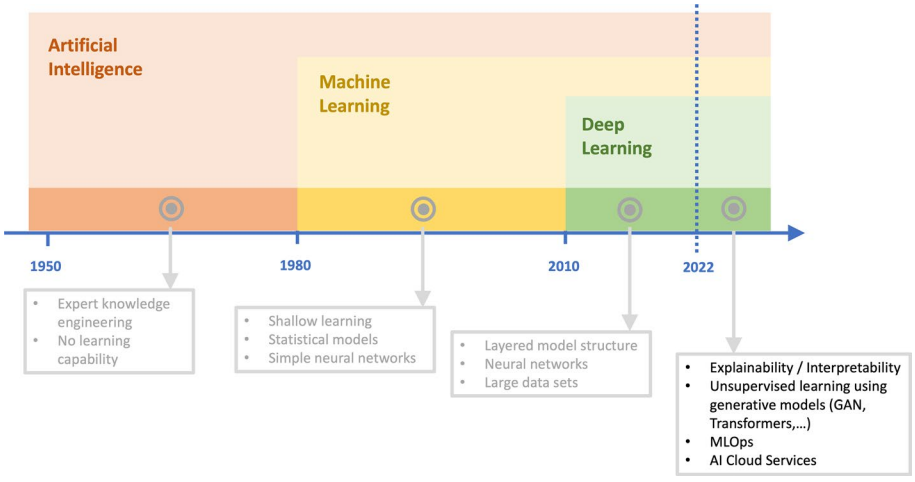

Fig. 1 Main waves and future trends of AI

b. Future approaches and best practices in ML Ops, including procedures and skills to facilitate the effective ML operations; and
c. Technology that addresses ML-specific issues related to the lack of labelled data and explainability of ML results, contributing to:

    i. A discussion of upcoming AI technologies and the promise they may hold. The prospective effect on a defense organization for technology seen to be promising solutions;
    ii. Proposed use scenarios that make use of cutting-edge technological solutions that are pertinent and that can show the advantages.

This paper presents an overview of AI/ML algorithms for unsupervised or poorly supervised data, and MLOps techniques applicable in the defense industry. The presented algorithms are divided according to their nature, focusing on techniques or on applications. Techniques can belong to the supervision spectrum or address explainability. Applications are either focused on text processing or computer vision. Then, MLOps techniques are presented and discussed, and cases are reported. The overview brings information regarding AI algorithms and MLOps that is expected to help further research in the field.

The paper is organized into four main sections: Sect. 2 presents the AI/ML algorithms and discusses their characteristics and applications. Deep generative models are firstly introduced, namely variational autoencoders, transformers and diffusion models, followed by contrastive methods. Afterwards, AI/ML algorithms focused on explainability are addressed. Additional AI/ML techniques are also presented, specifically active learning, semi-supervised learning, weak supervision, zero-shot learning and clustering & dimensionality reduction, along with diverse applications, such as text processing and computer vision. The Section ends with a discussion about limitations and opportunities involving AI/ML algorithms. Section 3 addresses MLOps techniques, their limitations and common concerns regarding their utilization. It starts by presenting the MLOps typical workflow, involved challenges and some usual software tools. Then, AI/ML cloud trends are debated, namely ML as a service, cloud versus local solutions and hybrid clouds, along with some expected future developments. Afterwards, issues regarding the selection of appropriated MLOps tools are discussed and use-cases are addressed, such as the bioinformatics application with Kubeflow for batch processing in clouds, the MLOps scaling ML lifecycle in an industrial setting, and the training and serving ML workloads with Kubeflow at CERN. A discussion about limitations and opportunities involving MLOps concludes the Section. Finally, Sect. 4 outlines the main conclusions of the paper, highlighting some limitations of the present review work and pointing out new paths for future research.

2 AI/ML algorithms

In this Section, we discuss the AI/ML algorithms for unsupervised or poorly supervised tasks. These algorithms can be divided according to their nature, meaning that they either focus on techniques, or on applications, as is illustrated in Fig. 2. Algorithms in the techniques group focus more on their technical characteristics, while the algorithms in the applications group focus more on the nature of tasks the algorithms are applied to. Techniques can belong to the supervision spectrum, or focus on explainability. Applications are either focused on text processing or computer vision.

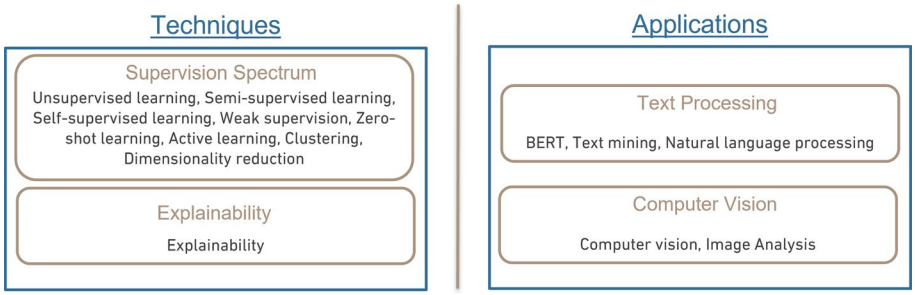

Fig. 2 Algorithms division according to their nature. Techniques can belong to the supervision spectrum, or focus on explainability. Applications are either focused on text processing or computer vision

While the division of algorithms between explainability, text processing and computer vision are relatively easy to understand, it is necessary to explain the concept of supervision spectrum. Upon reading the scientific literature, we came to the realization that the crisp division between different groups of techniques/tasks (e.g., unsupervised, supervised, semi-supervised) is artificial, as there are several techniques that can be used in different tasks (e.g., most deep generative models may be used in semi-supervised, self-supervised, unsupervised and fully supervised tasks). As such, we propose a supervision spectrum, which consists in positioning the techniques along two axes.

The first axis focuses on the objective of the techniques, in the sense of how many relations they are trying to discover. These relations occur between the variables of the dataset, where n is the number of existing variables, and m is a subset. The techniques can belong to up to three levels:

- Single label prediction [n → 1]
- Multi-label prediction [n → m]
- Clustering/Association [n → n]

The single label prediction focuses on the most common task in ML, where the objective is to relate all the independent variables to a single target variable. The multi-label prediction focuses on relating all the independent to another set of objective variables, which are different and smaller than the independent variable set. The clustering/association level is closely related to unsupervised tasks, where the objective is to find relations among all the variables, in order to group similar instances. This means that we are attempting to find relations between all the sets of variables.

The second axis focuses on the information gathering strategy of the techniques. This axis focuses the availability of information about the dependent variables (i.e., the labels), and if it is possible to obtain such information, and how to do so. Techniques can belong to up to five levels:

- Determinate labels [regular classification]
- On-demand labels [active learning]
- Uncertain labels [weak learning]
- Transferable labels [zero-shot learning]
- Automatic labels [self-supervised learning]

On one side of the axis, the determinate labels level reports to the standard supervised ML techniques, where the labels are known for all instances, and are assumed to be correct. The on-demand label level allows for the algorithms to ask an oracle (i.e., human user) to label some unlabelled instances. The uncertain labels level refers to situations when either we do not have the labels for all instance, or when their value is not certain. This is the focus of weak learning techniques. Techniques in the transferable labels level (zero-shot learning) focus on being able to transfer knowledge learned in one domain (with a set of labels), to another domain (with a different set of labels), using minimal knowledge about the relations between labels, usually provided by human users. This way, we may use a previous algorithms on unlabelled datasets. In the automatic labels level, techniques assume that there are no labels, or it is impossible to know anything about them, and as such will either try to group similar instances, or generate pseudo-labels from the characteristics of the dataset (as in unsupervised and self-supervised learning techniques).

We can then visualize the different groups of the techniques in the above mentioned axes in Fig. 3.

We have identified a total of 32 algorithms that represent the state-of-the-art (SOTA) in the domain of techniques and strategies to deal with unlabelled or poorly labelled data. This can be placed on the spectrum as seen in Fig. 4 (some algorithms appear more than once, in case they can perform different tasks).

Of the literature review, it was possible to identify three big trends that are more likely the provide future developments relevant to the scope of this report. These trends are: Deep Generative Models (DGM), contrastive methods, and explainability.

In the remaining of the Section we will describe each of the 32 algorithms identified as SOTA, dividing them according to the trend they belong to, and to the need they fulfil. We first describe the algorithms that belong to the three big trends, each corresponding to a Subsection, namely Sect. 2.1 to DGM, Sect. 2.2 to contrastive methods, and Sect. 2.3 to explainability. Then, we describe the algorithms that report to other technique needs in Sect. 2.4. We then discuss the needs related to applications. To summarize the findings, we present a discussion connecting the needs to the algorithms. Finally, we conclude by considering what are the most likely developments in the near future.

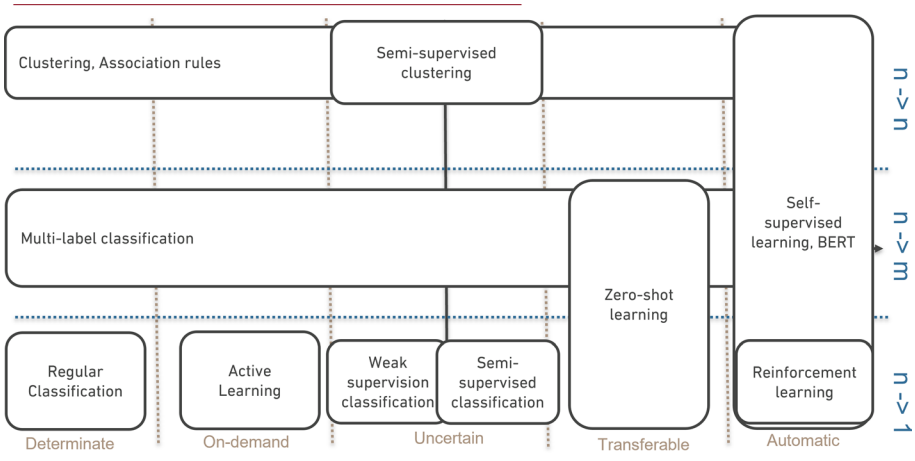

Fig. 3 Supervision spectrum with the techniques groups identified. The horizontal axis reflects the information gathering strategy, while the vertical axis reflects the objective/number of relation

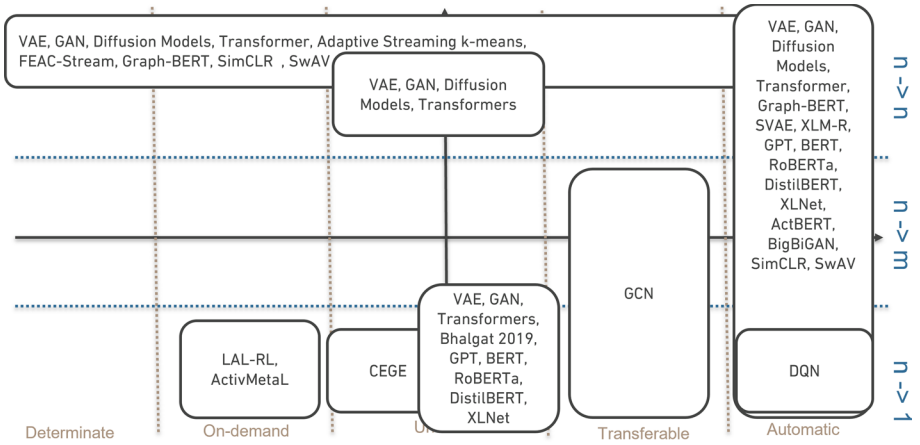

Fig. 4 Supervision spectrum with the 32 algorithms identified as SOTA. The horizontal axis reflects the information gathering strategy, while the vertical axis reflects the objective/number of relation

2.1 Deep generative models

Most ML models are said to be discriminative, as their focus on distinguishing between different labels and instances, and use their discriminative power to provide good predictions and clusters (Murphy 2012). Examples of discriminative models are DT and SVM. However, a more flexible approach is to develop generative models, which attempt to discover the underlying distribution that produces the data. This allows the models to be used to generate new data. Examples of generative models are BN and Hidden Markov Models (HMM).

Most recent developments in using ML to handle unlabelled or poorly labelled data focus precisely on the use of generative models, due to their flexibility that allows them to be used in many tasks, and ability discover the underlying distributions to generate artificial data and labels. In particular, attention has focused on the use of deep generative model, which combine deep neural network architectures with the aim of obtaining generative models. Four main families of algorithms have been used in the literature, representing the SOTA performances in the fields of text processing and computer vision: GAN, Variational Auto Encoders (VAE), Transformers, and Diffusion Models.

