Received: 2 July 2021 | Revised: 13 April 2022 | Accepted: 14 April 2022

DOI: 10.1002/widm.1470

ADVANCED REVIEW

WIREs DATA MINING AND KNOWLEDGE DISCOVERY

WILEY

Taxonomy of machine learning paradigms: A data-centric perspective

Frank Emmert-Streib[1] | Matthias Dehmer[2,3]

[1]Predictive Society and Data Analytics Lab, Faculty of Information Technology and Communication Sciences, Tampere University, Tampere, Finland

[2]Swiss Distance University of Applied Sciences, Brig, Switzerland

[3]Department of Mechatronics and Biomedical Computer Science, UMIT, Hall in Tyrol, Austria

Correspondence
Frank Emmert-Streib, Predictive Society and Data Analytics Lab, Faculty of Information Technology and Communication Sciences, Tampere University, Tampere, Finland.
Email: frank.emmert.streib@gmail.com

Edited by: Mehmed Kantardzic, Associate Editor and Witold Pedrycz, Editor-in-Chief

Abstract

Machine learning is a field composed of various pillars. Traditionally, supervised learning (SL), unsupervised learning (UL), and reinforcement learning (RL) are the dominating learning paradigms that inspired the field since the 1950s. Based on these, thousands of different methods have been developed during the last seven decades used in nearly all application domains. However, recently, other learning paradigms are gaining momentum which complement and extend the above learning paradigms significantly. These are multi-label learning (MLL), semi-supervised learning (SSL), one-class classification (OCC), positive-unlabeled learning (PUL), transfer learning (TL), multi-task learning (MTL), and one-shot learning (OSL). The purpose of this article is a systematic discussion of these modern learning paradigms and their connection to the traditional ones. We discuss each of the learning paradigms formally by defining key constituents and paying particular attention to the data requirements for allowing an easy connection to applications. That means, we assume a data-driven perspective. This perspective will also allow a systematic identification of relations between the individual learning paradigms in the form of a learning-paradigm graph (LP-graph). Overall, the LP-graph establishes a taxonomy among 10 different learning paradigms.

This article is categorized under:

Technologies > Machine Learning

Application Areas > Science and Technology

Fundamental Concepts of Data and Knowledge > Key Design Issues in Data Mining

KEYWORDS

artificial intelligence, machine learning, multi-label learning, multi-task learning, transfer learning

1 | INTRODUCTION

Machine learning has a long history and systematic studies date back at least to the 1940s. Despite the fact that there is no definite starting date for the field, Arthur Samuel is frequently credited for coining the term "machine learning" in the 1950s when working on checkers playing (Samuel, 1959).

In general, one needs to distinguish between a machine learning paradigm and methodological realizations thereof. For instance, supervised learning (SL) is a machine learning paradigm that deals either with classification or regression problems (Carbonell et al., 1983). In the case of classification, this allows the categorization of different instances into separate classes, for example, with a support-vector machine (SVM; Vapnik, 1995), which is a particular method that can be used for such an analysis. Since the beginnings of machine learning the field focused on three major learning paradigms: SL, unsupervised learning (UL), and reinforcement learning (RL). For these paradigms, thousands of different methods and algorithms (Bishop, 2006; Flach, 2012; Haste et al., 2009) have been developed over the decades and studied in various application domains ranging from biology, medicine, and engineering to sociology and psychology (Dwyer et al., 2018; Smolander et al., 2019; J. Wang et al., 2018). We think it is fair to say that to this day, machine learning is dominated by the above three learning paradigms and methodological realizations thereof to the extent that most people would even struggle to name extension thereof. For instance, a recent review of machine learning applications in manufacturing focused only on SL, UL, and RL (Wuest et al., 2016). Similar exclusive reviews from other domains can be found in Jordan and Mitchell (2015), Libbrecht and Noble (2015), and Vamathevan et al. (2019). The purpose of this article is a systematic review and discussion of machine learning paradigms beyond the traditional ones.

Specifically, we discuss seven more recent learning paradigms that extend and complement the three traditional machine learning paradigms. These learning paradigms are multi-label learning (MLL), semi-supervised learning (SSL), one-class classification (OCC), positive-unlabeled learning (PUL), transfer learning (TL), multi-task learning (MTL), and one-shot learning (OSL). In this article, each of these learning paradigms is introduced formally by defining key constituents. We will pay particular attention to the data requirements to allow an easy connection to applications. That means we assume a data-driven perspective.

There are many survey papers focusing on a variety of subfields of machine learning. For instance, there are reviews about methods for SSL (Van Engelen & Hoos, 2020), OCC (Rodionova et al., 2016), PUL (Bekker & Davis, 2020), few/OSL (Y. Wang et al., 2020), TL (Weiss et al., 2016), MTL (Y. Zhang & Yang, 2018), and MLL (Gibaja & Ventura, 2014) or applications in a variety of different fields. However, our paper is different. Specifically, the novel contribution of our paper is 4-fold. First, instead of focusing on methods or applications of methods, we focus on learning paradigms. That means we are concerned with the fundamental ideas and definitions underlying machine learning paradigms. Second, we do not only discuss one learning paradigm, but we present 10 different machine learning paradigms side-by-side. This gives a global overview and insights by avoiding a tunnel vision. Third, despite the formal nature of a learning paradigm, we assume a data-centric perspective which has two advantages. First, for applications this makes it easier to select the most appropriate paradigm for a given problem, and, second, it simplifies the formal comparison of machine learning paradigms. Fourth, we provide a comparison of 10 machine learning paradigms in the form of an interrelation diagram, which we call the learning-paradigm graph (LP-graph). In the LP-graph, two learning paradigms are connected if one can map one paradigm onto another one by a data or information influencing operation (which we define in Section 12). This will establishes a taxonomy among the 10 learning paradigms. Finally, we would like to mention that to the best of our knowledge so far, there is no review that would cover all 10 machine learning paradigms side-by-side and discuss their relations.

This article is organized as follows. In the next section, we present information about the research methods needed for the remainder of the paper. Then we briefly review the three traditional machine learning paradigms, that is, SL, UL, and RL, to make the later comparisons more clear. Thereafter, we discuss seven modern machine learning paradigms and present applications thereof. The following section discusses the interrelation between all machine learning paradigms and introduces formal mappings between these. This will establish a taxonomy. Finally, the paper finishes with conclusions.

2 | RESEARCH METHODS

In this article, our focus is on machine learning paradigms and their relations. According to Kuhn (1970) and Capra (1996), a *paradigm* is characterized as follows.

A scientific paradigm is a set of concepts, patterns, or assumptions to which those in a particular professional community are committed and which form the basis of further research.

In order to further emphasize the importance of a paradigm in a scientific context, the term *worldview* has been suggested as a synonym to describe "a way of thinking about and making sense of the complexities of the real world" (Kaushik & Walsh, 2019; Patton & Fund, 2002). Since machine learning is a scientific field, the above definition can be directly applied to define the machine learning paradigm.

The brief discussion above about the term paradigm should make it clear that machine learning paradigms correspond to worldviews of learning from data and as such are conceptually more important than individual methods or models used for studying practical application problems. Put simply, machine learning paradigms allow to perceive the entire universe whereas the methods correspond to its stars.

In this article, we discuss a variety of machine learning paradigms that are beyond the well-known traditional paradigms SL, UL, and RL. Specifically, we will discuss seven further learning paradigms: SSL, OCC, PUL, OSL, TL, MTL, and MLL. Furthermore, we discuss the interrelations between all these machine learning paradigms.

2.1 | Literature review of the different machine learning paradigms

In order to show that each of the above learning paradigms is already in use, at least to a certain extent, and not just a theoretical construct with no value for real-world applications, we conducted a literature study. The result of this literature search provides us with information about the (1) frequency of usage of a machine learning paradigm and their (2) time evolution.

2.1.1 | Frequency of usage of a machine learning paradigm

For our literature study, we conducted a keyword search for published articles in Google Scholar, PubMed, World of Science (WoS), and the Conference on Neural Information Processing Systems (NIPS) and counted the frequency of listed articles. The results are shown in Figure 1 where we show an overview of the usage of the 10 different machine learning paradigms. In order to show all results in one figure, we rescaled the numbers logarithmically (base 10) because the frequency of published articles varies greatly for the different paradigms.

We selected Google Scholar as a reference literature database for general publications in all areas of science, PubMed for publications in biology and biomedical sciences, WoS for a well-curated citation database and NIPS for its

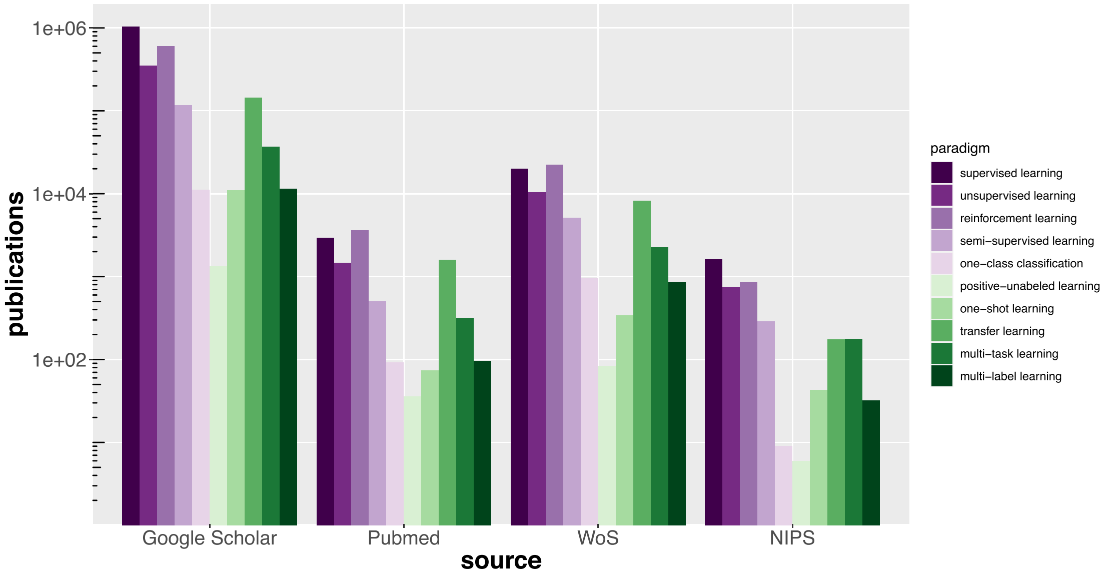


FIGURE 1 Number of published articles about different machine learning paradigms as found in different sources. The *y*-axis is in a logarithmic (base 10) scale because the number of articles varies greatly.

leading role in machine learning conferences. This diversity should allow to obtain a broad overview of the appearance of the machine learning paradigms in very different fields of research across nearly all scientific areas. From this literature search, we can draw the following conclusions.

From Figure 1, one can make a number of different observations. First, despite the fact that the total number of publications in the different sources is considerably different the overall pattern seems quite similar. Specifically, the order in the number of published articles in all different machine learning paradigms is similar across the sources (Figure 1). This is remarkable considering the vast differences in the underlying scientific communities because PubMed represents application studies in biology and the biomedical sciences whereas NIPS is a flagship conference for technical methods and machine learning theories.

Second, there is a large factor range between the usage of advanced machine learning paradigms compared to the traditional machine learning paradigms, for example, SL. Specifically, for TL one finds factors of 7.2, 1.8, 2.4, and 9.3 for Google Scholar, PubMed, WoS, and NIPS, whereas for PUL, the factors are 791.9, 81.1, 236.7, and 268.3 correspondingly. That means, for instance, for every published article in WoS about PUL one finds in average $236.7\,(=19,882/84)$ articles about SL. From these factor ranges one can conclude that all advanced machine learning paradigms are used to a much lesser extent than SL, UL, and RL regardless of the scientific community, as exemplified by the four different sources in Figure 1.

2.1.2 | Time evolution of the usage of machine learning paradigms

In order to complement information about the number of published articles about different machine learning paradigms, shown in Figure 1, we studied also the time evolution. Specifically, in Figure 2, we show the time series of the number of publications about different machine learning paradigms according to Google Scholar (A.) and Web of Science (B.). We use Google Scholar (GS) and Web of Science (WoS) because GS represents general publications in all areas of science while WoS is a well-curated database.