2.1.1 Generative adversarial networks

GAN have been proposed in Goodfellow et al. (2014), and are a deep generative model that frames the problem as a supervised learning one with two parts: the generator model that generates new (false) examples, and the discriminator model that tries to classify them between real and false. By training both adversarial models together, their performance is improved, and the resulting latent space possesses both good performance and capacity for generalisation. They may be used together with CNN to analyse image data. New data generation may be conditioned by certain features (e.g., pictures of men), which makes them semi-supervised (van Engelen and Hoos 2020). May be used for data augmentation, which makes them useful for self-supervised tasks (Brownlee 2019).

Advantages of GAN algorithms are: their flexibility; their ability to be combined with other algorithms for different tasks; they do not require labels; they can achieve very good performance (SOTA level); and they have a strong feature extraction and sample generation power (Ren et al. 2022).

In terms of disadvantages, GAN models require large amounts of data, and are hard to train due to the convergence not being ensured, which leads to unstable training.

A specific implementation of GAN that revealed itself as relevant in the literature is Large Scale Adversarial Representation Learning (BigBiGAN) (Donahue and Simonyan 2019). BigBiGAN is a method for image generation and representation learning. This method allows for the extraction of features in an unsupervised way from a GAN and scales up the previously existing algorithms leading to an improved GAN model. BigBiGAN, builds upon the SOTA BigGAN model (Donahue et al. 2016), extending it to representation learning by adding an encoder and modifying the discriminator.

2.1.2 Variational autoencoders

VAE (Kingma and Welling 2013) are deep generative models with an autoencoder architecture that regularizes ("simplifies") the distribution of the encodings in order to avoid overfitting, and make the latent space more robust and generalisable. Each instance is encoded as a distribution in the latent space instead of a single point. Regularisation occurs both locally and globally, to ensure completeness (meaningful post-decoding) and continuity (proximity=similarity) in the latent space. In the latent space, clusters may be formed naturally. It may be used in pretext tasks in self-supervised learning, and may use labels in a semi-supervised learning process (VAE are naturally unsupervised (van Engelen and Hoos 2020)).

VAE has as advantages more control over latent space modelling than other autoencoder based algorithms, it is flexible and can be combined with other algorithms for different tasks as other deep generative models. It does not require labels and can achieve very good performance. As it happened with GAN, it has strong feature extraction and sample generation power (Ren et al. 2022).

As disadvantages, VAE require large amounts of of data, are more theoretically complex than GAN. It is also difficult to model good loss function, as the resulting marginal likelihood is intractable to compute and optimize, and thus it is learned learned by optimizing a tractable "evidence lower bound", obtained using a tunable inference distribution (Zhao et al. 2017). This can lead to the generation of blurry pictures (Dosovitskiy and Brox 2016). In Zhao et al. (2017), a variation of VAE is proposed to tackle this issue.

2.1.3 Transformers

Transformers are generative models with an encoder/decoder architecture, consisting of several attention-like layers. The aim is to transform a sequence into another sequence, while using attention mechanisms to focus on the important parts of the sequences, without using recurrent networks. The attention mechanism can be parallelized. It also uses self-attention to determine relations between elements of the same sequence. This way, context is perceived more efficiently than using recurrent networks. During training, the focus is on predicting the next element in a sentence, in order to avoid overfitting. For the same reason, masks are used to hide the following elements of the sequence. Can be used for forecasting tasks and predicting protein unfolding structures. The Bidirectional Encoder

Representations from Transformers (BERT), Generative Pre-training (GPT), and XLNet are a particular types of transformers. Transformers have yielded many SOTA results in some natural language processing (NLP) applications (Acheampong et al. 2021).

Compared to other deep generative models, transformers provide better context analysis, and can be parallelized, which makes them faster. However, attention mechanisms can only deal with fixed-length text strings, which makes it necessary to separate long sequences into smaller ones, and this can cause context fragmentation.

The BERT (Devlin et al. 2018) is the most popular transformer application. It uses transformer's encoders as pre-trained models for NLP tasks such as Question Answering and Text Summarization. BERT's performs these tasks in two phases: 1) pre-training for language understanding, and 2) fine-tuning for a specific task. BERT can understand language by training on the Masked Language Modelling (MLM) and the Next Sentence Prediction (NSP) mechanisms. It takes as input some random sentences, masks some of the words in the sentences, and reconstructs the masked words from the surrounding texts at the output. Its ability to input two sentences at once and determine if the second sentence comes after the first makes it achieve NSP. This ability helps the model to maintain long-distance relationships between texts. After pre-training, the model is then trained on a NLP task by performing supervised training on a dataset and replacing the BERT's fully connected output with a new set of output layers. The BERT model trains faster since the other model parameters are only fine-tuned aside from the output parameters learned from scratch.

It conserves the advantages of transformers, and can be applied to a wide range of language modelling applications. In terms of disadvantages, it is limited to monolingual classifications, is limited by the sequence size, and suffers from pragmatic inference (Acheampong et al. 2021). Some variations can be computationally expensive.

The Robustly Optimized BERT pre-training Approach (RoBERTa) (Liu et al. 2019) is a BERT variant that seeks to ultimately optimize BERT by tweaking various methodological parameters in the initial version of BERT. Its main advantage is that the more massive pre-training data used yields better performance in a variety of tasks. It outperforms XLNet and BERT in downstream self-supervised NLP tasks (Acheampong et al. 2021). In terms of disadvantages, the massive pre-training makes the method resource intensive, which increases computational complexity.

DistilBERT (Sanh et al. 2019), in an opposite direction than RoBERTa, simplifies the initial architecture of BERT, by reducing the number of layers in the BERT-base model by a factor of 2, removing token embeddings and poolers in order to yield a much smaller and faster version of BERT for general-purpose use. DistilBERT is capable of language modelling and can be pre-trained on other language modelling tasks. It is faster and lighter than original BERT, but still suffers from context fragmentation (Acheampong et al. 2021).

XLNet (Yang et al. 2019) is an auto-regressive language model that utilizes the concepts of the Permutation Language Model (PLM) and the Transformer-XL model to achieve the SOTA. As a BERT variant, the significant difference between the BERT and the XLNet has to do with their training objective. XLNet uses the permutation objective, whereas BERT masks the data and tries to predict the masked data using a bi-directional context. Its main advantages are the ability to extract contextual information due to the PLM implementation in the model, the capacity to perform better than BERT in a broader range of language modelling applications, and, most importantly, fixing BERT's fixed-length limitation (Acheampong et al. 2021). However, these advantages come with an increased computational costs.

The GPT (Radford and Narasimhan 2018) leverages the semi-supervised learning approach to model language using transformer decoders. Mainly used for text

representation, the GPT is made up of 12 transformer layers, and 12 attention heads transformer decoder that uses the massive unlabelled datasets through pre-training and fine-tunes them on the limited supervised datasets. The sole task of the GPT is to predict the next token in the sequence. GPT-2 was designed to predict the next sentence in sentences and establishes that language models can learn tasks without direct supervision. The GPT-3 model scales up on the GPT-2 model even further with 175 billion trainable parameters. Its model architecture is the same as that of the GPT-2 except that the transformer layers have alternating dense and locally banded sparse attention patterns. GPT-4 most likely uses 100 Trillion parameters (Karhade 2022). ChatGPT (OpenAI 2022) has recently become the main reference in terms of text generation.

The main advantages of the model are an improved lexical robustness when GPT is applied, and the pre-trained model ability to be fine-tuned to perform other tasks without model customization. it outperforms various models trained on domain-specific datasets and produces SOTA results on a diverse range of domain-specific language modeling tasks (Acheampong et al. 2021). In terms of disadvantages, its resource-intensive nature makes the pre-training step expensive, and it cannot model dependencies longer than designated fixed lengths.

Cross-lingual Language Modeling-RoBERTa (XLM-R) (Conneau et al. 2019), developed by Facebook, uses self-supervised training techniques (XLM and RoBERTa) to achieve SOTA performance in cross-lingual understanding. It improves upon previous multilingual approaches by incorporating more training data and languages. It has SOTA performance in cross-lingual understanding and is the first multilingual model to outperform monolingual ones (Meta 2019). However, it is specifically focused on cross-lingual tasks.

2.1.4 Diffusion models

Diffusion models (Sohl-Dickstein et al. 2015) are generative models that use a Markov chain of diffusion steps to introduce random noise in the data, and then learn to reverse the diffusion process. The main intuition is that, if we build a learning model that can learn the systematic decay of information due to noise, then it should be possible to reverse the process and therefore, recover the information back from the noise. It is similar to VAE but, instead of learning the data distribution, the system's objective is to model a series of noise distributions. Albeit more time consuming, they outperform other models in certain tasks, such as text and image synthesis (Siddiqui 2022). An example of diffusion model application is image generation app DALL-E2 (Open AI 2022). Other examples may be consulted on this blog post (Awan 2022). Croitoru et al. (2022) presents a literature review on the application of diffusion models to computer vision.

The main advantages of diffusion models are their flexibility, SOTA performance, and tractability (since noise is added incrementally). Their main disadvantage is that the improved performance comes at the cost of increased computation time and sampling time, caused by the use of multiple denoising steps (Dhariwal and Nichol 2021).

2.2 Contrastive methods

Contrastive methods are usually considered part of the self-supervised group. Their central idea is to replicate a specific characteristic of how humans learn: to learn a new concept, humans compare it with other concepts they already know, and focus on learning in which

ways the new concept is similar from the previous ones, and in which way it is different. Contrastive methods have been used as a self-supervised technique in computer vision for image clustering (e.g., Swapping Assignments between multiple Views of the same image (SwaV) and Simple framework for contrastive learning of visual representations (SimCLR)), and to enhance the explainability of models (e.g., contrastive explanations method (CEM)).

A self-supervised approach to learn features by SwAV (Caron et al. 2020) combines online clustering with contrastive learning. It uses an online clustering mechanism to learn better representations by grouping similar features together by comparing representations with cluster centroids. The objective is not only to make the positive pairs of samples close to each other but also, to make sure that all other features that are similar to each other club together. SwaV uses a swapped prediction mechanism where we predict the cluster assignment of a view from the representation of another view. It was developed by Facebook. Its main advantage are a good performance and fast convergence, achieves SOTA when trained in the small-batch setting, with fewer epochs, and by using multi-crop argumentation (Ohri and Kumar 2021). However, it is very resource intensive, and still cannot surpass supervised algorithms' performance.

The SimCLR (Chen et al. 2020) is a scheme that is the base of many recent contrastive learning schemes. SimCLR adopts contrastive learning that attempts to attract different augmented views of the same image and repel augmented views coming from other images. The authors discovered that using larger batch sizes, larger backbones and large epochs during training significantly improves performance (Ohri and Kumar 2021). It is developed by Google.

SimCLR has good performance, specially with large batch size during training. However, it has slightly worse performance than SwAV, while remaining resource intensive. Batch size also limits the number of negative examples that can be included during training, which in turn may affect the capacity of the model to distinguish concepts.

In what concerns data augmentation strategies in contrastive self-supervised techniques, Ohri and Kumar (2021) presents a discussion where the main conclusions are:

- SwAV authors claim their multi-crop augmentation strategy is generic and may be implemented in other methods for enhanced performance
- Color distortion is very relevant, as claimed by SimCLR authors
- A well-tuned augmentation strategy is critical for the success of contrastive methods in computer vision

In the intersection between contrastive methods and explainability, the CEM (Dhurandhar et al. 2018) is capable of generating, contrastive explanations for any black box model. More specifically, given any input and its corresponding prediction, the method can identify not only which features should be minimally and sufficiently present for that specific prediction to be produced, but also which features what should be minimally and necessarily absent.

2.3 Explainability

An emerging trend in AI/ML is the development of explainability/interpretability strategies, as most recent developments are heavily focused on performance, in detriment of transparency and understanding. This can be an issue in high-risk situation, where it is

required to monitor to the performance and viability of the deployed models. As such, several methods have been developed for understanding and explaining model behaviour. Explainability methods have mostly focused on either model-agnostic methods (work with any model), or methods specific to certain tasks, in particular computer vision and identifying the most relevant pixels.

Focusing first on computer vision, Deep Learning Important FeaTures (DeepLIFT) (Shrikumar et al. 2017) is a popular algorithm that was designed to be applied on top of deep neural network predictions. The method's superiority was demonstrated by showing considerable benefits over gradient based methods when applied to models that were trained on natural images and genomics data (Linardatos et al. 2021). By observing the activation of each neuron, it assigns them contribution scores, calculated by comparing the difference of the output from some reference output to the differences of the inputs from their reference inputs. By optionally giving separate consideration to positive and negative contributions, DeepLIFT can also reveal dependencies that are missed by other approaches. It is a fast and exact method, but it requires careful understanding of the DL models where it is used.