As one can see from Figure 2a,b, the three traditional machine learning paradigms—SL, UL, and RL (denoted by the numbers 1–3 in the figure)—have been first adopted in the literature and are to this day quantitatively dominating the number of publications. It is this observation upon which we base the usage of the term “modern” when distinguishing the three traditional machine learning paradigms from the seven nontraditional machine learning paradigms.

2.2 | Traditional machine learning paradigms

Before we discuss the advanced machine learning paradigms, we review briefly the three traditional learning paradigms.

- Supervised learning.
- Unsupervised learning.
- Reinforcement learning.

This will allow a well-defined comparison in later sections to identify similarities and differences between the traditional and advanced learning paradigms.

2.2.1 | Supervised learning

For SL, one needs two components: (i) data and (ii) a task. The data provide information about the instances in the form

$$D_s = \{(x_i, y_i)\}_{i=1}^{n} \tag{1}$$

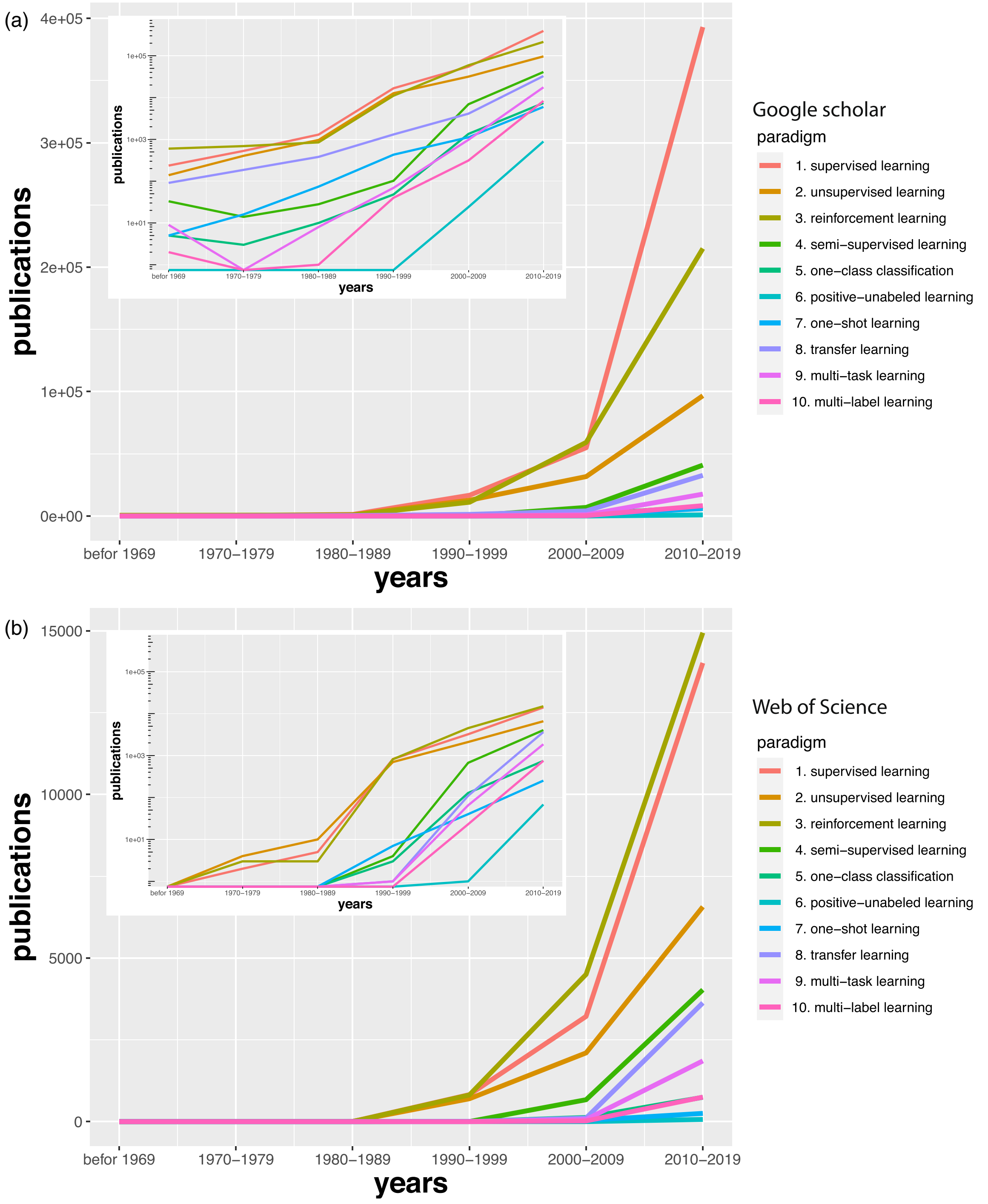


FIGURE 2 Time series of the number of publications about different machine learning paradigms according to Google scholar (a) and web of science (b). The y-axis of the inlay is in a logarithmic (base 10) scale to obtain a better resolution for the beginning years.

with $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$ where $\mathcal{X}$ is the feature space (also known as input space), $\mathcal{Y}$ is the outcome space, and n is the sample size. In general, the elements in the feature space represent information about an experiment, for example, the measurement of gene expression values, sensor signals, or buying behavior of a consumer. Depending on the underlying experiment, the dimensionality of the feature space and its type are defined. For instance, measuring the gene expression of m genes results in m-dimensional feature vectors $x \in \mathcal{X}$ with $dim(x) = m$. Furthermore, the way the components of a feature vector are measured defines the scale (or level) of a measurement. For example, this can correspond to real values, integer values, or categorical variables. For simplicity, in the following, we assume $\mathcal{X} = \mathbb{R}^m$, that is, the feature space corresponds to m-dimensional real numbers.

For the elements in $\mathcal{Y}$, we need to distinguish two cases. Case (i): Elements in $\mathcal{Y}$ are categorical. Case (ii): Elements in $\mathcal{Y}$ are real valued. The first case leads to classification problems whereas the second one corresponds to regression problems. In the simplest form a categorical space leads to binary classification, that is, $\mathcal{Y} = \{C_1, C_2\}$ with class labels C_1 and C_2. In this case, $\mathcal{Y}$ is called label space. In contrast, a one-dimensional real valued $\mathcal{Y} = \mathbb{R}$ leads to (multiple) linear (or nonlinear) regression. Hence, the scale of measurement of $\mathcal{Y}$ decides if one has a classification or regression problem.

Given the definition of elements in $\mathcal{X}$ and $\mathcal{Y}$, their common occurrence is given by the joint probability distribution $P(X,Y)$ which allows to draw samples, that is, $(x,y) \sim P(X,Y)$. Integration over the $\mathcal{Y}$ space results in the marginal distribution $P(X)$, that is, $P(X) = \int P(X,Y)dY$, of feature vectors with $X \in \mathcal{X}$.

Combining all the above allows us now to provide the definition of a domain.

Definition 2.1. *A domain $\mathcal{D}$ consists of a feature space χ and a marginal probability distribution $P(X)$ where $X \in \mathcal{X}$ given by $\mathcal{D} = \{\mathcal{X}, P(X)\}$.*

The second component of SL is a task. Put simply, the task provides a mapping from the feature space $\mathcal{X}$ into the outcome space $\mathcal{Y}$. Formally, a task, $\mathcal{T}$, is defined as follows.

Definition 2.2. *A task $\mathcal{T}$ consists of a outcome space $\mathcal{Y}$ and a prediction function $f(X)$ with $f: \mathcal{X} \rightarrow \mathcal{Y}$, that is, $\mathcal{T} = \{\mathcal{Y}, f(X)\}$.*

In the case of classification, the prediction function f assigns labels to $f(X)$ with $X \in \mathcal{X}$, whereas for regression the prediction function f assumes continuous values, that is, $f(X) \in \mathbb{R}$.

2.2.2 | Unsupervised learning

The difference between SL and UL is that for UL, there is no outcome space $\mathcal{Y}$ available. From this follow two implications. First, the data assume the form

$$D_u = \{(x_i)\}_{i=1}^{n} \tag{2}$$

with $x_i \in \mathcal{X}$. Here $\mathcal{X}$ is again the feature space and n the sample size. Second, due to the nonavailability of an outcome space, a task, as given in Definition 2.2, is no longer defined.

We would like to remark that the latter is responsible that UL finds application in exploratory data analysis (Hoaglin et al., 1983). For completeness, we would like to remark that it is also used for unsupervised classification, data discretization, and dimensionality reduction.

2.2.3 | Reinforcement learning

In contrast to SL and UL, RL is considerably different. In RL, there is no given data, either in the form of Equations (1) or (2). Instead, an agent is interacting with an environment via actions that lead to a change in the state of the environment resulting in a new state (Sutton & Barto, 1998). The iterative application of this cycle generates a sequence of actions and states from which the agent aims to learn the best policy for action making. Importantly, the agent receives a feedback about the quality of its actions consisting only of a scalar reinforcement signal indicating either “reward” or “punishment.” Hence, no quantitative feedback is available that could be used for the learning process by the agent. Overall, RL aims at learning a policy to maximize the future return which consists of all future rewards (Kaelbling et al., 1996). In Figure 3, we show the basic components that define the key elements of RL.

Given the above formulation of RL, it is no surprise that the historical roots of the field are inspired by behavioral psychology (Dayan & Abbott, 2001). Furthermore, it is interesting to note that RL has been used to describe the related action-perception cycle (Emmert-Streib, 2003; Sperry, 1952).

In order to establish an optimal policy, the agent faces the dilemma of exploring new states of the environment while maximizing its overall return at the same time. This dilemma is called the exploration versus exploitation tradeoff. To balance both goals, many different strategies have been developed providing different forms of learning

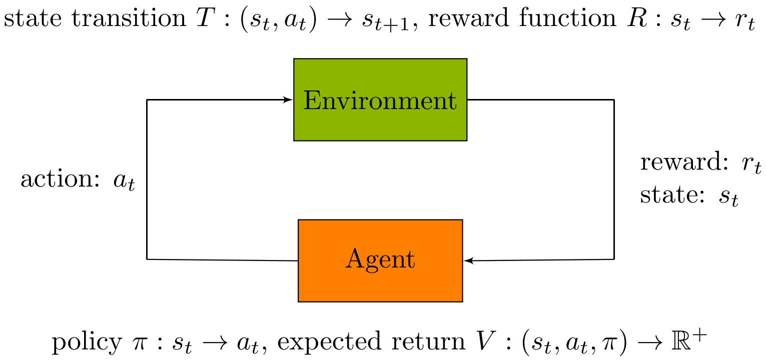

state transition $T:(s_t,a_t)\rightarrow s_{t+1}$, reward function $R:s_t\rightarrow r_t$
Environment
action: a_t
reward: r_t
state: s_t
Agent
policy $\pi:s_t\rightarrow a_t$, expected return $V:(s_t,a_t,\pi)\rightarrow\mathbb{R}^+$

FIGURE 3 Basic components of reinforcement learning. The policy, state transition and reward function define the agent and environment, respectively. The overall goal is the maximization of the expected return V of all future rewards.

approaches for this problem. For instance, by making Markovian assumptions for the transition between states one obtains a Markov Decision Process (MDP; Van Otterlo & Wiering, 2012), whereas for limited sensing capabilities of the agent the problem needs to be formulated as a Partially Observable Markov Decision Processes (POMDP; Jaakkola et al., 1995). For each assumption, practical realization has been suggested (Hauskrecht, 2000), for example, Q-learning for MDP (Watkins & Dayan, 1992) or deep variational learning for POMDP (Igl et al., 2018).

Overall, RL is quite different from all other machine learning paradigms discussed above and below due to its *generative* character. This is also reflected in its application domains. For instance, popular applications are in robotics, game playing, question-answering, trading, and recommendation systems.

3 | MODERN MACHINE LEARNING PARADIGMS

The above presentation of the traditional machine learning paradigms—supervised learning, UL, and RL—provides not only information about their basic definitions but also information on the characteristics of the underlying data. This is important because it allows a data-driven perspective. Specifically, depending on the characteristics of the data a learning paradigm can be selected or excluded. This may not be unique but at least allows a preselection of certain principle approaches.