Another method focused on computer vision the Class Activation Maps (CAM) algorithms (Zhou et al. 2015), and they are used for CNNs. More specifically, they are used to indicate the discriminative regions of an image used by a CNN to identify the category of the image. Grad-CAM (Selvaraju et al. 2019) is a strict generalization of CAM that can produce visual explanations for any CNN, regardless of its architecture, thus overcoming one of the limitations of CAM. Grad-CAM++ (Chattopadhay et al. 2018) is an extension of the Grad-CAM method that provides better visual explanations of CNN model predictions. More specifically, object localization is extended to multiple object instances in a single image while using a weighted combination of the positive partial derivatives of the last convolutional layer feature maps with respect to a specific class score as weights to generate a visual explanation for the corresponding class label. These methods can be used to identify failure modes of the model, and possess high-resolution and are highly class-discriminative (Linardatos et al. 2021). However, they can only be used with CNN.

Moving to model-agnostic explainability methods, the Local Interpretable Model-agnostic Explanations (LIME) (Ribeiro et al. 2016) can generate interpretations for single prediction scores produced by any classifier. For any given instance and its corresponding prediction, simulated randomly-sampled data around the neighbourhood of input instance, for which the prediction was produced, are generated. Subsequently, while using the model in question, new predictions are made for generated instances and weighted by their proximity to the input instance. Lastly, a simple interpretable model, such as a DT, is trained on this newly-created dataset of perturbed instances. LIME has the following advantages (Molnar 2022): works with any black-box model; it is faster than Shapley Additive explanations (SHAP); explanations are short and contrastive; works with tables, images and text; can use different features than the ones the model was trained on (embeddings vs. original words). In terms of disadvantages, we can list Molnar (2022): poor choices in terms of parameters could lead LIME to missing out on important features; only provides local explanations; neighbourhood definition, sampling process are unsolved problems; complexity must be defined *a priori*; Explanations may be unstable, and can be manipulated.

The SHAP (Lundberg and Lee 2017) is a method inspired in game-theory that aims to enhance interpretability through the computation of the importance values for each features for individual predictions. There are several methods for computing the shapley values. It has three desirable properties: local accuracy, missingness (missing features get a value of 0), consistency (if the relative importance of a feature changes because of

a model change, that is reflected on the SHAP values). Examples of SHAP applications are: Mosca et al. (2022), which uses the sentence structure to enhance interpretability; Le et al. (2022b), which combines BERT and SHAP to improve DNA analysis; Rizinski et al. (2022), which uses SHAP to promote ethically responsible ML in Fintech. SHAP's advantages are (Molnar 2022): solid theoretical foundation; has a fast implementation with tree-based models; it is possible to get a global interpretation of the model by computing several SHAP values. On the other hand, it presents as disadvantages (Molnar 2022): its kernel version is slow and ignores feature dependence; its tree-based version can produce some non-intuitive feature attributions; it is possible to create intentionally misleading interpretations.

The Anchors method (Ribeiro et al. 2018) is a model agnostic method that explains individual predictions by finding a decision rule that "anchors" the prediction sufficiently. A rule is said to anchor a prediction if any changes in other feature values do not affect the prediction. Anchors uses reinforcement learning techniques and a graph search algorithm to reduce the number of model calls to a minimum while still being able to avoid falling in local optima issues. Anchors main advantages are its capacity to work in non-linear and complex prediction frontiers, and its efficiency and capacity to parallelize (Molnar 2022). The disadvantages surge from it being highly configurable, which makes it sensitive to hyper-parameters. It also requires discretization in many scenarios, may require many calls to model, and its coverage undefined in some domains (Molnar 2022).

More of a graphical method, Individual Conditional Expectation (ICE) plots (Goldstein et al. 2013) is a model agnostic interpretability method for supervised algorithms. Each plot illustrates how each instance prediction changes when a feature changes. The new change in the predictions for each instance can be computed by fixing the values of all the other features, creating variants of this instance by replacing the selected feature's value with others, and thus making the changed predictions for the new, artificial instances (Linardatos et al. 2021). It is an intuitive method that can uncover heterogeneous relationships. However, it can only display a single feature at once; If many instances are plotted, it can become overcrowded; Some points in the lines may be invalid.

One alternative to enhance the explainability of DL models is combining them with graphical models (Barredo Arrieta et al. 2020). An example of such methods is Structural Variational Autoencoders (SVAE) (Johnson et al. 2016). SVAE mixes probabilistic graphical models with DL to get the tractability of the first with the flexibility of the latter. The main idea is to use a Conditional Random Field (CRF) variational family. The models learns recognition networks that output potential graphical models instead of outputting the complete variational distribution's parameters directly. These potential models are then used in graphical model inference algorithms in place of the observation likelihoods. SVAE provides a rich latent representations, and enables fast variational inference, while enhancing interpretability. Other combination alternative to enhance interpretability is presented in Chen et al. (2023), who combines Bayesian estimators and interactive features with ML models to improve predictive performance, in a way that these models become transparent to understand.

One note should be included for the relationship between DGM (as detailed in Sect. 2.1) and explainability. Indeed, knowing the generating process of data, as DGM allow for, is helpful when explaining data. However, DGM are not inherently interpretable, and can greatly benefit from explainability methods. The use of both methods can lead to a greater knowledge on how models work, and how certain data came to be.

2.4 Other techniques

In this Subsection, we will provide some examples of algorithms that do not fit directly into the main trends, but represent significant development for the unsupervised or poorly supervised techniques identified, namely active learning, semi-supervised learning, weak supervision, and clustering & dimensional reduction.

2.4.1 Active learning

Active meta-learning recommender system (ActivMetaL) (Sun-Hosoya et al. 2018) keeps the scores of multiple active learning (AL) approaches on given tasks in a sparse matrix, predicts the performance of each approach for a new task, and then fills the corresponding row of the matrix for this task. By doing so, the process predicts which algorithm will perform best for the new task/dataset. ActiveMetaL is capable of generalizing across learning task, but is limited to specific domains.

Learning Active Learning (LAL) is an AL process that formulates learns how to predict the reduction in the expected generalization error when we add a new label to the training set. Learning Active Learning Reinforcement (LAL-RL) (Konyushkova et al. 2017) defines AL as a Markov Decision Process to find the optimal and general-purpose strategy. LAL-RL is independent of the dataset and ML classifier (contrarily to LAL that is designed for Random Forest), and its objective does not depend on a specific performance measure. LAL-RL has very good performance, but does not handle highly unbalanced datasets well, being sensitive to the choice of evaluation metric, and requiring optimization of many hyper-parameters (Sayin et al. 2021).

For a review on the combination of DL and active learning, please refer to Ren et al. (2020).

2.4.2 Semi-supervised learning

Although most deep generative learning algorithms work in semi-supervised tasks, we discuss here a strategy that had relevance in the semi-supervised learning, in particular in co-training (Ning et al. 2021).

Bhalgat et al. (2019) introduces a new paradigm for tri-training, mimicking the real world teacher-student learning process. The proposed adaptive teacher-student thresholds used in the proposed method provides more control over the learning process with higher label quality. Experimental results show that the proposed method outperforms other strong semi-supervised baselines, while requiring less number of labelled training samples. However, it was only tested on sentiment analysis task.

2.4.3 Weak supervision

In terms of weak supervision, Poyiadzi et al. (2022) presents a framework that structures the different existing weak supervision strategies.

The most recent paper focusing on weak supervision is Gong et al. (2022), presenting Centroid estimation with guaranteed efficiency (CEGE). It is a method for weakly supervised learning with incomplete, inexact, and inaccurate supervision. The main idea of the framework is to use an unbiased and statistically efficient risk estimator that is applicable

to various weak supervision. By decomposing the loss function into a label-independent term and a label-dependent term, it is found out that only the latter is influenced by the weak supervision and is related to the centroid of the entire dataset. Focuses on statistical efficiency, in addition to estimation unbiasedness. CEGE can cover many weak supervised problems, and accommodate many loss functions, and can use SVM and neural networks for classification. However, it was only tested in 4 weak supervised tasks.

2.4.4 Zero-shot learning

The most recent review about zero-shot learning is Wang et al. (2019).

One of the most relevant developments in the area of zero-shot learning are Graph Convolutional Networks (GCN) (Ullah et al. 2022). GCN are a version of CNN where the input are graphs, and the objective is node classification. GCNs can be categorized into 2 major algorithms, Spatial Graph Convolutional Networks and Spectral Graph Convolutional Networks. In GCNs, the relationship among different classes can be represented in the graph, and this information could be used in large-scale zero-shot learning. In some applications, the training instances and semantic information may become available in an online manner. In this scenario, the ability to utilize the sequentially available data is needed. They can be used in online situations, with streaming data, but are not applicable to directed graphs, and cannot handle very dense graphs.

2.4.5 Clustering and dimensional reduction

A major research stream on clustering and dimensional reduction outside of deep generative models is the use of online clustering, to enable the completion of this type of task in Big Data (BD) context. Zubaroğlu and Atalay (2021) presents a survey on online clustering. Two of the most relevant development are Adaptive Streaming k-means, and Fast evolutionary algorithm for clustering data streams (FEAC-Stream).

Adaptive streaming k-means (Zhang et al. 2017) is an online, partitioning based data stream clustering algorithm. Overcomes the problems of required parametrization and adaptation to concept drift. The algorithm is composed of two main phases, which are initialization phase and continuous clustering phase. For concept drift detection, standard deviation and mean of the input data are stored during the execution. The algorithm tracks how these two values change over time and predicts a concept drift according to the change. It has good performance, allowing for fully online clustering with low parametrization effort. However, it can only detect hyper-spherical clusters (Zubaroğlu and Atalay 2021).

The FEAC-Stream (de Andrade et al. 2017) is an evolutionary algorithm for clustering data streams with a variable number of clusters. FEAC-Stream is a k-means based algorithm, which estimates k automatically using an evolutionary algorithm. Being fully online, FEAC-Stream does not store synopsis of the data, instead maintains the final clustering result. It is also a fully online algorithm with good performance that is also only able to detect hyper-spherical clusters, but the clustering quality is dependent on user defined parameters.

In addition to table data, text and images, we may also be interested in clustering graphs. Graph-BERT (Zhang et al. 2020) is a graph neural network based only on the attention mechanism. The method is fed with sub-graphs instead of the whole graph. Pre-trained Graph-BERT models may be transferred to other applications. Used in node attribute

reconstruction and structure recovery tasks, node classification and graph clustering. It can effectively learn the representations of the target node, serve as the graph representation learning component in graph learning pipeline, and the models may be transferred and applied to address new tasks. However, it may lose global context in larger graphs.

2.5 Other applications

In this Subsection, we will provide some examples of algorithms that do not fit directly into the main trends, but represent significant development in terms of areas of application, namely text processing and computer vision.

2.5.1 Text processing

Most recent developments in the field of text processing have been through the use of deep generative models, in particular transformers. However, there has been a development focused on transfer learning that is worth mentioning.

Text-To-Text Transfer Transformer (T5) (Raffel et al. 2019; Research 2020) proposes reframing all NLP tasks into a unified text-to-text-format where the input and output are always text strings, in contrast to BERT-style models that can only output either a class label or a span of the input.T5 allows the use of the same model, loss function, and hyperparameters on any NLP task, including machine translation, document summarization, question answering, and classification tasks (e.g., sentiment analysis). It results from a large-scale empirical survey conducted by Google on transfer learning applied to text processing. T5 is applicable to many NLP tasks, and achieves SOTA performance in many of them. However, this comes at the cost of a model with considerable size compared to others.

2.5.2 Computer vision

Considerable advances have been made in the field of computer vision, especially with deep generative models, contrastive methods, and explainability.

A different development was made using reinforcement learning. Deep Q-learning (DQN) (Mnih et al. 2015) is the most famous deep reinforcement learning model which learns policies directly from high-dimensional inputs by CNNs. DQN use a deep neural network to compute the reward for each action. Examples of application are: robotics, autonomous vehicles, faster video loading (Le et al. 2022a). DQN can be used in multiple tasks, but defining possible actions provides limited freedom, and may not be sufficient to tackle reality. Reinforcement learning also requires many observations for training.

Some advances attempt to bridge the gap between text processing and computer vision. A recent example is ActBERT (Zhu and Yang 2020). ActBERT is a method to learn video-text pairs in a self-supervised way, based on the BERT architecture (Ohri and Kumar 2021). It enables learning of joint video-text representations from large unlabelled video dataset. The earlier models used linguistic features for video-text joint modelling whereas ActBERT leverages three sources of information for cross-model pre-training such as action-related features, region features, and language embeddings. It has been able to outperform supervised models.