Following this line of thought allows to specify further data with additional characteristics. The advanced machine learning paradigms discussed in the following sections can be distinguished in this way. Specifically, in the following we discuss the seven learning paradigms:

1. Multi-label learning;
2. Semi-supervised learning;
3. One-class classification;
4. Positive-unlabeled learning;
5. Transfer learning;
6. Multi-task learning;
7. Few/one-shot learning.

As we will see below, each of these learning paradigms has different requirements for the underlying data. Hence, they do not merely provide alternative algorithmic or computational approaches for existing data characteristics but establish new conceptual frameworks in the form of machine learning paradigms. In order to emphasize this aspect, we neglect in the following to a large extent algorithmic realizations or statistical estimation techniques which address numerical implementations.

Regarding the presentation order, we made the following selection. We start by discussing MLL because it is a generalization of multiclass classification and is closest related to SL. Thereafter, we present SSL, OCC, and PUL which are the next closest to the traditional learning paradigms and are also related to each other. The following two learning paradigms, TL, and MTL, are as well related to each other and are further extensions of SL. However, both paradigms require significant modifications to the traditional conceptual framework. Finally, we discuss few/OSL which is most different from all other learning paradigms. In summary, the order of the following machine learning paradigms reflects the distance to the traditional learning paradigms with respect to the extensions/modifications required. We will return to this argument in Section 12.

4 | MULTI-LABEL LEARNING

The idea of MLL is to generalize the class labels of a traditional classification having single-valued entities to variable set sizes (Tsoumakas & Katakis, 2007). Therefore, the number of labels as the outcome of a prediction function is variable.

4.1 | Definition of MLL

In order to formally define MLL, we need to modify the definition of a data set D. Specifically, for MLL, D is defined as $D = \{(x_i, Y_i)\}_{i=1}^{n}$ with $x_i \in \mathcal{X}$ and $Y_i \subseteq \mathcal{Y}$ where $\mathcal{Y} = \{L_1, ..., L_q\}$. Here, Y_i can assume any subset of $\mathcal{Y}$ which makes the size of such a set variable, that is, the size is not constant. One can represent such a Y_i as a binary vector $b = (b_1, ..., b_q)$ of length $q = |\mathcal{Y}|$ defined by

$$b_j = \begin{cases} 1, & \text{for} L_j \in Y_i \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

That means a component of b is 1 if the corresponding label is in Y_i and zero otherwise.

The goal of MLL is to find a prediction function f that maps the elements of D correctly. Formally, the task is defined as follows.

Definition 4.1. *For multi-label learning, a task $\mathcal{T}$ consists of a label space $\mathcal{Y}$ and a prediction function $f(X)$ with $f : \mathcal{X} \to 2^{\mathcal{Y}}$*, that is, $\mathcal{T} = \{\mathcal{Y}, f(X)\}$.

Here, $2^{\mathcal{Y}}$ corresponds to the power set of $\mathcal{Y}$ which is the set of all subsets of $\mathcal{Y}$.

4.2 | Methodological approaches

For MLL, there are two key conceptual approaches allowing the categorization of available methods.

1. Problem transformation.
2. Algorithm adaptation.

First, approaches based on problem transformation can be further subdivided into (i) transformation to binary classification, (ii) transformation to label ranking, and (iii) transformation to multiclass classification (Gibaja & Ventura, 2014). Such approaches convert a MLL problem by means of transformations into well-established problem settings. Examples of this are Classifier Chains (Read et al., 2011) which transforms a MLL problem into a binary classification task, Calibrated Label Ranking (Fürnkranz et al., 2008) which maps MLL into the task of label ranking, and Random k-label sets (Tsoumakas et al., 2010) which transforms MLL into the task of multiclass classification. If such a mapping is performed, it is called label powerset (Tsoumakas et al., 2009).

From the definition of MLL and the description of transformation methods, one may wonder why $2^{\mathcal{Y}}$ is not always directly mapped to a multi-class classification problem because, theoretically, such a mapping is always possible. However, there is a practical problem with this for large $|\mathcal{Y}|$. For instance, let us assume $\mathcal{Y} = \{y_1, ..., y_{20}\}$. In this case, the size of the power set is $1,048,576 \ (= 2^{20})$. Hence, if we would map the multi-label problem to a multi-class classification one would have $1,048,576$ different classes. It is clear that this can result in severe learning problems for such a classifier. For this reason, MLL tries to be more resourceful.

Second, methods based on algorithm adaptation modify existing learning methods to adopt them to the multi-label case. In M.-L. Zhang and Zhou (2013), four approaches are distinguished: (i) lazy learning (e.g., ML-kNN; M.-L. Zhang & Zhou, 2007), (ii) decision tree (e.g., ML-DT; Clare & King, 2001), (iii) kernel learning (e.g., Rank-SVM; Elisseeff & Weston, 2001), and (iv) information-theoretic methods (e.g., CML; Ghamrawi & McCallum, 2005).

Dedicated reviews of MLL and applications can be found in M.-L. Zhang and Zhou (2013) and Gibaja and Ventura (2014).

5 | SEMI-SUPERVISED LEARNING

The idea of SSL is to use both labeled and unlabeled data for performing a SL task (Chapelle et al., 2006).

5.1 | Definition of SSL

For defining SSL formally, one needs the definition of a domain $\mathcal{D}$ and a task $\mathcal{T}$ and the characterization of the data.

Definition 5.1. *A domain $\mathcal{D}$ consists of a feature space χ and a marginal probability distribution $P(X)$ where $X = \{X_1, ..., X_n\} \in \mathcal{X}$, that is, $\mathcal{D} = \{\mathcal{X}, P(X)\}$.*

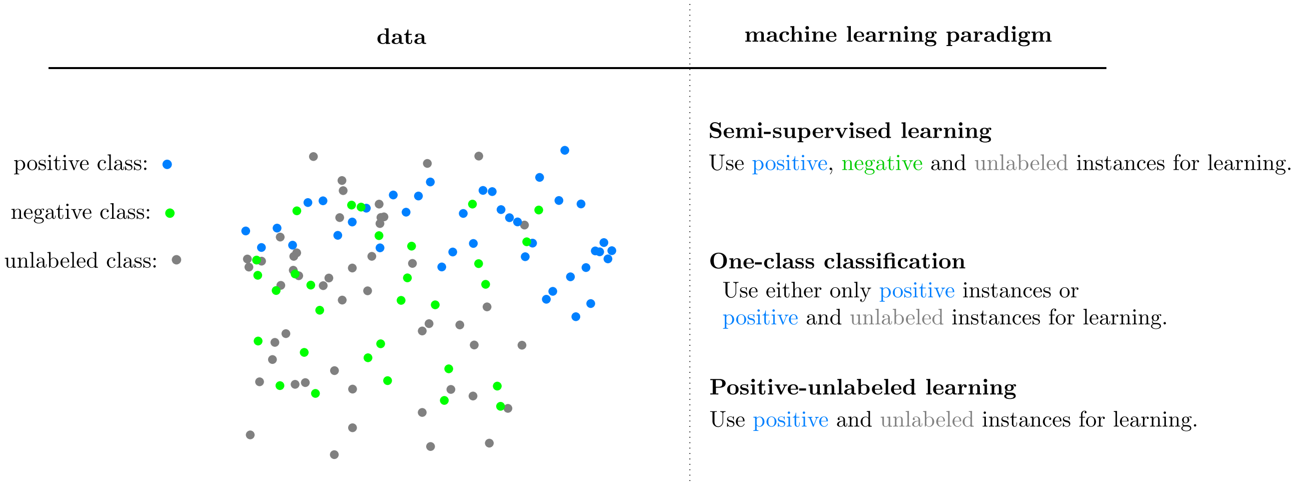
Definition 5.2. *A task $\mathcal{T}$ consists of a label space $\mathcal{Y}$ and a prediction function $f(X)$ with $f: \mathcal{X} \to \mathcal{Y}$, that is, $\mathcal{T} = \{\mathcal{Y}, f(X)\}$.*

The definition of a domain is similar to SL, however, the resulting data are different. Specifically, for SSL, there are two parts of the data, a labeled part, $D_L = \{(x_i, y_i)\}_{i=1}^{n_L}$ with $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$, and an unlabeled part $D_U = \left\{(x_j)\right\}_{j=1}^{n_U}$. This means that the available data are of the form $D = D_L \cup D_U$. In Figure 4, we visualize such data by showing data points with a positive label in blue, data points with a negative label in green, and unlabeled data points in gray.

Formally, SSL can be defined as follows.

Definition 5.3. *Given domain, $\mathcal{D}$, with task, $\mathcal{T}$, labeled data $D_L = \{(x_i, y_i)\}_{i=1}^{n_L}$ with $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$ and unlabeled data $D_U = \left\{(x_j)\right\}_{j=1}^{n_U}$, semi-supervised learning is the process of improving the prediction function, f, by utilizing the labeled and unlabeled data.*

We would like to remark that given the data D_L are labeled, SSL can be used for classification or regression problems.



Semi-supervised learning
Use positive, negative and unlabeled instances for learning.

One-class classification
Use either only positive instances or positive and unlabeled instances for learning.

Positive-unlabeled learning
Use positive and unlabeled instances for learning.

FIGURE 4 Characterization of semi-supervised learning, one-class classification, and positive-unlabeled learning. The class label of instances are distinguished by the color, that is, a positive class is blue, a negative class is green and an unlabeled class is gray.

5.2 | Methodological approaches

For SSL, a broad variety of methods have been proposed. However, there are two key concepts based on which they can be distinguished (X. Zhu & Goldberg, 2009).

1. Inductive methods.
2. Transductive methods.

Both concepts are fundamentally different from each other and the training and prediction parts of such methods are vastly different (Gammerman et al., 2013). Put simply, this can be formulated as follows.

Induction is reasoning from observed training cases to general rules, which are then applied to the test cases.

In contrast, transduction has the following meaning.

Transduction is reasoning from observed, specific (training) cases to specific (test) cases.

It is important to note that this implies that transductive learning does not distinguish between the training and testing steps of a model. Instead, it uses both the training and testing data for training the model, in contrast to inductive learning. As a consequence of this transductive learning does not build a predictive model. For this reason, in case one wants to test a new instance then one needs to train the model again for all available data. This is not necessary for inductive learning because it leads to a predictive model that can be used for new instances without re-training the model.

It is interesting to note that many transductive learning approaches are either explicitly or implicitly graph-based because the propagation of information between different data points which can be seen as nodes in a graph (W. Liu et al., 2012; Van Engelen & Hoos, 2020).

A very recent comprehensive review of SSL including details about algorithmic realizations can be found in Van Engelen and Hoos (2020).

6 | ONE-CLASS CLASSIFICATION

The idea of OCC is to distinguish instances from one particular class from instances outside this class (Moya, 1993; Moya & Hush, 1996; Tax, 2001). This is quite different from ordinary classification and for this reason, OCC has also been called outlier detection, novelty detection, anomaly detection, or concept learning (Japkowicz, 1999; Ruff et al., 2018). Hence, OCC focuses on one particular class only.

6.1 | Definition of OCC

OCC is either based on data containing only positive instances, that is, $D = D_p$ with $D_p = \{(x_i, y_i)\}_{i=1}^{n_p}$ and $x_i \in \mathcal{X}, y_i \in \mathcal{Y}$ or on data that are a combination of positive and unlabeled instances, that is, $D = D_p \cap D_u$ with $D_u = \{(x_i)\}_{i=1}^{n_u}$ and $x_i \in \mathcal{X}$. It is assumed that also the unlabeled x_i have a label but it is not known to us. We would like to remark that the case for $D = D_p \cap D_u$ is usually called PUL which we discuss in the next section.

Definition 6.1. *Given $D = D_p$, one-class learning is the process of improving the scoring function $z : x \to \mathbb{R}$ to assign novelty scores to previously unseen test instances $x \in \mathcal{X}$.*

Using such scores, a decision is made based on thresholding (Pimentel et al., 2014).

6.2 | Methodological approaches

According to Khan and Madden (2014), one-class learning approaches can be categorized with respect to the way they are using the training data. This allows to distinguish approaches utilizing only positive data from approaches that learn from positive and unlabeled data. The latter has found widespread interest and is called PUL. Due to the importance of such methods, we discuss this subcategory of one-class learning in the next section.

From a methodological point of view, there are three key concepts for OCC that use only positive-labeled data (Bartkowiak, 2011; Tax, 2001).

1. Density estimation.
2. Boundary estimation.
3. Reconstruction methods.

First, density estimation methods are estimating the density of the data points having a positive label. A new instance is classified according to a threshold (Tarassenko et al., 1995). Second, boundary estimation methods focus on setting boundaries around a small set of points, called target points. Example methods from this category utilize SVMs or neural networks (Manevitz & Yousef, 2000; Schölkopf et al., 1999).

For completeness, we would like to remark that one can also find other categorizations of methodological approaches in the literature. For instance, in Perera et al. (2021), one-class learning is divided into six categories: (i) statistical; (ii) representation-based; (iii) deep Learning-based; (iv) discriminative methods; (v) generative models; and (vi) knowledge distillation whereas in Chandola et al. (2009). OCC is divided into (i) classification-based; (ii) nearest-neighbor-based; (iii) clustering-based; (iv) statistical-based; (v) information theoretic-based; and (vi) spectral-based approaches.

It is interesting to note that OCC using only positive-labeled data for density estimation is conceptually similar to statistical hypothesis testing (Emmert-Streib & Dehmer, 2019). However, methodologically these approaches are different because OCC is not based on the concept of a sampling distribution, which specifies not only the estimation precisely but also the statistical interpretations thereof. In contrast, OCC approaches for density estimation are more broad and for this reason, vary in their interpretation considerably.

Comprehensive reviews of OCC learning can be found in Khan and Madden (2014) and Rodionova et al. (2016).

7 | POSITIVE-UNLABELED LEARNING

For PUL, we are facing a classification problem when only labeled instances of one class are available. In addition, we have unlabeled data which can come from any class but their labels are unknown. For this reason, we have labeled data from one class (termed as "positive") complemented by unlabeled data. The goal is to utilize these data for a classification task.

7.1 | Definition of PUL

For obtaining the data, we assume that n_p positive samples are randomly drawn from the marginal distribution $P(x|Y=+1)$ and n_i unlabeled samples are randomly drawn from $P(x)$ (Niu et al., 2016), resulting in the two data sets $D_p=\{(x_i,y_i)\}_{i=1}^{n_p}$ with $x_i \in \mathcal{X}, y_i \in \mathcal{Y}$ and $D_u=\{(x_i)\}_{i=1}^{n_u}$ with $x_i \in \mathcal{X}$. Hence, in total, we have the data $D=D_p \cup D_u$ with $n=n_p+n_u$ samples. Furthermore, we assume that also for $x_i \in D_u$ exist labels in $\mathcal{Y}$, however, these are not observed.

Due to the lack of observable instances for the entire label space $\mathcal{Y}$, the problem is limited to a binary label space (simplifying the complexity).

Definition 7.1. *The task $\mathcal{T}$ of positive-unlabeled learning consists of a label space $\mathcal{Y}$ and a prediction function $f(X)$ with $f:\mathcal{X}\rightarrow\mathcal{Y}$, that is, $\mathcal{T}=\{\mathcal{Y},f(X)\}$, whereas the label space $\mathcal{Y}$ is binary, that is, $|\mathcal{Y}|=2$.*

Based on this definition and the above assumptions, PUL can be formally defined as follows.

Definition 7.2. *Given $D=D_p \cup D_u$, positive-unlabeled learning is the process of improving the prediction function f of the binary task $\mathcal{T}$ utilizing D_p and D_u.*

Such approaches exploit inductive and transductive learning approaches, both of which adopt an iterative procedure to obtain reliable negative training data from the unlabeled data [Perera et al., 2021]. An example of such an inductive PU learning algorithm using bagging SVM to infer a GRN (gene regulatory network) is presented in [Mordelet & Vert, 2014].

7.2 | Methodological approaches

The main methodological approaches for PUL can be distinguished as follows.

1. Two-step methods.
2. Weighting methods.

First, the two-step methods use the unlabeled data in step one to identify negative instances, and then in step two use a traditional classifier. Second, the weighting methods estimate real valued weights for the unlabeled data and then learn a classifier based on these weights. The weights represent the likelihood, or conditional probability, that an unlabeled instance belongs to a certain class. Hence, the problem is converted into a (constrained) regression problem. Recently, a generative adversarial network (GAN) has been introduced for PU-learning called GenPU (Hou et al., 2017). GenPu consists of a number of generators and discriminators similar to a minimax game. These components generate simultaneously positive and negative samples with realistic properties which can then be used with a standard classifier.

For completeness, we would like to mention that there are also approaches that can be neither categorized as a two-step method or a weighting method. For instance, the method introduced in Yu and Li (2007) aims at increasing the number of positive instances by combining a graph-based SSL method with the two-step approach.

For comprehensive reviews of PUL, the reader is referred to Bekker and Davis (2020), Jaskie and Spanias (2019), and B. Zhang and Zuo (2008).

8 | TRANSFER LEARNING

The basic idea of TL is to utilize information from one task to improve the learning for a second one. In order to distinguish the two tasks from each other, the former is called the source task and the latter target task. Correspondingly, for each task, there is a domain and data distinguished in a similar way. In Figure 5, we show a visualization of the underlying idea of TL.

8.1 | Definition of TL

Similar to SL (see above) also for TL we need the definition of a domain, $\mathcal{D}$, and a task, $\mathcal{T}$.

Definition 8.1. *A domain $\mathcal{D}$ consists of a feature space χ and a marginal probability distribution $P(X)$ where $X = \{X_1, ..., X_n\} \in \mathcal{X}$, that is, $\mathcal{D} = \{\mathcal{X}, P(X)\}$.*

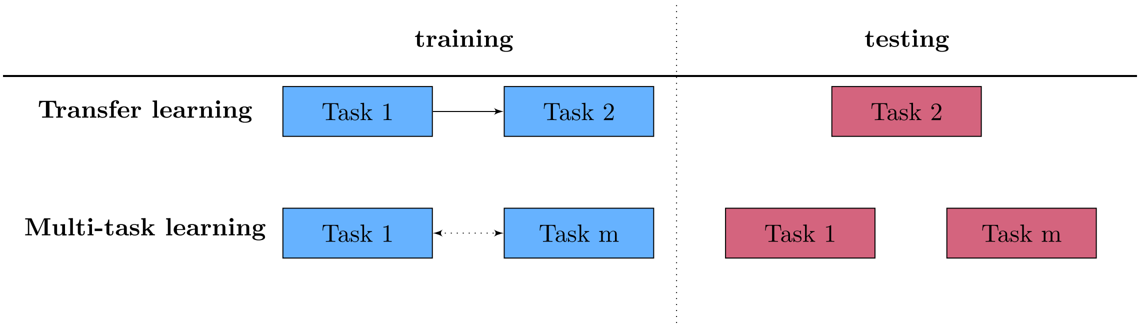


FIGURE 5 Visualization of training and testing for transfer learning (top) and multi-task learning (bottom). For transfer, learning task 1 is usually called source task and task 2 target task. A crucial difference between transfer learning and multi-task learning is that for the latter all tasks are equal, whereas the former focuses only on task 2 (the target task). Furthermore, it is important to note that for multi-task learning all tasks are evaluated independently from each other.

Definition 8.2. *A task $\mathcal{T}$ consists of a label space $\mathcal{Y}$ and a prediction function $f(X)$ with $f:\mathcal{X}\rightarrow\mathcal{Y}$, that is, $\mathcal{T}=\{\mathcal{Y},f(X)\}$.*

The prediction function $f(X)$ is learned from a data set $D=\{(x_i,y_i)\}_{i=1}^{n}$ with $x_i\in\mathcal{X}$ and $y_i\in\mathcal{Y}$ and $i\in\{1,\ldots,n\}$ where n is the sample size. Some machine learning methods do explicitly provide probabilistic estimates of f in the form of conditional probability distributions, that is, $f(X)=P(Y|X)$. Hence, this is a generalized form of a prediction function because in the deterministic case this reduces to a delta distribution $\delta_{x,y}$ with

$$\delta_{x,y_i}=\begin{cases}1 & \text{if } x=x_i \text{ with } (x_i,y_i)\\ 0 & \text{otherwise}\end{cases} \tag{4}$$

For TL, one needs to distinguish between two kinds of domains and tasks which are called source domain, $\mathcal{D}_S$, and source task, $\mathcal{T}_S$, and target domain, $\mathcal{D}_T$, and target task, $\mathcal{T}_T$ with corresponding source data, D_S, and target data, D_T. From these one can now formally define TL.

Definition 8.3. *Given a source domain, $\mathcal{D}_S$, with source task, $\mathcal{T}_S$, and target domain, $\mathcal{D}_T$, with target task, $\mathcal{T}_T$, transfer learning is the process of improving the prediction function, f_T, of the target task by utilizing $\mathcal{D}_S$ and $\mathcal{T}_S$.*

The above definition is quite general in the sense that it does not specify various aspects. Hence, specifying these leads to different subtypes of TL. In the following, we distinguish various subtypes from each other.

- *Case*: *$\mathcal{D}_S=\mathcal{D}_T$ and $\mathcal{T}_S=\mathcal{T}_T$*: This corresponds to the traditional machine learning setting when we learn f_S from source data D_S and continue the learning process with target data D_T where the resulting prediction function is renamed to f_T. From this follows that TL is obtained from $\mathcal{D}_S\neq\mathcal{D}_T$ or $\mathcal{T}_S\neq\mathcal{T}_T$. Here it is important to emphasize the "or" between the conditions which results in three different cases.
- *Case*: $\mathcal{D}_S\neq\mathcal{D}_T$: Given that $\mathcal{D}_S=\{\mathcal{X}_S,P_S(X)\}$ and $\mathcal{D}_T=\{\mathcal{X}_T,P_T(X)\}$ this can either correspond to $\mathcal{X}_S\neq\mathcal{X}_T$ or $P_S(X)\neq P_T(X)$.
    - Homogeneous TL: The case when the feature space of the source domain and target domain are the same, that is, $\mathcal{X}_S=\mathcal{X}_T$, is called homogeneous TL.
    - Heterogeneous TL: The case when the feature space of the source domain and target domain are different, that is, $\mathcal{X}_S\neq\mathcal{X}_T$, is called heterogeneous TL.
    - $P_S(X)\neq P_T(X)$.
- Case: $\mathcal{T}_S\neq\mathcal{T}_T$: Given that $\mathcal{T}_S=\{\mathcal{Y}_S,f_S(X)\}$ and $\mathcal{T}_T=\{\mathcal{Y}_T,f_T(X)\}$ this can either correspond to $\mathcal{Y}_S\neq\mathcal{Y}_T$ or $f_S(X)\neq f_T(X)$.
    - $\mathcal{Y}_S\neq\mathcal{Y}_T$: This case means that the label space of the source task and the target task is different. For instance, this can be due to a different number of classes in the source task and target task.
    - $f_S(X)\neq f_T(X)$: Given that the prediction functions generalize to conditional probability distributions this means $P_S(Y|X)\neq P_T(Y|X)$.

8.2 | Methodological approaches

For TL, a variety of different perspectives have been suggested for the categorization of this learning paradigm. For instance, one could assume a view with respect to traditional paradigms distinguishing between inductive, transductive, and unsupervised TL (Pan & Yang, 2009) or a model-based view (Zhuang et al., 2020). However, the most common categorization is based on "what to transfer" (Pan & Yang, 2009).