2.6 Discussion: limitations and opportunities

We will now discuss and summarize the findings of this Subsection, presenting the main trends and most likely future developments.

We can see that most of the recent development when dealing with unlabelled data have been about the use of self-supervised techniques. This is mainly due to the trend of deep generative models, which focus precisely on this type of task, but can be extended to other tasks. Around half of these algorithms can also perform semi-supervised tasks, and that represents all but one of the algorithms with semi-supervised capabilities. Most of the algorithms for clustering and dimensionality reduction tasks are also self-supervised, as they focus on extracting groups or relevant feature from the latent spaces created by the generative models.

One relevant note on the trend of deep generative models is their relevance over time. GAN and VAE started to have relevance sooner, and as such are more mature, especially in the scientific literature. In the last two years, the focus has been shifting, first towards transformers, and more recently towards diffusion models. Transformers, due to their architecture, are mostly used in text processing tasks, while diffusion models focus mostly on computer vision.

Very few algorithms were specifically dedicated to the areas of active learning, zero-shot learning, and weak supervision. This may be based on the fact that self-supervised techniques require less information and assumptions from the user when developing these models. Active learning requires direct input from users, zero-shot learning requires that users provide semantic relations between old and new labels, and weak supervision requires assumptions about the labelling process. This reflects an idea underlying the supervision spectrum presented in Sect. 2, that the division of the groups of technique/tasks are not crisp, but occur along an axis, specially when it comes to the information gathering strategy. If there is the possibility to gather more information from the data alone, those strategies will be usually preferred when compared with strategies that require more input.

Algorithms for explainability tasks represent a group of their own, with the notable exception of SVAE, which aims at fusing graph representation and DL to enhance model interpretability.

In what concerns the applications, most algorithms focus on either text processing or computer vision. The exception are the larger families of deep generative algorithms (VAE, GAN, transformers, diffusion models), which reflect the flexibility and broad application of this kind of algorithms. As a considerable amount of the reviewed algorithms are specific applications of these generic algorithms to a certain field, it is expected that they focus on an area in particular. Outside the generic deep generative learning algorithms, there were eight algorithms for each field.

Based on the previous discussion, we can say that one of the most relevant trends is the use DL generative models, such as GAN and VAE, Diffusion models, and Transformers due to their high performance, flexibility and transfer learning capabilities. Most of the opportunities for future developments focus on making the training process more stable (especially in GAN), improving the computation of loss functions (VAE), improving global context by minimizing context fragmentation in transformers, and making the diffusion models' training faster.

Another relevant trend are contrastive methods, which are SOTA in self-supervised learning tasks for computer vision. Given their characteristics, it is expected that

contrastive methods may assume a larger role in other applications, such as text processing or explainability.

The last emerging trend is the development of explainability/interpretability strategies, as most recent developments are heavily focused on performance, in detriment of transparency and understanding. This could be an issue in high-risk situation, where it is required to monitor to the performance and viability of the deployed models. In terms of future research, contrastive and adversarial strategies can be used to enhance explainability. Another research direction could be as suggested in Barredo Arrieta et al. (2020) (of which Johnson et al. (2016) is an example), and combine DL with graphical representation to enhance interpretability.

Finally, a future research direction could be the use of meta-learning or AutoML for unsupervised tasks. This direction is suggested in van Engelen and Hoos (2020), as these methods have been applied extensively to supervised tasks, but not to situations with unlabeled data. The main obstacle to this implementation lies most likely in defining good performance measures for unsupervised/semi-supervised/self-supervised tasks that would allow for the meta-learning algorithms to guide their process.

In addition to the future research directions, we would like to include a brief discussion on the current and future risks and securities issues with AI, and DL in particular. Although AI has been used to improve cyber security, we are currently witnessing the use of AI technology in cyber attacks (Radanliev et al. 2022b). Although AI is used to predict cyber attacks and other preventive measures, AI and ML are now being used to develop bots with malicious intents, such as spread of disinformation and propaganda (Radanliev et al. 2022b), polluting and poisoning training data, leading to bias and undesired effects (Radanliev et al. 2022b; Guo et al. 2022). In DGM in particular, backdoor attacks can occur, as enterprises make use of third-party models, given their large computation requirements in terms of training (Rawat et al. 2022). There is also the issue of model extraction attacks, where the aim is to steal the original models by training surrogate models (Lee et al. 2022). However, recent studies have detected these issues, and already formulated solutions to tackle these issues, such as Guo et al. (2022), Lee et al. (2022), and Rawat et al. (2022). In terms of future risks of AI, Radanliev et al. (2022a) forecasts that an increase of investment on edge computing will lead to more autonomous machines. This is in line with one of our future research directions, of AutoML developing to include unsupervised tasks. This increasing automation and "self-reliance" of ML algorithms may lead to bias, as setting the wrong direction at the start of a self-evolving, self-procreating AI may escalate into bigger risks and possible damages. One thing that may bias in such ways is the referred strong investment in military and defensive oriented AI, which can lead to militarization of future AI systems (Radanliev et al. 2022a). This is increasingly probable as research indicates that AI is evolving based on human capabilities, and not necessarily due to human needs or desires (Radanliev et al. 2022a), and these capabilities are developing at a faster pace in sectors with higher investment.

3 Machine learning operations

In this Section, Machine Learning Operations, or MLOps, techniques, tools and practices are discussed, revealing approaches and reporting experiences with the objective of declaring how to make the operationalization of ML integrated systems more efficient.

3.1 Introduction

The concept of MLOps comes from the combination of ML with DevOps (Development and Operations) (Kreuzberger et al. 2022), which, in turn, can be understood as a set of practices and tools, that together with an appropriate mindset of the stakeholders creates an agile methodology that brings the development and operations teams together, automates processes and reduces manual intervention (Subramanya et al. 2022b). MLOps is a recent phenomenon with a wide spectrum of activities, supported by different software tools, and as new tools appear with different purposes and objectives, the responsibilities and requirements become less clear (Ruf et al. 2021). The investigation carried out within the scope of this work, still revealed a scarce experience report, being those found based on partial implementations with several interconnected technologies and focused on automating parts of the workflow. This Section also discusses trends and emerging developments regarding AI/ML services in Cloud environments, namely those under the concept of ML as a Service (MLaaS). A detailed analysis and discretization on relevant factors, advantages and disadvantages related to the use of Cloud Computing vs On-Premises Systems is realized, and also, based on the collected information in recent literature, a discussion is promoted about what to expect in the near future of cloud services for AI/ML.

3.2 Typical workflow

The design, operationalization and deployment of a ML system is based on a set of sequential steps, which are not pre-defined or standardized in any way and may vary depending on the specific needs of the use case. The study carried out in this work revealed that it is possible to characterize a typical workflow of MLOps based on three major areas of Engineering (Recupito et al. 2022), as depicted in Fig. 5: Data Engineering, Model Engineering and Operations and Deployment. Part of the literature also mentions the Requirements phase as relevant to MLOps, however, this will not be the objective of a very in-depth study, but the ones previously mentioned (Schlegel and Sattler 2022).

Within the scope of the MLOps workflow, Data Engineering emerges as an area composed of several tasks whose main objective is to guarantee that the data has the necessary quality (Baesens et al. 2021) and is adequate to the identified requirement for the use case.

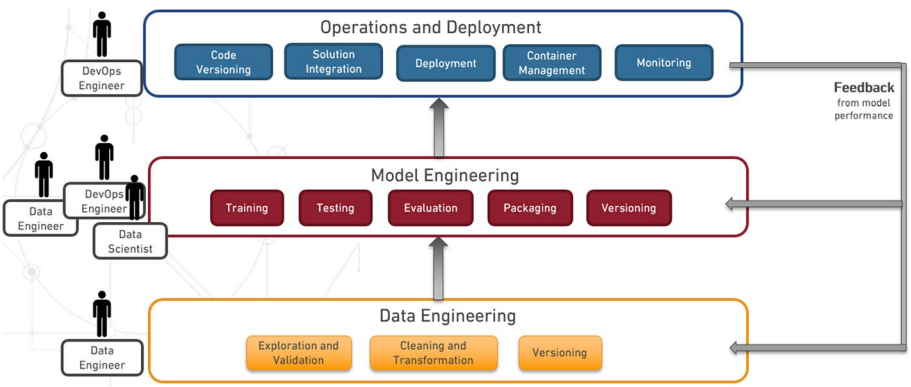

Fig. 5 Typical MLOps workflow

Tasks such as Data Exploration, Cleaning, Transformation or Versioning fit into this phase of the workflow, mostly performed by a Data Engineer. Data quality issues must be identified as soon as possible, and in accordance with Schelter et al. (2018), through a multidimensional approach, so that the quality assessment is guaranteed for various prisms, such as completeness, which aims to assess the presence of empty data, consistency, to ensure that a set of semantic rules are not violated, and accuracy to how correct the data is syntactically and semantically. Model Engineering concerns a phase of the workflow where tasks mainly related to ML models are expected to be carried out (Kreuzberger et al. 2022), such as, defining the ML pipeline, which may vary depending on the input data types or the techniques used to run the algorithms, training models, selecting algorithms and hyperparameters, versioning and validating the models, etc. However, the input for this phase results from Data Engineering and the output is directed towards integration with Operations, which promotes interaction between the Data Engineer, Data Scientist and DevOps Engineer profiles.

3.3 Challenges

The implementation of a ML workflow and consequent operationalization and automation of steps brings challenges inherent to the nature of the work that ML implies to be carried out (Ruf et al. 2021; Kreuzberger et al. 2022). In a solution intended to be integrated with a software ecosystem, it is inevitable to identify and anticipate the difficulties that may arise in the process of automating this part, with some of the most relevant being described below, from Organizational to Software Engineering and Operational issues. The process of identifying the best model for a ML use case is iterative and based on trial and error (Zhang et al, 2022), in a cycle where several experiments must be carried out, tracking the relevant metrics so that the optimal solution is found, in a process that is highly dependent on the human eye and that may be complex to automate. The logging and registration of actions and results in experiments is manual. Data is not versioned by nature and changes over time, plus, producing results in ML requires code to implement an algorithm and data to train and test the model, so the whole experiment needs to be versioned, being that this would bring the concept of maintenance of experiments, allowing future reproducibility. Code review and testing strategies, already matured and recurrently used in software development methodologies, are not commonly seen in ML projects, just as there is no development environment shared by teams, being the use of Jupyter notebooks or individual Python scripts the most common approach (Ruf et al. 2021). Collaboration between team members is complex due to the difficulty in sharing models or other ML artefacts and the scarce existence of dedicated collaborative work software tools. The deployed model is not automatically retrained, which from a Continuous Integration and Continuous Delivery perspective should be mandatory, as there will be a tendency to degrade its performance over time, plus data input variation in volume will also make hard to predict the needed requirements for infrastructure resources (CPU, RAM and GPU). Changing a parameter, for example a specific hyperparameter that impacts the dimensions of a deep neural network, influences the entire assembled pipeline, which may require changes in the following steps. In a production infrastructure based on a combination of technologies, it can be challenging to identify the root cause for failures, given that they can happen in different parts of it, and the addition of ML systems to DevOps projects introduces another layer of complexity. It is also relevant to mention aspects related to the mindset and culture as a consequence of the multidisciplinary team that will inevitably need to interact, sharing the

same vision, which must be product-driven. The multidisciplinarity of necessary profiles for the implementation of MLOps practices is also, in itself, a challenge, due to the wide-range of skills required.

3.4 Software tools

This study allowed the identification of several software tools available in the market, related to some of the activity areas involved in the MLOps process. Some tools have redundant functionalities, others were built for more specific purposes, being focused on smaller parts of the process. No tool can be indicated as the only solution for all scenarios, due to the diversity of needs that these will present, but some tools can complement each other, allowing to design a solution according to the needs. Table 3 presents the tools identified, as well as some of their main features.

3.5 AI/ML cloud trends

The present Subsection explores recent developments and the trends in the offer and use of cloud computing tools and platforms for ML applications. The objective is to raise important topics that allow framing particular work processes with the available and emerging tools on these platforms. The aim is to provide an important set of information for decision-making on the use of cloud computing tools vs. localized tools and the main advantages and disadvantages of each of these approaches.