1. Feature-based TL.
2. Parameter-based TL.
3. Instance-based TL.
4. Relational-based TL.

(1) For feature-based TL good feature representations are learned from the source task and assumed to be useful for the target task as well. Hence, in this case, the knowledge transfer between source task and target task is via learning feature representations. (2) For parameter-based TL some parameters or prior distribution of hyperparameters are transferred from the source task to the target task. This assumes a similarity between the source model and the target model. Unlike multitask learning, where both the source and target tasks are learned simultaneously, for TL, we may apply additional weightage to the loss of the target domain to improve overall performance. (3) The idea of instance-based TL is to reuse parts of the instances from the source task for the target task. Usually, instances cannot be used directly, instead, this is accomplished via instance weighting. (4) Relational-based TL assumes that instances are not independent and identically distributed but they are dependent. This implies that the underlying data form some kind of network, for example, a transcription regulatory network or a social network.

Comprehensive reviews of TL can be found in Bashath et al. (2022), Pan and Yang (2009), Weiss et al. (2016), and Zhuang et al. (2020).

9 | MULTI-TASK LEARNING

The idea of MTL compared to TL is two-fold. First, instead of considering exactly 2 tasks, the source and target task, in MTL, there can be $m > 2$ tasks. Second, these m tasks do not have one or more dedicated targets but all tasks are equally important. That means that there are m source tasks and m target tasks (Caruana, 1997).

9.1 | Definition of MTL

Formally, MTL can be described as follows.

> **Definition 9.1.** *Given m learning tasks, $\{\mathcal{T}_k\}_{k=1}^m$, where all tasks or a subset of tasks are related, multi-task learning aims to improve each learning task $\mathcal{T}_k$ by utilizing information from some or all other models.*

For clarity, we would like to emphasize that for each learning task $\mathcal{T}_k$, there is a corresponding domain $\mathcal{D}_k = \{\mathcal{X}_k, P(X_k)\}$ and data set D_k given, from which information can be utilized. In the following, we denote the data set of task k by $D_k = \left\{\left(x_{ki}, y_{ki)}\right\}_{i=1}^{n_k}$ with $x_{ki} \in \mathcal{X}_k$ and $y_{ki} \in \mathcal{Y}_k$ and $i \in \{1, \ldots, n_k\}$ where n_k is the sample size.

- Case: The case where $x_{ki} = x_{li}$ and $n_k = n_l = n$ for all $k, l \in \{1, \ldots, m\}$ and $i \in \{1, n\}$, is called *multi-view learning*. Therefore in this case the x-values of the data D_k for all tasks are identical but can have different labels, that is, $\mathcal{Y}_k \neq \mathcal{Y}_l$ for all $k, l \in \{1, \ldots, m\}$.

9.2 | Methodological approaches

For MTL, there are three key methodological approaches used to study such problems (Y. Zhang & Yang, 2018).

1. Feature-based MTL.
2. Parameter-based MTL.
3. Instance-based MTL.

First, feature-based MTL models assume that different tasks share the same or at least similar features. This includes also methods that perform feature selection or transformation of the original features. Second, parameter-based MTL models utilize parameters between different models to relate the learning between different tasks. Examples of this include methods based on regularization or priors on model parameters. In general, this conceptual approach is very diverse with many different realizations. Third, instance-based MTL models estimate weights for the membership of instances in tasks and then use all instances for learning all tasks in a weighted manner.

Comprehensive reviews of MTL can be found in Y. Zhang and Yang (2018), Ruder (2017), and Sosnin et al. (2019).

10 | FEW/ONE-SHOT LEARNING

The idea of few/OSL is to utilize a (large) training set for learning a similarity function which is then used in combination with a very small data set containing only one or a few instances about unknown classes to make predictions about these unknown classes (Fei-Fei et al., 2006). Hence, few/OSL utilizes semantic information from the training data to deal with few/one instances for new classes that are unknown from the training data. In Figure 6, we summarize the idea of few/OSL.

10.1 | Definition of OSL

Few/OSL utilizes three key components. (1) A labeled data set D. (2) A support set D_{Su}. (3) A query q representing a new instance for which a class label should be predicted. The labeled data D is given by $D = \{(x_i, y_i)\}_{i=1}^{n}$ with $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$ and $i \in \{1, ..., n\}$, n is the sample size, $\mathcal{X}$ feature space and $\mathcal{Y}$ label space. If the cardinality of the label space is larger than two, that is, $|\mathcal{Y}| > 2$, then we have a multi-class classification problem, otherwise a binary classification. The data set D serves as a training data to learn a similarity function g. This similarity function will then be used for evaluating the similarity of a query q to instances given in support set D_{su}. The support set D_{su} is defined as follows.

Definition 10.1. *A support set D_{su} is a labeled data set $D_{su} = \left\{\{(x_i, y_i)\}_{i=1}^{n_s}\right\}_{s=1}^{S}$ providing information about labeled instances of S classes with $y_i \in \mathcal{Y}'$. For $n_1 = \cdots = n_S = 1$ one obtains one-shot learning and for $n_i > 1$ for all $i \in \{1, ..., S\}$ with $|n_i|$ small, few-shot learning. For $n_1 = \cdots = n_S = n$, this is called n-shot, S-way learning.*

It is important to note that the label space of the support set D_{su} and the training data D are different, that is, $\mathcal{Y} \neq \mathcal{Y}'$. Hence, the semantic transfer from the training data is accomplished via the similarity function and the support set is utilized as a kind of dictionary to *look-up* the similarity with the query q. In this way, it is possible to make predictions about new classes that have not been present in the training data.

The task that is important for few/OSL is to learn a prediction function, $f_{su}: \mathcal{X} \to \mathcal{Y}'$, that maps into the classes given by $\mathcal{Y}'$, not $\mathcal{Y}$.

Definition 10.2. *The task $\mathcal{T}_{su}$ for few/one-shot learning consists of outcome space $\mathcal{Y}'$ and prediction function $f_{su}(X)$ with $f_{su}: \mathcal{X} \to \mathcal{Y}'$, that is, $\mathcal{T}_{su} = \{\mathcal{Y}', f_{su}(X)\}$.*

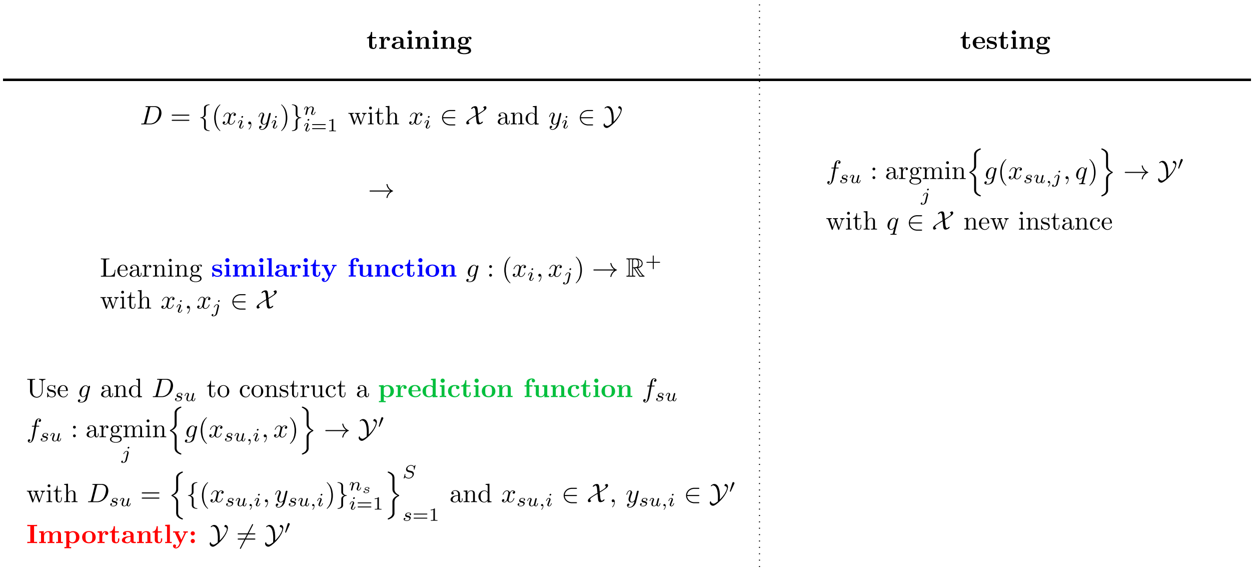

training
testing
$D = \{(x_i, y_i)\}_{i=1}^{n}$ with $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$
$\rightarrow$
Learning similarity function $g : (x_i, x_j) \rightarrow \mathbb{R}^+$
with $x_i, x_j \in \mathcal{X}$
Use g and D_{su} to construct a prediction function f_{su}
$f_{su} : \underset{j}{\text{argmin}}\left\{g(x_{su,i}, x)\right\} \rightarrow \mathcal{Y}'$
with $D_{su} = \left\{\{(x_{su,i}, y_{su,i})\}_{i=1}^{n_s}\right\}_{s=1}^{S}$ and $x_{su,i} \in \mathcal{X}$, $y_{su,i} \in \mathcal{Y}'$
Importantly: $\mathcal{Y} \neq \mathcal{Y}'$
$f_{su} : \underset{j}{\text{argmin}}\left\{g(x_{su,j}, q)\right\} \rightarrow \mathcal{Y}'$
with $q \in \mathcal{X}$ new instance

FIGURE 6 Overview of few/one-shot learning. There are three key components: (1) Labeled data set D. (2) Support set D_{Su} with $\mathcal{Y}' \neq \mathcal{Y}$. (3) Query q representing a new instance for which a class label should be predicted. For the testing, the prediction function f_{su} is used to evaluate the similarity between q and the instances in the support set D_{Su}.

The distinction between $\mathcal{Y}'$ and $\mathcal{Y}$ may appear strange at first because it means the classes of the training data and the testing data are different. So how can one learn from the instances provided by the training data for the testing data when the outcome spaces are entirely different? The trick of few/OSL is to assume that the similarity among instances in the training data and the testing data are similar. Hence, learning such a similar function in the form of the function g allows to learn from the training data for the testing data despite the fact that $\mathcal{Y}' \neq \mathcal{Y}$.

We would like to remark that the above assumption about the similarity among instances in the training data and the testing data determines the quality of the outcome. Specifically, for infinitely large training data, it should be possible to learn the similarity function g with high accuracy. However, in the case when the similarity in the testing data are not captured by g, the prediction function f_{su} will not be able to provide meaningful results. Strictly, this is true irrespective of the sample size of the training data and the number of instances in the support set. Hence, if the similarity assumption is violated no learning occurs even within the limit of infinite large sample sizes.

Based on the above definitions, few/OSL can now be defined as follows.

Definition 10.3. *Given a training data set D and a support set D_{su}, few/one-shot learning is the process of improving a prediction function, $f_{su} : \mathcal{X} \rightarrow \mathcal{Y}'$, for task $\mathcal{T}_{su}$ by utilizing D and D_{su}.*

10.2 | Methodological approaches

In order to establish a few/OSL model, there are essentially two main conceptual approaches.

1. Semantic transfer via similarities.
2. Semantic transfer via features.

First, semantic transfer via similarities means that knowledge extracted from the training data is utilized for unknown classes via learning similarity concepts. An example of this is the Siamese network used in Koch et al. (2015). Here, the authors learn an image verification task instead of predicting the classes of instances directly. Conceptually, this means to learn the similarity (or lack thereof) between pairs of instances. This network is trained for D and then utilized with D_{su} whereas an instance from D_{su}, that is, $x_{su,i}$, is used together with a query x. If x is similar to $x_{su,i}$ then the predicted class is $y_{su,i}$. Second, semantic transfer via features has been suggested by Bart and Ullman (2005). The authors showed that the similarity of novel features to existing features learned from training data can help in feature adaptation.

Recently, deep learning approaches have been used. For instance, in Vinyals et al. (2016), a neural architecture called Matching Networks has been introduced that utilizes an augmented memory including an attention kernel. Another example is Relation Network (RelNet) introduced in Sung et al. (2018). RelNet learns an embedding and a deep nonlinear distance metric with a convolutional neural network for comparing query and sample items.

Reviews of few/OSL can be found in Y. Wang et al. (2020), Kadam and Vaidya (2018), X. Li et al. (2020), and Rezaei and Shahidi (2020).

11 | APPLICATIONS

From the above definitions and descriptions of the individual learning paradigms, it seems clear that their application potential is enormous due to the extended flexibility offered by these modern paradigms. In order to underline this impression, we provide in Table 1 a brief overview of real-world application domains to which these learning paradigms have been already applied to. Specifically, this table provides some example studies for four different kinds of data, namely, biomedical data, text data, sensor data, and image data.