3.5.1 Machine learning as a service

The evolution of services based on Cloud Computing over the last few years, especially with regard to the computational power made available by these vendor's infrastructures, was one of the factors that boosted the emergence of cloud-based services aimed at ML (Yousif 2017). These, adopted by the term ML as a Service (MLaaS) (Lee 2018), are highly suited to the daily work of data scientists and data engineers but also for agents that do not have much expertise in data science, some providing a user-friendly interface, that cover a wide spectrum of tasks that can range from data transformation to the deployment of ML models. The cloud environment provides a suitable ML platform, since it can allow, in a modular manner, to add or subtract computing power, other hardware or even software features on demand, the pricing model is based on pay-as-you-go, meaning that the customer will only pay for what he uses, it has the capability to easily store large volumes of data, and to add to this, serve the model as a webservice. MLaaS gives accessibility to ML tools without the need to purchase specific hardware or install dedicated software and/or dependencies. Although the literature still does not demonstrate many results using this type of solution, it was possible to identify that the main providers of cloud computing services on the market have been making ML resources available for different areas and with different applicabilities (Neptunes 2022). Amazon AWS provides services such as Amazon Polly, which is a text-to-speech with support for several languages (English, Brazilian Portuguese, Danish, French, among other), Amazon SageMaker, to build, train, and deploy ML models with support for all leading deep learning frameworks such as TensorFlow, PyTorch or Apache MXNet, or Amazon Lex, which is a conversational AI for chatbots to build conversational interfaces, etc. Google Cloud Platform have been investing in the full management of ML Workflows with the availability of a unified platform,

Table 3 MLOps tools and features identified

Tool	Data engineering	Model engineering	Operations
KubeFlow	Data transformation	Model selection, training, tracking, or hyperparameter tuning	Monitoring of deployment infrastructure
TensorFlow extended	Data validation	Model training, Tracking, Packaging or Registry	Model Performance Monitoring
ZenML	Data validation	Model training, tracking, integration with multiple ML frameworks, model versioning	Model deployment
Polyaxon	Data verification	Model training, tracking, hyperparameter tuning, or model packaging	Pipeline monitoring
MetaFlow	N/A	Artefacts versioning, pipeline orchestration	System logging, deployment scheduling
MLFlow		Model training, tracking, integration with multiple ML frameworks	
DVC	Data source handling, preprocessing, versioning, etc	Model registry	Experiment tracking
GoCD			Workflow design and monitoring, deployment automation
Flyte		Model training	Data and model monitoring, workflow orchestration
Prometheus			System and data monitoring, model performance monitoring

Vertex AI, and in providing an adequate infrastructure to train deep-learning models cost-effectively using high-performance cloud GPUs and TPUs with the availability of its AI Infrastructure. Microsoft Azure has Azure ML Studio under its umbrella, which is a no-code web interface for developers and data scientists that provides a large range of services for building, training, and deploying ML models faster and has built-in modules that help pre-process the data. The Azure environment also provides Azure Automated ML, which accelerates model creation by automating iterative tasks, and Azure MLOps that aims to accelerate automation, collaboration, and reproducibility of ML workflows. IBM also provides Watson, which consists of a ML wide range of tools, such as, visual recognition through image analysis, natural language classification, speech-to-text, among others. Most of these platforms provide a free trial or free usage with a specified limit. The study on cloud platforms for AI/ML solutions allowed to identify several tools under the term MLaaS, which are focused on quite different algorithms and applicabilities. The lack or difficulty in finding the necessary skills for internal teams to delve into these topics favors solutions based on MLaaS, as these tend to require less expertise, however, the fact that there is a greater dependence on internet access, as well as on third-party managed storage are disadvantages.

3.5.2 Cloud vs local

In many technological areas, such as IoT, business analytics, engineering and advanced calculation, and data science, which use intensively information technologies, the question of whether to use cloud or localized services arises. In the case of data science and in particular when using DL models, which are highly consuming of computer resources and eager for data, having localized or on-premises systems comes at a high cost. On-premises servers for DL require high-end solutions in terms of CPU (Zhao et al. 2021), multiple GPUs and very fast storage and intermittent memory solutions that reach very high budget investments. These investments are also characterized by high rates of depreciation and obsolescence with frequent changes, by system providers, to system architectures and solution performance. In a few years (less than 2 to 3) the equipment could be completely outdated compared to recently launched equipment. This is where cloud systems like MLaaS have the greatest advantages as they provide access to special hardware configurations, including recent GPUs and massively parallel high-performance computing (HPC) systems. Access to these systems is costed based on usage on a pay-as-you-go basis, and there is no initial investment, which makes them especially attractive for businesses or activities where there is a preference for variable costs.

The same reasoning applies to software. When installing an on-premises solution, licenses must be acquired for a (usually limited) set of software, while in cloud systems a wide range of tools are available to use, experiment and test, allowing to verify which software solutions are best suited to the needs.

Necessarily, maintaining an on-premises solution also requires maintaining a team of hardware and software specialists to carry out system maintenance and solve problems that may arise. In cloud systems this need does not exist, or at least it won't be as intense, especially in (PaaS - Platform as a Service) offering solutions. In the case of IaaS (Infrastructure as a Service) the need for a specialist team remains.

In cloud systems, it becomes necessary to pass data back and forth to run the models in the cloud. Performance thus becomes dependent on internet connectivity. Model latency is a more often recognized reason to opt for localized solutions, with more controlled

connection speeds, especially in weaker connectivity applications such as mobile devices or edge applications. The same applies in the course of training DL models. DL models use large amounts of data in their training and the need to constantly transfer data to and from platforms creates latencies in the process.

Some difficulties are pointed out to cloud systems related to changing tools, functionalities and eventually discontinuities that disrupt established processes and force users to constantly change their tools. In local systems, with greater control over versions, updates and tools used, these difficulties are minimized.

Privacy is one of the main problems pointed out to cloud services (Neicu et al. 2020). Especially in ML tasks, where algorithms must access large amounts of raw data, it is complex to guarantee the security and privacy of this information throughout the entire ML process, from training to deployment and use of models (Philipp et al. 2021). Simultaneously, there is the possibility that models already in use in the cloud may be the target of subversive attacks that alter their content and behaviour, leading to different results than intended.

In ML operations, we can classify users into basic, intermediate and advanced levels. The basic level uses the models as black boxes applied to their data, later analysing the results without deep knowledge of ML or programming. The intermediate level, on the other hand, does not yet have in-depth knowledge in data science, but makes advanced use of existing models, varying parameters and adjusting their databases, requiring some programming skills to do so. Finally, the advanced level, called data scientists, who adapt or develop their own models and architectures, therefore have advanced programming skills (López García et al. 2020). There is no direct allocation rule for the adaptability of each of the levels to cloud or local tools, but the first two levels, basic and intermediate, are levels well adapted to taking advantage of the tools offered by MLaaS services, while the advanced level, with operations of very accurate MLOps and a well-established software ecosystem, working in advanced DL with very frequent operations certainly makes an on-premises system viable.

Although the business model associated with cloud services points to advantages in terms of costs compared to on-premises systems, companies with significant AI programs report high costs in large-scale operations of cloud infrastructure, especially for heavy GPU use and other AI-optimized gear. This suggests that an accurate analysis of costs, possibly in the face of multiple scenarios of evolution, is fundamental to devise strategies for choosing solutions.

3.5.3 Hybrid cloud

With the aim of building solutions capable of taking advantage of each of the strategies, local and cloud, the concept of hybrid architecture emerged, that is, the possibility of integration between a local infrastructure and resources of a public cloud platform through the deployment of their native services to on-premises, giving greater flexibility to the solution design, and allowing a granular level of resource management through cloud bursting, that is, in case there are peaks in resource needs, the response capacity can be transferred to the cloud (Mansouri et al. 2020). Due to the complexity of specifying a solution of this typology, the literature is scarce in concrete examples of hybrid architectures, nevertheless, Mansouri et al. (2020) demonstrates an automated implementation of a hybrid architecture using WireGuard, a Linux-based VPN, to ensure a secure connection between private infrastructure and public cloud, and Terraform software tool for infrastructure resources

deployment based on the required number of Virtual Machine instances. Private cloud components were implemented with OpenStack, and public cloud was provided by Microsoft Azure. Also, within the same work, an evaluation of performance of six database systems was done as they burst into the public cloud, revealing better results for MongoDB and MySQL in throughput and latency of read/write operations, contrasting with Cassandra, Riak, CouchDB and Redis.

Some major and other emerging cloud service providers offer hybrid cloud solutions that, as an example, let customers keep sensitive data locally while still using cloud resources for analytics. Organizations may profit from cloud computing while still keeping control over sensitive data that must be retained on-premises. In the same way, the usage of encrypted data is strongly tied to the use of hybrid cloud technologies, as encrypted data aids addressing security and privacy concerns while keeping sensitive information in the cloud. Before being transferred or stored, information is converted into a safe, unreadable format by encryption, preventing theft or unwanted access. For data stored in the cloud, several cloud service providers offer encryption capabilities, either through built-in encryption services or by enabling the use of third-party encryption solutions. Depending on the unique needs of the company, encrypted data can be kept both locally or in the cloud, with the encryption keys being controlled locally or remotely. One of the possible approaches is to use ML/DL techniques in a hybrid cloud setup with sensitive and encrypted data stored on-premises. On this configuration, the ML/DL algorithms would be executed in the cloud, being able to leverage on the computing power of the cloud for complex data analytics, but the encrypted data would be secured kept on-site. In a safe way, the encrypted data would be sent to the cloud, where it would be decrypted, processed, and then re-encrypted before being sent back to the on-premises environment. Table 4 present some major and emerging service providers that offer hybrid cloud solutions that can be explored to allow for sensitive data to be secured stored locally on-premises though using cloud resources for performing complex analytics.

3.5.4 Future developments

Future developments in AI/ML in cloud environments, and more specifically, MLaaS, are expected to happen. In general, this area does not seem to be moving towards the existence of a single platform capable of carrying out all types of tasks, but towards several integrated tools that allow achieving the objectives, therefore, the interoperability between tools is a growing factor. Moreover, for existing software scenarios based on micro-services architecture, solutions such as MLaaS fit very well, as they allow ML to also function as one more service capable of responding to different sources of requests. It is also expected that there will be a tendency for MLaaS tools to be increasingly user-friendly and with greater capacity to automatically apply the correct ML models, as well as the optimal hyperparameters configurations (AutoML), supporting teams that do not have the skills necessary to do so independently. It is also relevant to mention that several contributions highlight MLaaS as platforms that enable the sharing and validation of models, and the evaluation to what extent the validation process belongs to the MLaaS platform is a promising area. Ethics and data privacy should also be subject of future development within the context of MLaaS.

Since privacy is becoming one of the main concerns of MLaaS services, the topic is being the subject of research at the academic level (Philipp et al. 2021) and the focus of interest on the part of the main service providers and it is expected that there will be developments in the near future. MLaaS providers are working in offers that provide added value

Table 4 Hybrid cloud services

Service provider	Service name	Brief description
Microsoft Azure	Azure arc	Platform for integration and management of multi-cloud or on-premises resources with Azure. Servers, Data Services, Kubernetes Clusters or Virtual Machines are among the compatible services
Amazon AWS	AWS outposts	IT-as-a-service with dedicated hardware delivered by Amazon itself, configured accordingly, providing a secure connection to Cloud services via a VPN
Google Cloud	Anthos	Container orchestration platform (Google Kubernetes Engine) for infrastructure provisioning in both cloud and on-premises
IBM Cloud	Cloud private; cloud satellite	Provides a distributed cloud architecture that integrates the scalability and flexibility of cloud services to the applications and data that run in on-premises secure private cloud
Snowflake	Data Lake	Offers a data lake-based architecture with a strong governance of data and advanced data collaboration though maintaining high levels of security
H2O.ai	H_2O AI cloud - hybrid	Provides a hybrid solution, giving customers possibility of operation in any cloud or on-premises with complete control over infrastructure, software updates, security and compliance

in terms of accommodating customer data concerns about region-specific compliance and data sovereignty.

Developments are also foreseen in hybrid architectures specially tailored to ML operations that try to take advantage of the strengths of each of the approaches (cloud and on-premises). Although offers of hybrid cloud services can be found on the market, some of them tailored to MLOps, evolutions in their consistency and maturity are expected. Sensitive data security and encryption is a topic in focus in the near future. Organizations plan storing their sensitive data internally while scaling and optimizing their workload on external cloud platforms, because each cloud provider offers different services, prices, support levels, etc., a solution could be to have more than one service provider, exchanges of information have to take place between local and cloud services and between cloud services and security problems can occur. Security features are actually focused on protecting particular services and less prone to secure complex transmission of data. Hybrid cloud brings together the advantages of on-premises systems and cloud services, but they can increase costs, as to the costs of cloud services overlap the costs of owning an infrastructure, not to mention the increased need for IT skills.

3.6 Selection of tools

One of the main challenges is that there is not a defined workflow, with standard and clearly identified steps. Needs appear depending on the scenarios that entities face in their infrastructures, and consequently, software tools are being developed and made available to the community to solve parts of the workflow. This causes several tools to have redundant functionalities, partially solving the same problems, sometimes with different technological restrictions, and many present gaps with regard to integration and automation.As such, currently building an MLOps workflow for the first time is an iterative process (Ruf et al. 2021) based on a trial-and-error strategy, and whose first focus should be to identify which steps/stages of the workflow need or benefit from automation, starting with smaller steps or individual stages. This analysis will make it possible to clearly identify the needs or bottlenecks, which tools or technologies are most appropriate for the context, and, also relevant, which teams will work with these tools, and if there is a need for specific training in a timely manner. Many of the identified tools, as well as others available on the market, provide a wide range of functionalities, sometimes solving problems adjacent to different workflow steps, bringing complexity and redundancy of functionalities. The prior existence of tools or frameworks already used within the organization itself is an aspect to be taken into account, as it will be good practice to ensure that any extra software tool is compatible and integrable with the existing ecosystem. This will also ease the learning curve in the skills needed and to promote internally. For example, if the software ecosystem already installed in an organization is mostly developed in Python or based on a Microsoft environment, compatibility with these technologies should be guaranteed before moving further with any extra tool. Another relevant point in the implementation of MLOps in an organization is the development and release methodology, which must be agile and respect the principles of DevOps, promoting collaboration between development and operations teams, based on methods of continuous integration and continuous delivery. This alignment between all the teams involved in the process makes it possible to avoid conflicting situations and incompatibility between technologies used by the development team and by the operations team, therefore spending extra time initially is beneficial in the longer run. During the process of implementing MLOps, the goal should be to fulfil the requirements,

solve prioritized bottlenecks and automate in small steps, keeping in mind a final vision of the implementation to avoid decisions that will be harmful in the long run.

3.7 Use-cases in machine learning operations

In this Subsection real use-cases of ML Operations tools and practices implementations are reported, covering from data engineering practices to model training and deployment.

3.7.1 Bioinformatics application with Kubeflow for batch processing in clouds

ML has wide applications in Bioinformatics, for example, genomic sequence assembly, literature analysis and image processing. Some ML pipelines take weeks to complete a training cycle, exceeding the time-limit of High Performance Computing queues. The training cycles need to be repeated many times for hyperparameter tuning. This use case (Yuan and Wildish 2020) reports the usage of Kubeflow on top of Kubernetes for job scheduling, workflow management and first class support for ML. The paper discloses that several pipelines were deployed and executed on Kubernetes in OpenStack, Google Cloud Platform and Amazon Web Services. Kubeflow and Kubernetes were chosen to avoid the overhead of provisioning of virtual machines, to achieve rapid scaling with containers, and to be truly cloud-agnostic in all cloud environments. The used method were based on Docker and Kubernetes for container and container orchestration, since those are almost standard across the industry. Plus, all major cloud providers and operating systems provide first class support for them, thus in previous investigation, the authors have confirmed that Bioinformatics pipelines could be migrated from HPC into public clouds with ease. Kubernetes clusters for HPC on three clouds are reported: OpenStack, Google Cloud Platform and Amazon AWS, running Kubernetes Engines (Rancher Kubernetes Engine on OpenStack, built-in Kubernetes Engines on Google and Amazon). KubeFlow for batch processing was also deployed with a cloud-agnostic script (maintained by the open source community or third party) that was completely portable, allowing the deployment of Kubeflow on OpenStack, GCP and AWS without any modification, reducing operational costs in production and implementing a hybrid cloud strategy. It turned out to have a consistent mechanism for authentication and authorization. The deployment could also have been done with cloud-specific scripts (maintained by the cloud providers) that provide tight integration with the underlying cloud infrastructure. Regarding storage, since Bioinformatics pipelines almost always assume local access to POSIX-like file systems for both read and write an NFS persistent volume was used as workaround to make the pipelines cloud-agnostic. To integrate internal networks created by Kubeflow with the outside world, three options for networking were used, port-forward, load balancer and Ingress gateway service. For Data Access, commands such as curl, wget, or scp, were used, as well as specific clients, to download or upload files in the pipelines. This approach turned out to have some drawbacks, being the biggest issue the scalability. Since data files have to be moved in batch mode and then processed, and they often require large amounts of storage from Terabytes to Petabytes, it gets an inefficient strategy. For Monitoring the Kubernetes clusters, Elastichsearch with Kibana and Beats were used, allowing logging and telemetry of the clusters in the different cloud providers. Regarding GPU usage, Kubeflow runs on Kubernetes clusters with or without it. The OpenStack pipeline were positioned for CPU-only training, but on Google Cloud Platform and AWS, clusters may include GPU, allowing to bypass the timeout issues with HPC queues, to avoid long GPU procurement cycles, to acquire larger capacities and to

minimise the cost in public clouds. As a result, the use case reports to have successfully ran two types of pipelines on Kubeflow/Kubernetes on GCP, AWS and OSK, enhancing and proving the capability of the platform for two Bioinformatics applications, a Classic Bioinformatics pipelines for genomic sequence analysis, representing high throughput workload, and ML pipelines for image classification on cardiomyocytes from an Image Data Repository, representing high performance workload. Both were successfully deployed, confirming that Kubeflow and Kubernetes can satisfy complex requirements by Bioinformatics via container orchestration in all major clouds with excellent portability.

3.7.2 MLOps scaling machine learning lifecycle in an industrial setting

This work (Zhao et al. 2022) presents a framework focused on streamlining and introducing best practices to facilitate the ML lifecycle in an industrial setting, which can be used as a template and be customized to implement various ML experiments. The proposed framework is modular, can be recomposed to be adapted to different use cases and inherits practices from DevOps introducing the automation of the entire ML lifecycle, approximating development and operations. The first step discussed in the paper was the requirement analysis, which overviewed the ML development lifecycle in a classification task use-case that tells whether a meal is a combo-meal or not. Here, there was an understanding of the business requirements, what were the behaviours expected from the model and what were the features needed. The most time-consuming parts turned out to be Data Preparation and Data Labelling. Meal data was collected from various online food delivery platforms whit the need of manual labelling first, based on the definition of combo meal. After this, data scientists worked on model building, training and evaluation, where different versions of data and ML artefacts were generated by this iterative process. The identified problems were the lack of data, ML artefacts version control, standardized development process, and an automated development flow. The proposed architecture is composed of two main parts: ML Model development and Model Deployment. Model Development starts with a git repository, which contains all necessary ML algorithm code for developing the model, and where DVC is used to keep track of the data. After that, building ML models, generating features, and tuning parameters and execute ML jobs in local environment or within Amazon AWS cloud platform. Then, the experiment results and other ML artefacts are tracked by MLflow tracking. After the iterative experiment process, DVC is used to version control the final models and any other ML artefacts. Lastly, Git is used to release a new version for the model in that specific state. All dataset, models and ML artefacts are stored on a Cloud Environment, as long as they have been tracked by DVC and/or MLflow tracking. During the development, Jenkins, an open-source automation server, is used for unit testing of code, automatic building and continuous deployment.After obtaining the trained model and artefacts that will be used in production, starts the deployment part. Here, another git repository is used, where a docker image is prepared with prediction logic and inference code to be used in AWS Sagemaker, start the batch transform job, apply the prediction on unseen data and store the transformed data into AWS S3. The model and other artefacts used in production are loaded during runtime via a DVC API. Airflow is also used to automatically trigger the model deployment pipeline. As a closing note, this paper presented the design of a MLOps framework and respective infrastructures that improve the ML lifecycle management process. The proposed framework streamlines and automates the ML lifecycle, reducing the delivery time and labour work, and giving more reliability, trust, traceability and scalability to the ML development process.

3.7.3 Training and serving ML workloads with Kubeflow at CERN

This work (Golubovic and Rocha 2021) describes a new service available at CERN,based on Kubeflow and managing the full ML lifecycle: data preparation and inter-active analysis,large scale distributed model training and model serving. Specific features for hyper-parameter tuning and model meta-data management,as well as infrastructure details to integrate accelerators and external resources are also covered. The first step was to identify what would be the objectives for the service to implement. This was important, since very different objectives could result in a very different architecture for the solution. As a result of this analysis, some key features identified were ease of use, meaning that end users should be able to continue using their favorite tools and libraries such as Jupyter Notebooks with TensorFlow, PyTorch or MXNET, availability, which is important considering that the service is also serving the models, capability of aggregating similar workloads in one place, scalability, and sustainability meaning that selected tools must be well established and have strong communities contributing for discussion and problem solving. Based on these, KubeFlow was the chosen platform. CERN already ran in production a large private cloud based on OpenStack, originally built to offer virtual machines and the ability to attach virtual storage appliances, which grew significantly and expanded to include advanced networking features like security groups or Load Balancing as a Service(LBaaS) and a managed Kubernetes service, capable of handling accelerators such as GPUs. The new service was implemented to cover a large variety of ML use cases, therefore, some of the main features were, capability for the users to maintain multiple notebook servers, provide a common workflow to perform data retrieval followed by pre-processing, execute a few variations of model training in parallel and aggregate multiple results, hyperparameter optimization,distributed training and model serving. These were all provided by KubeFlow and Kubernetes. A use case is presented with a fast calorimeter simulation with 3D GANs, to test the new service, where linear scaling was achieved with a large number of GPUs, as well as the equivalent results when using TPUs (Tensor Processing Units). Future developments on the service are also identified, such as, integrating additional data sources, in particular the systems hosting the log data for CERN systems,improving the multi cluster and multi cloud experience, and integrating and evaluating new types of resources, particularly FPGAs which promise a significant improvement in model service.

3.8 Limitations and opportunities

The study carried out within the scope of this work allowed to understand that the concept of MLOps itself is still immature, and despite the presented potential, it is still necessary for entities to start adopting, on an iterative manner, practices and tools in order to test limits in real scenarios. The most recent advances were noted, above all, in the implementation of ML pipelines and workflow orchestration, with the availability of various tools for this purpose, such as MLFlow or KubeFlow. Data and Model Versioning also suffered developments along the last years, with tools such as Data Version Control (DVC). There will be room for future developments in the synergies created between teams, due to different necessary roles and skills for the implementation of MLOps Workflows in the entities infrastructures to create any real business value. It will be necessary for DevOps teams to understand the needs and how to operationalize ML systems, as well as for Data Engineering or ML development teams to understand how Continuous Integration and Continuous Delivery is implemented by Software Engineering and Operations teams. Different tools

have been evolving differently, thus, creating overlapping features and increasing redundancy, and as the tool variety for supporting the different phases in ML projects is constantly evolving, many operational tools are expected to be refined, particularly within interconnection capabilities.Several tools have been presenting solutions for Data Validation or Data Verification, given that is an important step on the pipeline, which can avoid unnecessary efforts and time spending, therefore, strategies and tools for assessing data quality are expected to continue to be the subject of developments. The current limitations identified in the literature, result from the lack of governance, guidelines or manifestos (Mboweni et al. 2022) for the implementation of MLOps projects, although, recently, principles and processes of ML development for design, build and manage reproducible and testable ML-powered software have been under work (MLOpsOrg 2022).