We used these four data categories to cover a large range of fields from which such data can arise. To make this more clear, we searched the Web of Science to obtain the number of publications from such fields. Specifically, we searched the number of publications for applications in engineering, telecommunications, image science, automation, biology, acoustics, robotics, and materials. The results of this search are shown in Figure 7. There we show detailed information for SSL, OCC, PUL, OSL, TL, MTL, and MLL. We would like to note that the y-axis is again on a logarithmic (base 10) scale due to the large differences in the observed number of publications.

TABLE 1 An overview of real-world application domains for semi-supervised learning (SSL), one-class classification (OCC), positive-unlabeled learning (PUL), one-shot learning (OSL), transfer learning (TL), multi-task learning (MTL), and multi-label learning (MLL)

Paradigms/ data	Biomedical data	Text data	Sensor data	Image data
SSL	(Shi & Zhang, 2011; Xia et al., 2010)	(Devlin et al., 2019; Lee et al., 2019)	(Pulkkinen et al., 2011)	(Guillaumin et al., 2010)
OCC	(Alashwal et al., 2006)	(Koppel & Schler, 2004)	(Das et al., 2016)	(Ruff et al., 2018)
PUL	(Yang et al., 2014)	(X.-L. Li et al., 2009)	(Wu et al., 2020)	(Kiryo et al., 2017)
OSL	(Altae-Tran et al., 2017)	(Yan et al., 2018)	(Feng & Duarte, 2019)	(Lake et al., 2011)
TL	(Mignone et al., 2020)	(Do & Ng, 2005)	(Hu & Yang, 2011)	(Y. Zhu et al., 2011)
MTL	(Bi et al., 2008)	(P. Liu et al., 2016)	(Peng et al., 2018)	(Han et al., 2017)
MLL	(Y.-X. Li et al., 2011)	(Schapire & Singer, 2000)	(Kayaalp et al., 2017)	(Wu et al., 2015)

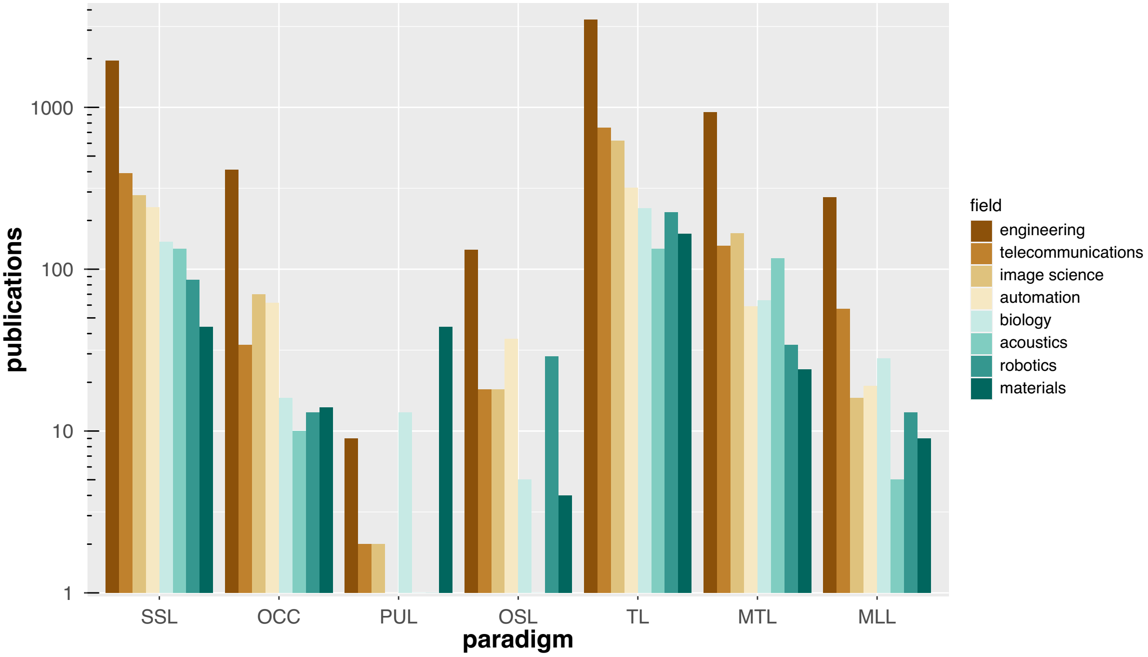


FIGURE 7 Number of published articles in different application fields about semi-supervised learning (SSL), one-class classification (OCC), positive-unlabeled learning (PUL), one-shot learning (OSL), transfer learning (TL), multi-task learning (MTL), and multi-label learning (MLL). The numbers are from web of science. The *y*-axis is in a logarithmic (base 10) scale

Comparing the scale of the *y*-axis in Figure 7 with Figure 1 indicates that all of the modern learning paradigms, especially PUL or OSL, are severely underutilized in essentially all fields. We hypothesize the main reason for this difference is not due to the inadequacy of particular learning paradigms for certain application fields but the lack of knowledge of the application-oriented communities due to the inaccessibility of the presentation of modern learning paradigms. Importantly, for the application-oriented communities, algorithmic details of methods or learning paradigms are in general as less administrable than information about the applicability of such methods or learning paradigms to particular data. For this reason, in this article, we assumed a data-driven perspective making it easy to decide if a particular learning paradigm is basically suited for analyzing a given data set. This allows to narrow down all options so one can then focus on the remaining learning paradigms and the selection of appropriate methods.

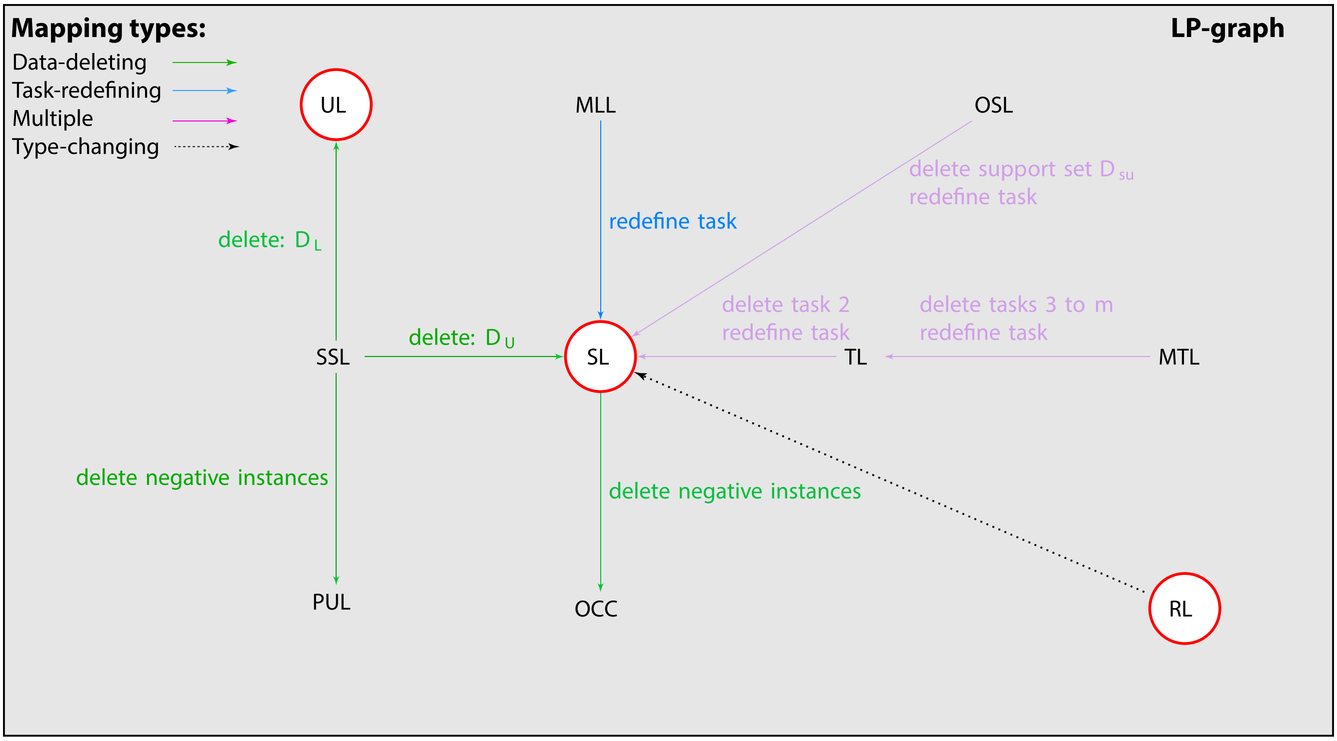


FIGURE 8 The shown diagram, called the learning-paradigm graph (LP-graph), provides information about connections between the different machine learning paradigms. The acronyms correspond to supervised learning (SL), unsupervised learning (UL), reinforcement learning (RL), multi-label learning (MLL), semi-supervised learning (SSL), one-class classification (OCC), positive-unlabeled learning (PUL), transfer learning (TL), multi-task learning (MTL), and one-shot learning (OSL). Two nodes in the diagram are connected via a specific type of mapping (see main text) indicated by the color.

12 | INTERRELATIONS BETWEEN MACHINE LEARNING PARADIGMS

After reaching conceptual clarity of the different learning paradigms, their definitions, and applications, we study now the relations between them.

In Figure 8, we show an interrelation diagram for all 10 machine learning paradigms. In this figure, the used acronyms correspond to SL, UL, RL, MLL, SSL, OCC, PUL, TL, MTL, and OSL. We constructed this diagram by defining mappings between the learning paradigms. Specifically, if there is a relation between two paradigms in the form of a mapping then there is an edge connecting these. Each edge has a direction defining a start node (S), an end node (E), and a label. Overall, the interrelation diagram defines a directed, labeled graph, we call the LP-graph.

For the mappings between the learning paradigms, one needs to distinguish between different cases corresponding to different types of mappings. In the following, we distinguish between four different mapping types.

1. *Data-deleting mapping*: This type of mapping deletes data from the available set of data of a learning paradigm. Specifically, in order to map from SSL to UL, PUL, and SL and from SL to OCC one can define the following data-deleting mappings.

$$d_{SSL\to UL}: D = D_L \cup D_U \to D_U \tag{5}$$

$$d_{SSL\to PUL}: D = D_L \cup D_U \to D_{L^+} \cup D_U \tag{6}$$

$$d_{SSL\to SL}: D = D_L \cup D_U \to D_L \tag{7}$$

$$d_{SL\to OCC}: D = D_{L^+} \cup D_{L^-} \to D_{L^+} \tag{8}$$

As one can see, each of the above mappings deletes a part of the available data given by D. Due to the fact that different learning paradigms are based on different data, the meaning of D is paradigm specific.

2. *Task-redefining mapping*: The second type of mapping does not require a deletion of data but a redefinition of a task. In Figure 8, this occurs just once from MLL to SL.

$$d_{MLL \to SL}: \text{Convert a multi-label designation into multi-class categories.} \tag{9}$$

3. *Multiple mapping (data-deleting and task-redefining)*: The third type of mapping provides simultaneous data-deleting and task-redefining mapping. Such a mapping is required to connect TL → SL, MTL → TL, and OSL → SL.

$$d_{TL \to SL}: D_S \cup D_T \to D_T \tag{10}$$

Redefining the task.

$$d_{MTL \to TL}: D_{T_1} \cup \ldots \cup D_{T_m} \to D_{S_i} \cup D_{T_j} \tag{11}$$

Redefining the task.

Here, i and j with $i \neq j$ correspond to just one task from $\{1, \ldots, m\}$. For simplicity, one can assume $i = 1$ and $j = 2$.

$$d_{OSL \to SL}: D \cup D_{su} \to D \tag{12}$$

Redefining the task.

This type of mapping is more complex because the redefinition of the task is a significant deviation from the original problem.

4. *Type-changing mapping*: The fourth type of mapping is the most severe one in the sense that it changes the characteristics of a learning paradigm entirely. This mapping is required for RL → SL.