4 Conclusions

The first steps on AI theories and applications go back to the 60 s of the last century, with the so-called first AI wave, focused on knowledge reasoning. The second AI wave started in the 80 s and continued till the end of the first decade of this century, founded on statistical models and simple neural networks able to deal with uncertainty and trained to adapt to different environments. The third and present AI wave began with the appearance of DL, which is still regarded as the central technology for pushing AI in the next years. Indeed, in the short and medium terms, future trends and technologies are expected to continue focused on DL, while awaiting, in a longer term, for disruptive developments, both in models and support hardware that can constitute a basis for a fourth, not yet foreseen, AI wave. Most recent advances on AI/ML techniques have been focused on DL generative models, such as GANs and VAEs, Transformers and Diffusion models. This is due to these models characteristics, namely their high performance, flexibility and transfer learning capabilities. Indeed, the potential of generative models may be explored to reduce the dimensionality, find exploratory factors, and learn representations in the presence of unlabelled or poorly labelled data. Most of the opportunities for future developments focus on increasing the stability of the training processes (especially in GANs), improving the computation of loss functions (namely in VAEs), improving global context by minimizing context fragmentation (in transformers), and making the diffusion models training faster. In the near future, contrastive methods, being SOTA in self-supervised learning tasks for computer vision, are expected to assume a key role in certain applications, such as text processing or explainability. A crucial emerging trend is the development of explainability/interpretability strategies, which will overcome one of the main limitations of DL, which is the difficulty in explaining actions and results. Indeed, recent AI/ML developments have been deeply focused on performance, in detriment of transparency and understanding. However, this may be an issue in high-risk situations, where it is required to monitor the performance and viability of the deployed models. In terms of future research, contrastive and adversarial strategies are expected to be used to enhance explainability. Another research direction can be to combine DL with graphical representations to enhance interpretability. In terms of AI/ML algorithms, another future research direction can be the use of meta-learning or AutoML for unsupervised tasks. Their potential to create AI systems that enhance or proactively find methods that mitigate issues, such as excessive time and resources required in the search and tuning of algorithms, can be explored. Meta-learning methods have been applied extensively to supervised tasks, but not to unlabelled data. The main obstacle to

this laying, most likely, in defining good performance measures for unsupervised/semi-supervised/self-supervised tasks that would allow for the meta-learning algorithms to guide their process. On another hand, AI/ML systems consist of algorithms, which need to be implemented on computer-based infrastructures composed of hardware and software. The automation of the various tasks involved and the selection of adequate work methodologies, with the aim of making AI/ML systems quickly available, while reducing human intervention has become a key issue. The concept of MLOps emerged, representing an advance in the synergies between several fields. This concept is still immature, and despite its potential, it is still necessary for entities to start adopting, on an iterative manner, practices and tools in order to test limits in real scenarios. Indeed, many limitations result from the lack of governance, guidelines or manifestos for the implementation of MLOps projects. At the same time that MLOps investigation is being carried out in the research community, tools are appearing on the market that target workflows for model deployment and administration. The most recent advances are mostly in the implementation of ML pipelines and workflow orchestration, with the availability of various tools, such as MLFlow or KubeFlow. Data and Model Versioning also suffered developments along the last years, with tools such as DVC. An accelerated evolution in MLOps is expected in the next years. There will be room for future developments in the synergies created between teams, due to different necessary roles and skills for the implementation of MLOps Workflows. It will be necessary for DevOps teams to understand the needs and how to operationalize ML systems, as well as for Data Engineering or ML development teams to understand how Continuous Integration and Continuous Delivery is implemented by Software Engineering and Operations teams. Different tools have been evolving differently, thus, creating overlapping features and increasing redundancy, and as the tool variety for supporting the different phases in ML projects is constantly evolving, many operational tools are expected to be refined, particularly within interconnection capabilities. Several tools have been presenting solutions for Data Validation or Data Verification. Therefore, strategies and tools for assessing data quality are expected to see further developments. Finally, it is expected a pronounced evolution of the AI Cloud services. These services bring promises of access to high performance systems without high investments and maintenance costs, accompanied with an offer of advanced tools, easy parametrization and continuous evolution. However, concerns about confidentiality and unpredictable costs are key obstacles.

In terms of methodological limitations of the present research, we can mention the following points:

- Given the huge amount of information that is nowadays available, searching criteria were adopted, imposing restrictions that could have not allowed to capture the full breadth of existing technologies;
- The exponential development of AI technologies and the speed at they evolve lead to the necessity of constantly update any review research;
- Related to the previous point, many novel technologies that were addressed herein may not become fully developed, due to the lack of investment and research, as more prominent alternatives are constantly emerging;
- The present review focused mainly on scientific literature. This may have led to an issue, as a lot of current developments on AI are done by global companies, which reserve information due to commercial reasons, limiting the availability of information.

These mentioned limitations suggest some future research avenues, namely:

- To perform reviews focused on each of the identified trends (e.g., DGM, image processing, hybrid cloud MLOps), with greater detail;
- To develop an historical analysis that evaluates how each technology developed, which may enable better forecasts of the performance of new technologies;

Acknowledgements This work was undertaken within a project sponsored by NATO's Allied Command Transformation (ACT).

Author contributions E.O., M.R., J.P.P., A.M.L., I.I.M., S.B.: Conceptualization, Structure. E.O., M.R., J.P.P., A.M.L.: Methodology, Data curation, Writing-original draft. E.O., M.R., J.P.P., A.M.L., I.I.M., S.B.: Writing-review, editing, and validation. Supervision: J.P.P., A.M.L.

Funding Open access funding provided by FCT|FCCN (b-on).

Declarations

Conflict of interest The authors declare no conflicts of interest.

References

Acheampong FA, Nunoo-Mensah H, Chen W (2021) Transformer models for text-based emotion detection: a review of bert-based approaches. Artif Intell Rev 54(8):5789–5829. https://doi.org/10.1007/s10462-021-09958-2

Ahmad M, Batcha MS, Jahina SR (2021) Testing Lotka's law and pattern of author productivity in the scholarly publications of artificial intelligence. arXiv preprint arXiv:2102.09182, https://doi.org/10.48550/ARXIV.2102.09182

Awan AA (2022) Top 7 diffusion-based applications with demos. https://www.kdnuggets.com/2022/10/top-7-diffusionbased-applications-demos.html. Accessed 15 Dec 2022

Azad N (2022) Understanding devops critical success factors and organizational practices, pp 83–90. https://doi.org/10.1145/3524614.3528627,

Baesens B, Höppner S, Verdonck T (2021). Data engineering for fraud detection. https://doi.org/10.1016/j.dss.2021.113492

Barredo Arrieta A, Díaz-Rodríguez N, Del Ser J et al (2020) Explainable artificial intelligence (xai): concepts, taxonomies, opportunities and challenges toward responsible ai. Inf Fusion 58:82–115

Bhalgat Y, Liu Z, Gundecha P et al (2019) Teacher-student learning paradigm for tri-training: an efficient method for unlabeled data exploitation. https://doi.org/10.48550/ARXIV.1909.11233, arXiv:1909.11233

Brownlee J (2019) A gentle introduction to generative adversarial networks (gans). https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/. Accessed 12 Dec 2022

Caron M, Misra I, Mairal J et al (2020) Unsupervised learning of visual features by contrasting cluster assignments. https://doi.org/10.48550/ARXIV.2006.09882, arXiv: 2006.09882

Chattopadhay A, Sarkar A, Howlader P et al (2018) Grad-cam++: generalized gradient-based visual explanations for deep convolutional networks. In: 2018 IEEE winter conference on applications of computer vision (WACV), pp 839–847, https://doi.org/10.1109/WACV.2018.00097

Chen S, Ngai E, Ku Y et al (2023) Prediction of hotel booking cancellations: integration of machine learning and probability model based on interpretable feature interaction. Decis Supp Syst. https://doi.org/10.1016/j.dss.2023.113959

Chen T, Kornblith S, Norouzi M et al (2020) A simple framework for contrastive learning of visual representations. https://doi.org/10.48550/ARXIV.2002.05709, arXiv: 2002.05709
Conneau A, Khandelwal K, Goyal N et al (2019) Unsupervised cross-lingual representation learning at scale. https://doi.org/10.48550/ARXIV.1911.02116, arXiv: 1911.02116
Croitoru FA, Hondru V, Ionescu RT et al (2022) Diffusion models in vision: a survey. https://doi.org/10.48550/ARXIV.2209.04747, arXiv: 2209.04747
de Andrade Silva J, Hruschka ER, Gama J (2017) An evolutionary algorithm for clustering data streams with a variable number of clusters. Expert Syst Appl 67:228–238. https://doi.org/10.1016/j.eswa.2016.09.020
Deng L (2018) Artificial intelligence in the rising wave of deep learning: the historical path and future outlook [perspectives]. IEEE Signal Process Mag 35(1):177–180. https://doi.org/10.1109/MSP.2017.2762725
Devlin J, Chang MW, Lee K et al (2018) Bert: pre-training of deep bidirectional transformers for language understanding. https://doi.org/10.48550/ARXIV.1810.04805, arXiv: 1810.04805
Dhariwal P, Nichol A (2021) Diffusion models beat gans on image synthesis. https://doi.org/10.48550/ARXIV.2105.05233, arXiv: 2105.05233
Dhurandhar A, Chen PY, Luss R et al (2018) Explanations based on the missing: towards contrastive explanations with pertinent negatives. https://doi.org/10.48550/ARXIV.1802.07623, arXiv: 1802.07623
Donahue J, Simonyan K (2019) Large scale adversarial representation learning. Curran Associates Inc., Red Hook
Donahue J, Krähenbühl P, Darrell T (2016) Adversarial feature learning. https://doi.org/10.48550/ARXIV.1605.09782, arXiv: 1605.09782
Dong S, Wang P, Abbas K (2021) A survey on deep learning and its applications. Comput Sci Rev 40(100):379. https://doi.org/10.1016/j.cosrev.2021.100379
Dosovitskiy A, Brox T (2016) Generating images with perceptual similarity metrics based on deep networks. https://doi.org/10.48550/ARXIV.1602.02644, arXiv: 1602.02644
Goldstein A, Kapelner A, Bleich J et al (2013) Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. https://doi.org/10.48550/ARXIV.1309.6392, arXiv: 1309.6392
Golubovic D, Rocha R (2021) Training and serving ml workloads with Kubeflow at CERN. EPJ Web Conf 251(02):067. https://doi.org/10.1051/epjconf/202125102067
Gong C, Yang J, You J et al (2022) Centroid estimation with guaranteed efficiency: a general framework for weakly supervised learning. IEEE Trans Pattern Anal Mach Intell 44(6):2841–2855. https://doi.org/10.1109/TPAMI.2020.3044997
Goodfellow IJ, Pouget-Abadie J, Mirza M et al (2014) Generative adversarial networks. https://doi.org/10.48550/ARXIV.1406.2661, arXiv: 1406.2661
Guo Q, Wu D, Qi Y, et al (2022) FLMJR: Improving Robustness of Federated Learning via Model Stability. In: Atluri V, Di Pietro R, Jensen CD, Meng W (eds) Computer security - ESORICS 2022. ESORICS 2022. Lecture notes in computer science, vol 13556. Springer, Cham. https://doi.org/10.1007/978-3-031-17143-7_20,
John MM, Olsson HH, Bosch J (2021) Towards mlops: a framework and maturity model, pp 1–8. https://doi.org/10.1109/SEAA53835.2021.00050
Johnson MJ, Duvenaud D, Wiltschko AB et al (2016) Composing graphical models with neural networks for structured representations and fast inference. https://doi.org/10.48550/ARXIV.1603.06277, arXiv: 1603.06277
Karhade M (2022) What is gpt-4 (and when?). https://pub.towardsai.net/what-is-gpt-4-and-when-9f5073f25a6d. Accessed 12 Dec 2022
Kingma DP, Welling M (2013) Auto-encoding variational bayes. https://doi.org/10.48550/ARXIV.1312.6114, arXiv: abs/1312.6114
Konyushkova K, Sznitman R, Fua P (2017) Learning active learning from data. https://doi.org/10.48550/ARXIV.1703.03365, arXiv: 1703.03365
Kreuzberger D, Kühl N, Hirschl S (2022) Machine learning operations (mlops): overview, definition, and architecture. https://doi.org/10.48550/ARXIV.2205.02302, arXiv: 2205.02302
Le N, Rathour VS, Yamazaki K et al (2022) Deep reinforcement learning in computer vision: a comprehensive survey. Artif Intell Rev 55(4):2733–2819. https://doi.org/10.1007/s10462-021-10061-9
Le NQK, Ho QT, Nguyen VN et al (2022) Bert-promoter: an improved sequence-based predictor of dna promoter using bert pre-trained model and shape feature selection. Comput Biol Chem 99(107):732. https://doi.org/10.1016/j.compbiolchem.2022.107732
Lee YS (2018) Analysis on trends of machine learning-as-a-service. Int J Adv Cult Technol 6(4):303–308

Lee, Y., Jun, S., Cho, Y., et al (2022) Precise extraction of deep learning models via side-channel attacks on edge/endpoint devices. In: Atluri, V., Di Pietro, R., Jensen, C.D., Meng, W. (eds) Computer Security - ESORICS 2022. ESORICS 2022. Lecture Notes in Computer Science, vol 13556. Springer, Cham. https://doi.org/10.1007/978-3-031-17143-7_18,

Linardatos P, Papastefanopoulos V, Kotsiantis S (2021) Explainable ai: a review of machine learning interpretability methods. Entropy. https://doi.org/10.3390/e23010018

Liu Y, Ott M, Goyal N et al (2019) Roberta: a robustly optimized bert pretraining approach. https://doi.org/10.48550/ARXIV.1907.11692, arXiv: 1907.11692

López García A, De Lucas JM, Antonacci M et al (2020) A cloud-based framework for machine learning workloads and applications. IEEE Access 8(18):681–692. https://doi.org/10.1109/ACCESS.2020.2964386

Lundberg S, Lee SI (2017) A unified approach to interpreting model predictions. https://doi.org/10.48550/ARXIV.1705.07874, arXiv: 1705.07874