In order to obtain a better understanding of the LP-graph and its mappings, we discuss in the following some of its implications.

The label of an edge in the LP-graph provides information about the type of a mapping: In general, a label provides information that needs to be deleted/changed from node S to node E. Hence, all mappings correspond to a reduction of information from node S to node E. For instance, in order to obtain UL from SSL one needs to delete the data providing labeled information called D_L from the formulation of the SSL problem.

Considering the direction of a mapping, there is only one directed edge that starts from SL (see Figure 8). In this case, information from SL toward OCC needs to be deleted whereas for all other mappings involving SL the deletion of information occurs toward SL. We would like to remind that each mapping has reductionist properties, that is, a mapping deletes information in the form of available data and potentially redefines tasks. Hence, the inverse of such a mapping would require the addition or creation of information in the form of data. Theoretically, this is of course possible, however, we consider the deletion/reduction of information the simpler and more natural perspective because the addition/creation of information is more demanding with respect to the explanation of the origin of this new information. Nevertheless, a revised perspective may be fruitful despite the fact that we did not find good arguments therefor.

MTL, TL, and OSL require the redefinition of a task: In addition to the deletion of data, MTL, TL, and OSL require also a redefinition of the underlying task. This is emphasized by the purple mappings in Figure 8. For instance, for the mapping from MTL to TL one needs to delete the tasks 3 to m including the underlying data D_3 to D_m. Furthermore, one needs to redefine the target task which includes for TL only task 2 but not task 1. Similarly, for the mapping from OSL to SL one needs to delete the support data D_{su} and redefine the task because new instances from $\mathcal{Y}'$ can no longer be studied.

The mapping for RL is a pseudo-mapping: The mapping between RL and SL requires severe modifications. One reason for this is the explorative character of RL that requires the agent to choose actions (make decisions) for changing the state of the environment. This process can be seen as a data generation which is entirely absent in any other learning paradigm. Instead, all other learning paradigms assume that the data are already given.

Interestingly, a proof-of-concept for such a mapping has been provided by Wiering et al. (2011). Specifically, the authors used RL together with definitions for actions and the reward function to perform a standard classification task. Their idea is to define an intricate Classification Markov Decision Process (CMDP) that models the classification task as a sequential decision-making problem by allowing the agent to explore the input data. If such an approach has a substantial advantage over traditional classification methods is currently unclear and remains to be seen.

Meta-learning is not a learning paradigm but a meta-learning paradigm: One may wonder why meta learning (Thrun & Pratt, 1998; Vanschoren, 2019) has not been included in the LP-graph. The reason for this is that meta learning, which is also called “learning to learn” or “lifelong learning,” is more than a learning paradigm. Specifically, it assumes two interrelated learning mechanisms, that is, an inner mechanism for a base learner and an outer mechanism that helps the inner mechanism to learn and improve (Hospedales et al., 2020) whereas the outer mechanism leads to a kind of knowledge transfer for the inner mechanism. The iterations of the interaction between outer and inner learning mechanism, which are called episodes, are an important element of meta learning. Due to this iterative element, meta learning has been interpreted as evolutionary principle of learning (Schmidhuber, 1987). Hence, all this establishes meta learning as a meta paradigm of learning paradigms.

From the explanation above, it should be clear that none of the 10 learning paradigms discussed in this article are genuinely within a meta learning framework despite the fact that TL and MTL contain knowledge transfer elements. However, the missing part is the iteration establishing many episodes of learning.

The LP-graph is a taxonomy of learning paradigms: The LP-graph shown in Figure 8 provides a bird's-eye-view of the relations between the different machine learning paradigms. Given the fact that each learning paradigm itself represents a particular problem class based on dedicated definitions (Section 2.2 to 10) and applications (Section 11) which can be quite sophisticated, the LG-graph masks this complexity by projecting transitions between the learning paradigms. Hence, the LP-graph provides information not contained in any individual learning paradigm but the collective thereof. In addition, the structure of the LP-graph organizes the learning paradigm hierarchically due to the direction of the mappings whereas a key element of the mappings is the data-centric perspective.

13 | CONCLUSIONS

In this article, we provided a discussion of 10 machine learning paradigms. Specifically, we defined key constituents of SL, UL, RL, MLL, SSL, OCC, PUL, TL, MTL, and OSL and their data requirements. Our data-driven perspective allowed a systematic identification of relations between the individual learning paradigms in the form of a LP-graph. This established a taxonomy among the seven modern learning paradigms and the three traditional paradigms, that is, SL, UL, and RL. Overall, the joint presentation and discussion of all those machine learning paradigms should allow a wider appreciation of the broader community of modern learning paradigms and foster their applications.

AUTHOR CONTRIBUTIONS

Frank Emmert-Streib: Conceptualization (lead); visualization (lead); writing – original draft (equal); writing – review and editing (equal). **Matthias Dehmer:** Conceptualization (supporting); writing – original draft (equal); writing – review and editing (equal).

CONFLICT OF INTEREST

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

DATA AVAILABILITY STATEMENT

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

ORCID

Frank Emmert-Streib https://orcid.org/0000-0003-0745-5641

REFERENCES

Alashwal, H., Deris, S., & Othman, R. M. (2006). One-class support vector machines for protein-protein interactions prediction. *International Journal of Biological and Medical Sciences*, *1*(2), 120–127.

Altae-Tran, H., Ramsundar, B., Pappu, A. S., & Pande, V. (2017). Low data drug discovery with one-shot learning. *ACS Central Science*, *3*(4), 283–293.

Bart, E., & Ullman, S. (2005). Cross-generalization: Learning novel classes from a single example by feature replacement. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, June 20 2005 to June 26, 2005. San Diego, CA (Vol. *1*, pp. 672–679).

Bartkowiak, A. M. (2011). Anomaly, novelty, one-class classification: A comprehensive introduction. *International Journal of Computer Information Systems and Industrial Management Applications*, *3*(1), 61–71.

Bashath, S., Perera, N., Tripathi, S., Manjang, K., Dehmer, M., & Emmert-Streib, F. (2022). A data-centric review of deep transfer learning with applications to text data. *Information Sciences*, *585*, 498–528.

Bekker, J., & Davis, J. (2020). Learning from positive and unlabeled data: A survey. *Machine Learning*, *109*(4), 719–760.

Bi, J., Xiong, T., Yu, S., Dundar, M., & Rao, R. B. (2008). An improved multi-task learning approach with applications in medical diagnosis. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (pp. 117–132).

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Capra, F. (1996). *The web of life: A new scientific understanding of living systems*. Anchor.

Carbonell, J. G., Michalski, R. S., & Mitchell, T. M. (1983). An overview of machine learning. In R. S. Michalski, J. G. Carbonell, & T. M. Mitchell (Eds.), *Machine Learning. Symbolic Computation* (pp. 3–23). Springer.

Caruana, R. (1997). Multitask learning. *Machine Learning*, *28*(1), 41–75.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, *41*(3), 1–58.

Chapelle, O., Schölkopf, B., & Zien, A. (2006). *Semi-supervised learning*. The MIT Press.

Clare, A., & King, R. D. (2001). Knowledge discovery in multi-label phenotype data. In *European Conference on Principles of Data Mining and Knowledge Discovery* (pp. 42–53).

Das, B., Cook, D. J., Krishnan, N. C., & Schmitter-Edgecombe, M. (2016). One-class classification based real-time activity error detection in smart homes. *IEEE Journal of Selected Topics in Signal Processing*, *10*(5), 914–923.

Dayan, P., & Abbott, L. (2001). *Theoretical neuroscience: Computational and mathematical modelling of neural systems*. MIT Press.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*: *Human Language Technologies*, *Vol. 1 (long and short papers)* (pp. 4171–4186).

Do, C. B., & Ng, A. Y. (2005). Transfer learning for text classification. *Advances in Neural Information Processing Systems*, *18*, 299–306.

Dwyer, D. B., Falkai, P., & Koutsouleris, N. (2018). Machine learning approaches for clinical psychology and psychiatry. *Annual Review of Clinical Psychology*, *14*, 91–118.

Elisseeff, A., & Weston, J. (2001). A kernel method for multi-labelled classification. *Advances in Neural Information Processing Systems*, 14, 1-7.

Emmert-Streib, F. (2003). *Aktive computation in offenen systemen. Lerndynamiken in biologischen systemen: Vom netzwerk zum organismus (Unpublished doctoral dissertation)*. University of Bremen.

Emmert-Streib, F., & Dehmer, M. (2019). Understanding statistical hypothesis testing: The logic of statistical inference. *Machine Learning and Knowledge Extraction*, *1*(3), 945–961. https://doi.org/10.3390/make1030054

Fei-Fei, L., Fergus, R., & Perona, P. (2006). One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(4), 594–611.

Feng, S., & Duarte, M. F. (2019). Few-shot learning-based human activity recognition. *Expert Systems with Applications*, *138*, 112782.

Flach, P. (2012). *Machine learning: The art of science and algorithms that make sense of data*. Cambridge University Press.

Fürnkranz, J., Hüllermeier, E., Loza Mencía, E., & Brinker, K. (2008). Multilabel classification via calibrated label ranking. *Machine Learning*, *73*(2), 133–153.

Gammerman, A., Vovk, V., & Vapnik, V. (2013). Learning by transduction. *arXiv preprint arXiv:1301.7375*.

Ghamrawi, N., & McCallum, A. (2005). Collective multi-label classification. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management* (pp. 195–200).

Gibaja, E., & Ventura, S. (2014). Multi-label learning: A review of the state of the art and ongoing research. *WIREs Data Mining and Knowledge Discovery*, *4*(6), 411–444.

Guillaumin, M., Verbeek, J., & Schmid, C. (2010). Multimodal semi-supervised learning for image classification. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 902–909).

Han, H., Jain, A. K., Wang, F., Shan, S., & Chen, X. (2017). Heterogeneous face attribute estimation: A deep multi-task learning approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(11), 2597–2609.

Haste, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference and prediction*. Springer.

Hauskrecht, M. (2000). Value-function approximations for partially observable Markov decision processes. *Journal of Artificial Intelligence Research*, *13*, 33–94.

Hoaglin, D., Mosteller, F., & Tukey, J. (1983). *Understanding robust and exploratory data analysis*. Wiley.

Hospedales, T., Antoniou, A., Micaelli, P., & Storkey, A. (2020). Meta-learning in neural networks: A survey. *arXiv preprint arXiv:2004.05439*.

Hou, M., Chaib-Draa, B., Li, C., & Zhao, Q. (2017). Generative adversarial positive-unlabelled learning. *arXiv preprint arXiv:1711.08054*.

Hu, D., & Yang, Q. (2011). Transfer learning for activity recognition via sensor mapping. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, *Barcelona*, *Catalonia*, *Spain* (p. 1962).

Igl, M., Zintgraf, L., Le, T. A., Wood, F., & Whiteson, S. (2018). Deep variational reinforcement learning for POMDPs. In *International Conference on Machine Learning* (pp. 2117–2126).

Jaakkola, T., Singh, S. P., & Jordan, M. I. (1995). Reinforcement learning algorithm for partially observable Markov decision problems. In *Advances in Neural Information Processing Systems* (pp. 345–352). MIT Press.

Japkowicz, N. (1999). *Concept-learning in the absence of counter-examples: An autoassociation based approach to classification (Unpublished doctoral dissertation)*. State University of New Jersey.

Jaskie, K., & Spanias, A. (2019). Positive and unlabeled learning algorithms and applications: A survey. In *2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA)* (pp. 1–8).

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, *349*(6245), 255–260.

Kadam, S., & Vaidya, V. (2018). Review and analysis of zero, one and few shot learning approaches. In *International Conference on Intelligent Systems Design and Applications* (pp. 100–112).

Kaelbling, L., Littman, M., & Moore, A. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, *4*, 237–285.

Kaushik, V., & Walsh, C. A. (2019). Pragmatism as a research paradigm and its implications for social work research. *Social Sciences*, *8*(9), 255.

Kayaalp, F., Zengin, A., Kara, R., & Zavrak, S. (2017). Leakage detection and localization on water transportation pipelines: A multi-label classification approach. *Neural Computing and Applications*, *28*(10), 2905–2914.