Mäkinen S, Skogström H, Laaksonen E et al (2021) Who needs mlops: what data scientists seek to accomplish and how can mlops help? https://doi.org/10.48550/ARXIV.2103.08942, arXiv: 2103.08942

Mansouri Y, Prokhorenko V, Babar MA (2020) An automated implementation of hybrid cloud for performance evaluation of distributed databases. J Netw Comput Appl 167(102):740. https://doi.org/10.1016/j.jnca.2020.102740

Mboweni T, Masombuka T, Dongmo C (2022) A systematic review of machine learning devops. In: 2022 international conference on electrical, computer and energy technologies (ICECET), pp 1–6. https://doi.org/10.1109/ICECET55527.2022.9872968

Meta AI (2019) Xlm-r: state-of-the-art cross-lingual understanding through self-supervision. https://ai.facebook.com/blog/-xlm-r-state-of-the-art-cross-lingual-understanding-through-self-supervision/. Accessed 12 Dec 2022

MLOpsOrg (2022) Mlopsorg. https://ml-ops.org/

Mnih V, Kavukcuoglu K, Silver D et al (2015) Human-level control through deep reinforcement learning. Nature 518(7540):529–533. https://doi.org/10.1038/nature14236

Molnar C (2022) Interpretable machine learning, 2nd edn. https://christophm.github.io/interpretable-ml-book

Mosca E, Demirtürk D, Mülln L et al (2022) GrammarSHAP: an efficient model-agnostic and structure-aware NLP explainer. In: Proceedings of the first workshop on learning with natural language supervision. association for computational linguistics, Dublin, Ireland, pp 10–16. https://doi.org/10.18653/v1/2022.lnls-1.2, https://aclanthology.org/2022.lnls-1.2

Murphy KP (2012) Machine learning: a probabilistic perspective. The MIT Press, New York

Neicu A, Radu A, Zaman G, Stoica I, Rapan F (2020) Cloud computing usage in SMEs. An empirical study based on SMEs employees perceptions. https://doi.org/10.3390/su12124960

Nelson FF, Ernst R, Akesson B et al (2022) Deep machine learning for cyber defence. Report of STO Research Task IST-163 (IWA) - The NATO Science and Technology Organization

Neptunes AI (2022) Neptunes.ai. https://neptune.ai/blog/best-machine-learning-as-a-service-platforms-mlaas

Ning X, Wang X, Xu S et al (2021) A review of research on co-training. Concurr Comput. https://doi.org/10.1002/cpe.6276

Ohri K, Kumar M (2021) Review on self-supervised image recognition using deep neural networks. Knowl-Based Syst 224(107):090. https://doi.org/10.1016/j.knosys.2021.107090

Open AI (2022) Dall-e 2. https://openai.com/dall-e-2/. Accessed 15 Dec 2022

OpenAI (2022) Chatgpt: optimizing language models for dialogue. https://openai.com/blog/chatgpt/. Accessed 14 Dec 2022

Philipp R, Mladenow A, Strauss C et al (2021) Machine learning as a service: challenges in research and applications. In: Proceedings of the 22nd international conference on information integration and web-based applications & services. Association for Computing Machinery, New York, NY, USA, iiWAS '20, p 396-406, https://doi.org/10.1145/3428757.3429152,

Poyiadzi R, Bacaicoa-Barber D, Cid-Sueiro J et al (2022) The weak supervision landscape. https://doi.org/10.48550/ARXIV.2203.16282, arXiv: 2203.16282

Radanliev P, De Roure D, Maple C et al (2022) Forecasts on future evolution of artificial intelligence and intelligent systems. IEEE Access. 10:45280–45288. https://doi.org/10.1109/ACCESS.2022.3169580

Radanliev P, De Roure D, Maple C et al (2022) Super-forecasting the 'technological singularity' risks from artificial intelligence. Evol Syst 13:747–757. https://doi.org/10.1007/s12530-022-09431-7

Radford A, Narasimhan K (2018) Improving language understanding by generative pre-training

Raffel C, Shazeer N, Roberts A et al (2019) Exploring the limits of transfer learning with a unified text-to-text transformer. https://doi.org/10.48550/ARXIV.1910.10683, arXiv: 1910.10683

Rawat A, Levacher K, Sinn M et al (2022) The devil is in the GAN: backdoor attacks and defenses in deep generative models. In: Atluri V, Di Pietro R, Jensen CD, Meng W (eds) Computer Security - ESORICS 2022. ESORICS 2022. Lecture notes in computer science, vol 13556. Springer, Cham. https://doi.org/10.1007/978-3-031-17143-7_41

Recupito G, Pecorelli F, Catolino G et al (2022). A multivocal literature review of MLOps tools and features. https://doi.org/10.1109/SEAA56994.2022.00021

Ren P, Xiao Y, Chang X et al (2020) A survey of deep active learning. https://doi.org/10.48550/ARXIV.2009.00236, arXiv: 2009.00236

Ren Y, Pu J, Yang Z et al (2022) Deep clustering: a comprehensive survey. https://doi.org/10.48550/ARXIV.2210.04142, arXiv: 2210.04142

Research G (2020) Exploring transfer learning with t5: the text-to-text transfer transformer. https://ai.googleblog.com/2020/02/exploring-transfer-learning-with-t5.html. Accessed 12 Dec 2022

Ribeiro MT, Singh S, Guestrin C (2016) "why should i trust you?": explaining the predictions of any classifier. https://doi.org/10.48550/ARXIV.1602.04938, arXiv: 1602.04938

Ribeiro MT, Singh S, Guestrin C (2018) Anchors: High-precision model-agnostic explanations. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence. AAAI Press, AAAI'18/IAAI'18/EAAI'18

Rizinski M, Peshov H, Mishev K et al (2022) Ethically responsible machine learning in fintech. IEEE Access 10:97531–97554. https://doi.org/10.1109/ACCESS.2022.3202889

Ruf P, Madan M, Reich C et al (2021) Demystifying mlops and presenting a recipe for the selection of open-source tools. Appl Sci. https://doi.org/10.3390/app1198861, http://www.mdpi.com/2076-3417/11/19/8861

Sanh V, Debut L, Chaumond J et al (2019) Distilbert, a distilled version of Bert: smaller, faster, cheaper and lighter. https://doi.org/10.48550/ARXIV.1910.01108, arXiv: 1910.01108

Sayin B, Krivosheev E, Yang J et al (2021) A review and experimental analysis of active learning over crowdsourced data. Artif Intell Rev 54(7):5283–5305. https://doi.org/10.1007/s10462-021-10021-3

Schelter S, Lange D, Schmidt P et al (2018) Automating large-scale data quality verification. Proc VLDB Endow 11:1781–1794. https://doi.org/10.14778/3229863.3229867

Schlegel M, Sattler KU (2022) Management of machine learning lifecycle artifacts: a survey https://doi.org/10.48550/ARXIV.2210.11831, arXiv: 2210.11831

Selvaraju RR, Cogswell M, Das A et al (2019) Grad-CAM: Visual explanations from deep networks via gradient-based localization. Int J Comput Vision 128(2):336–359. https://doi.org/10.1007/s11263-019-01228-7

Shao Z, Zhao R, Yuan S et al (2022) Tracing the evolution of ai in the past decade and forecasting the emerging trends. Expert Syst Appl 209(118):221. https://doi.org/10.1016/j.eswa.2022.118221

Shrikumar A, Greenside P, Kundaje A (2017) Learning important features through propagating activation differences https://doi.org/10.48550/ARXIV.1704.02685, arXiv: 1704.02685

Siddiqui JR (2022) Diffusion models made easy. https://towardsdatascience.com/diffusion-models-made-easy-8414298ce4da. Accessed 12 Dec 2022

SliceTeller: a data slice-driven approach for machine learning model validation.https://doi.org/10.1109/TVCG.2022.3209465

Sohl-Dickstein J, Weiss EA, Maheswaranathan N et al (2015) Deep unsupervised learning using nonequilibrium thermodynamics. https://doi.org/10.48550/ARXIV.1503.03585, arXiv: 1503.03585

Subramanya R, Sierla S, Vyatkin V (2022) From devops to mlops: overview and application to electricity market forecasting. Appl Sci 12:19. https://doi.org/10.3390/app12199851

Subramanya R, Sierla S, Vyatkin V (2022) From devops to mlops: overview and application to electricity market forecasting. Appl Sci. https://doi.org/10.3390/app12199851, http://www.mdpi.com/2076-3417/12/19/9851

Sun-Hosoya L, Guyon I, Sebag M (2018) Activmetal: algorithm recommendation with active meta learning. In: IAL@PKDD/ECML

Svenmarck P, Luotsinen L, Nilsson M et al (2018) Possibilities and challenges for artificial intelligence in military applications. In: NATO big data and artificial intelligence for military decision making specialists' meeting

Ullah I, Manzo M, Shah M et al (2022) Graph convolutional networks: analysis, improvements and results. Appl Intell 52(8):9033–9044. https://doi.org/10.1007/s10489-021-02973-4

van Engelen JE, Hoos HH (2020) A survey on semi-supervised learning. Mach Learn 109(2):373–440. https://doi.org/10.1007/s10994-019-05855-6

Wang W, Zheng VW, Yu H et al (2019) A survey of zero-shot learning: settings, methods, and applications. ACM Trans Intell Syst Technol. https://doi.org/10.1145/3293318

Yang Z, Dai Z, Yang Y et al (2019) Xlnet: generalized autoregressive pretraining for language understanding. https://doi.org/10.48550/ARXIV.1906.08237, arXiv: 1906.08237

Yousif M (2017) Intelligence in the cloud - we need a lot of it. IEEE Cloud Comput. https://doi.org/10.1109/MCC.2018.1081057

Yuan DY, Wildish T (2020) Bioinformatics application with Kubeflow for batch processing in clouds. In: Lecture notes in computer science. Springer, Berlin. pp 355–367. https://doi.org/10.1007/978-3-030-59851-8_24

Zhang C, Lu Y (2021) Study on artificial intelligence: the state of the art and future prospects. J Ind Inf Integr 23(100):224. https://doi.org/10.1016/j.jii.2021.100224

Zhang Y, Tangwongsan K, Tirthapura S (2017) Streaming k-means clustering with fast queries. In: 2017 IEEE 33rd international conference on data engineering (ICDE), pp 449–460. https://doi.org/10.1109/ICDE.2017.102

Zhang J, Zhang H, Xia C et al (2020) Graph-bert: only attention is needed for learning graph representations. https://doi.org/10.48550/ARXIV.2001.05140, arXiv: 2001.05140

Zhao S, Song J, Ermon S (2017) Towards deeper understanding of variational autoencoding models. https://doi.org/10.48550/ARXIV.1702.08658, arXiv: 1702.08658

Zhao L, Wang Q, Wang C, Li Q, Shen C, Feng B (2021) VeriML: enabling integrity assurances and fair payments for machine learning as a service https://doi.org/10.1109/TPDS.2021.3068195

Zhao Y, Belloum ASZ, da Costa GM et al (2022) Mlops scaling machine learning lifecycle in an industrial setting. Int J Ind Manuf Eng 16(5):138–148

Zhou B, Khosla A, Lapedriza A et al (2015) Learning deep features for discriminative localization. https://doi.org/10.48550/ARXIV.1512.04150, arXiv: 1512.04150

Zhu L, Yang Y (2020) Actbert: learning global-local video-text representations. https://doi.org/10.48550/ARXIV.2011.07231, arXiv: 2011.07231

Zubaroğlu A, Atalay V (2021) Data stream clustering: a review. Artif Intell Rev 54(2):1201–1236. https://doi.org/10.1007/s10462-020-09874-x

Authors and Affiliations

Eduardo e Oliveira[1,2] · Marco Rodrigues[1] · João Paulo Pereira[1] · António M. Lopes[1,2] · Ivana Ilic Mestric[3] · Sandro Bjelogrlic[3]

✉ António M. Lopes
aml@fe.up.pt

Eduardo e Oliveira
eoliveira@inegi.up.pt

Marco Rodrigues
mrodrigues@inegi.up.pt

João Paulo Pereira
jptp@inegi.up.pt

Ivana Ilic Mestric
Ivana.IlicMestric@ncia.nato.int

Sandro Bjelogrlic
sandro.bjelogrlic@ncia.nato.int

[1] INEGI - Institute of Science and Innovation in Mechanical and Industrial Engineering, Porto, Portugal

[2] LAETA/INEGI, Faculty of Engineering, University of Porto, Rua Dr. Roberto Frias, 4200-465 Porto, Portugal

[3] NATO Communications and Information Agency, Oude Waalsdorperweg 61, 2597 AK The Hague, The Netherlands