Khan, S. S., & Madden, M. G. (2014). One-class classification: Taxonomy of study and review of techniques. *The Knowledge Engineering Review*, *29*(3), 345–374.

Kiryo, R., Niu, G., Plessis, M. C. d., & Sugiyama, M. (2017). Positive-unlabeled learning with non-negative risk estimator. *arXiv preprint arXiv:1703.00593*.

Koch, G., Zemel, R., & Salakhutdinov, R. (2015). Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop* (Vol. 2).

Koppel, M., & Schler, J. (2004). Authorship verification as a one-class classification problem. In *Proceedings of the Twenty-First International Conference on Machine Learning* (p. 62).

Kuhn, T. S. (1970). *The structure of scientific revolutions*. University of Chicago Press.

Lake, B., Salakhutdinov, R., Gross, J., & Tenenbaum, J. (2011). One shot learning of simple visual concepts. In *Proceedings of the Annual Meeting of the Cognitive Science Society* (Vol. 33).

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2019). BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, *36*, 1234–1240. https://doi.org/10.1093/bioinformatics/btz682

Li, X., Sun, Z., Xue, J.-H., & Ma, Z. (2020). A concise review of recent few-shot meta-learning methods. *arXiv preprint arXiv:2005.10953*.

Li, X.-L., Yu, P. S., Liu, B., & Ng, S.-K. (2009). Positive unlabeled learning for data stream classification. In *Proceedings of the 2009 SIAM International Conference on Data Mining* (pp. 259–270).

Li, Y.-X., Ji, S., Kumar, S., Ye, J., & Zhou, Z.-H. (2011). Drosophila gene expression pattern annotation through multi-instance multi-label learning. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *9*(1), 98–112.

Libbrecht, M. W., & Noble, W. S. (2015). Machine learning applications in genetics and genomics. *Nature Reviews Genetics*, *16*(6), 321–332.

Liu, P., Qiu, X., & Huang, X. (2016). Recurrent neural network for text classification with multi-task learning. *arXiv preprint arXiv: 1605.05101*.

Liu, W., Wang, J., & Chang, S.-F. (2012). Robust and scalable graph-based semisupervised learning. *Proceedings of the IEEE*, *100*(9), 2624–2638.

Manevitz, L. M., & Yousef, M. (2000). Document classification on neural networks using only positive examples. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 304–306).

Mignone, P., Pio, G., D'Elia, D., & Ceci, M. (2020). Exploiting transfer learning for the reconstruction of the human gene regulatory network. *Bioinformatics*, *36*(5), 1553–1561.

Mordelet, F., & Vert, J. P. (2014). A bagging SVM to learn from positive and unlabeled examples. *Pattern Recognition Letters*, *37*, 201–209.

Moya, M. (1993). One-class classifier networks for target recognition applications. In *Proceedings of the World Congress on Neural Networks*, (pp. 797–801).

Moya, M. M., & Hush, D. R. (1996). Network constraints and multi-objective optimization for one-class classification. *Neural Networks*, *9*(3), 463–474.

Niu, G., du Plessis, M. C., Sakai, T., Ma, Y., & Sugiyama, M. (2016). Theoretical comparisons of positive-unlabeled learning against positive-negative learning. In *Advances in Neural Information Processing Systems* (pp. 1199–1207).

Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1345–1359.

Patton, M., & Fund, R. E. C. M. (2002). *Qualitative research & evaluation methods*. SAGE Publications. Retrieved from. https://books.google.fi/books?id=FjBw2oi8El4C

Peng, L., Chen, L., Ye, Z., & Zhang, Y. (2018). Aroma: A deep multi-task learning based simple and complex human activity recognition method using wearable sensors. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, *2*(2), 1–16.

Perera, P., Oza, P., & Patel, V. M. (2021). One-class classification: A survey. *arXiv preprint arXiv:2101.03064* .

Pimentel, M. A., Clifton, D. A., Clifton, L., & Tarassenko, L. (2014). A review of novelty detection. *Signal Processing*, *99*, 215–249.

Pulkkinen, T., Roos, T., & Myllymäki, P. (2011). Semi-supervised learning for wlan positioning. In *International Conference on Artificial Neural Networks* (pp. 355–362).

Read, J., Pfahringer, B., Holmes, G., & Frank, E. (2011). Classifier chains for multi-label classification. *Machine Learning*, *85*(3), 333–359.

Rezaei, M., & Shahidi, M. (2020). Zero-shot learning and its applications from autonomous vehicles to covid-19 diagnosis: A review. *Intelligence-Based Medicine*, *3*, 100005.

Rodionova, O. Y., Oliveri, P., & Pomerantsev, A. L. (2016). Rigorous and compliant approaches to one-class classification. *Chemometrics and Intelligent Laboratory Systems*, *159*, 89–96.

Ruder, S. (2017). An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.

Ruff, L., Vandermeulen, R., Goernitz, N., Deecke, L., Siddiqui, S. A., Binder, A., Müller, E., & Kloft, M. (2018). Deep one-class classification. In *International Conference on Machine Learning* (pp. 4393–4402).

Samuel, A. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, *3*, 210–229.

Schapire, R. E., & Singer, Y. (2000). Boostexter: A boosting-based system for text categorization. *Machine Learning*, *39*(2), 135–168.

Schmidhuber, J. (1987). *Evolutionary principles in self-referential learning, or on learning how to learn: The meta-meta-... hook (Diploma thesis)*. Technische Universität München.

Schölkopf, B., Williamson, R. C., Smola, A. J., Shawe-Taylor, J., Platt, J. C., et al. (1999). Support vector method for novelty detection. *Advances in Neural Information Processing Systems*, *12*, 582–588.

Shi, M., & Zhang, B. (2011). Semi-supervised learning improves gene expression-based prediction of cancer recurrence. *Bioinformatics*, *27*(21), 3017–3023.

Smolander, J., Dehmer, M., & Emmert-Streib, F. (2019). Comparing deep belief networks with support vector machines for classifying gene expression data from complex disorders. *FEBS Open Bio*, *9*(7), 1232–1248.

Sosnin, S., Vashurina, M., Withnall, M., Karpov, P., Fedorov, M., & Tetko, I. V. (2019). A survey of multi-task learning methods in chemoinformatics. *Molecular Informatics*, *38*(4), 1800108.

Sperry, R. W. (1952). Neurology and the mind-brain problem. *American Scientist*, *40*(2), 291–312.

Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H., & Hospedales, T. M. (2018). Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1199–1208).

Sutton, R., & Barto, A. (1998). *Reinforcement learning*. MIT Press.

Tarassenko, L., Hayton, P., Cerneaz, N., & Brady, M. (1995). Novelty detection for the identification of masses in mammograms. In *Proceeding of the 4th International Conference on Artificial Neural Networks*, Vol. 1995, pp. 442–447.

Tax, D. M. J. (2001). *One-class classification: Concept learning in the absence of counter-examples (Doctoral dissertation)*. Technische Universiteit Delft. Retrieved from. http://proquest.umi.com/pqdweb?did=728104171&Fmt=2&clientId=36097&RQT=309&VName=PQD

Thrun, S., & Pratt, L. (1998). *Learning to learn*. Springer Science & Business Media.

Thung, K.-H., & Wee, C.-Y. (2018). A brief review on multi-task learning. *Multimedia Tools and Applications*, *77*(22), 29705–29725.

Tsoumakas, G., & Katakis, I. (2007). Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, *3*(3), 1–13.

Tsoumakas, G., Katakis, I., & Vlahavas, I. (2009). Mining multi-label data. In *Data mining and knowledge discovery handbook* (pp. 667–685). Springer.

Tsoumakas, G., Katakis, I., & Vlahavas, I. (2010). Random k-labelsets for multilabel classification. *IEEE Transactions on Knowledge and Data Engineering*, *23*(7), 1079–1089.

Vamathevan, J., Clark, D., Czodrowski, P., Dunham, I., Ferran, E., Lee, G., Li, B., Madabhushi, A., Shah, P., Spitzer, M., & Zhao, S. (2019). Applications of machine learning in drug discovery and development. *Nature Reviews Drug Discovery*, *18*(6), 463–477.

Van Engelen, J. E., & Hoos, H. H. (2020). A survey on semi-supervised learning. *Machine Learning*, *109*(2), 373–440.

Van Otterlo, M., & Wiering, M. (2012). Reinforcement learning and Markov decision processes. In *Reinforcement learning* (pp. 3–42). Springer.

Vanschoren, J. (2019). Meta-learning. In *Automated machine learning* (pp. 35–61). Springer.

Vapnik, V. N. (1995). *The nature of statistical learning theory*. Springer.

Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., & Wierstra, D. (2016). Matching networks for one shot learning. *arXiv preprint arXiv:1606.04080*.

Wang, J., Ma, Y., Zhang, L., Gao, R. X., & Wu, D. (2018). Deep learning for smart manufacturing: Methods and applications. *Journal of Manufacturing Systems*, *48*, 144–156.

Wang, Y., Yao, Q., Kwok, J. T., & Ni, L. M. (2020). Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys (CSUR)*, *53*(3), 1–34.

Watkins, C. J., & Dayan, P. (1992). Q-learning. *Machine learning*, *8*(3–4), 279–292.

Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). A survey of transfer learning. *Journal of Big data*, *3*(1), 9.

Wiering, M. A., van Hasselt, H., Pietersma, A.-D., & Schomaker, L. (2011). Reinforcement learning algorithms for solving classification problems. In *2011 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)* (pp. 91–96).

Wu, B., Lyu, S., Hu, B.-G., & Ji, Q. (2015). Multi-label learning with missing labels for image annotation and facial action unit recognition. *Pattern Recognition*, *48*(7), 2279–2289.

Wu, B., Qiu, W., Jia, J., & Liu, N. (2020). Landslide susceptibility modeling using bagging-based positive-unlabeled learning. *IEEE Geoscience and Remote Sensing Letters*, *18*, 766–770.

Wuest, T., Weimer, D., Irgens, C., & Thoben, K.-D. (2016). Machine learning in manufacturing: Advantages, challenges, and applications. *Production & Manufacturing Research*, *4*(1), 23–45.

Xia, Z., Wu, L.-Y., Zhou, X., & Wong, S. T. (2010). Semi-supervised drug-protein interaction prediction from heterogeneous biological spaces. *BMC Systems Biology*, *4*, 1–16.

Yan, L., Zheng, Y., & Cao, J. (2018). Few-shot learning for short text classification. *Multimedia Tools and Applications*, *77*(22), 29799–29810.

Yang, P., Li, X., Chua, H.-N., Kwoh, C.-K., & Ng, S.-K. (2014). Ensemble positive unlabeled learning for disease gene identification. *PLoS One*, *9*(5), e97079.

Yu, S., & Li, C. (2007). Pe-puc: A graph based pu-learning approach for text classification. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition* (pp. 574–584).

Zhang, B., & Zuo, W. (2008). Learning from positive and unlabeled examples: A survey. In *Proceedings of the 2008 International Symposiums on Information Processing* (pp. 650–654).

Zhang, M.-L., & Zhou, Z.-H. (2007). Ml-knn: A lazy learning approach to multi-label learning. *Pattern Recognition*, *40*(7), 2038–2048.

Zhang, M.-L., & Zhou, Z.-H. (2013). A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, *26*(8), 1819–1837.

Zhang, Y., & Yang, Q. (2018). An overview of multi-task learning. *National Science Review*, *5*(1), 30–43.

Zhu, X., & Goldberg, A. B. (2009). Introduction to semi-supervised learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, *3*(1), 1–130.

Zhu, Y., Chen, Y., Lu, Z., Pan, S., Xue, G.-R., Yu, Y., & Yang, Q. (2011). Heterogeneous transfer learning for image classification. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 25).

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., & He, Q. (2020). A comprehensive survey on transfer learning. *Proceedings of the IEEE*, *109*(1), 43–76.

How to cite this article: Emmert-Streib, F., & Dehmer, M. (2022). Taxonomy of machine learning paradigms: A data-centric perspective. *WIREs Data Mining and Knowledge Discovery*, e1470. https://doi.org/10.1002/widm.1470