IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received October 31, 2021, accepted November 26, 2021, date of publication December 14, 2021, date of current version December 27, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3135514

A Survey on Cost Types, Interaction Schemes, and Annotator Performance Models in Selection Algorithms for Active Learning in Classification

MAREK HERDE, DENIS HUSELJIC, BERNHARD SICK, (Member, IEEE), AND ADRIAN CALMA

Department of Intelligent Embedded Systems, University of Kassel, 34121 Kassel, Germany

Corresponding author: Marek Herde (marek.herde@uni-kassel.de)

This work was supported by the CIL Project through the University of Kassel under Grant P/710 and Grant P/1082.

ABSTRACT Pool-based active learning (AL) aims to optimize the annotation process (i.e., labeling) as the acquisition of annotations is often time-consuming and therefore expensive. For this purpose, an AL strategy queries annotations intelligently from annotators to train a high-performance classification model at a low annotation cost. Traditional AL strategies operate in an idealized framework. They assume a single, omniscient annotator who never gets tired and charges uniformly regardless of query difficulty. However, in real-world applications, we often face human annotators, e.g., crowd or in-house workers, who make annotation mistakes and can be reluctant to respond if tired or faced with complex queries. Recently, many novel AL strategies have been proposed to address these issues. They differ in at least one of the following three central aspects from traditional AL: 1) modeling of (multiple) human annotators whose performances can be affected by various factors, such as missing expertise; 2) generalization of the interaction with human annotators through different query and annotation types, such as asking an annotator for feedback on an inferred classification rule; 3) consideration of complex cost schemes regarding annotations and misclassifications. This survey provides an overview of these AL strategies and refers to them as real-world AL. Therefore, we introduce a general real-world AL strategy as part of a learning cycle and use its elements, e.g., the query and annotator selection algorithm, to categorize about 60 real-world AL strategies. Finally, we outline possible directions for future research in the field of AL.

INDEX TERMS Active learning, classification, error-prone annotators, human-in-the-loop learning, interactive learning, machine learning.

I. INTRODUCTION

Information and communication technology has become an integral part of humans' lives and supports us embedded in our surroundings [1]. In particular, improving computational power and the ease of collecting a plethora of data has promoted *machine learning* (ML) [2], [3]. Nowadays, ML models are employed in various fields [4], ranging from recommender systems [5], text classification [6], and speech recognition [7] to object detection in videos [8]. In this survey, we consider ML for building classification models. They learn from data sets consisting of instances and their corresponding annotations (e.g., class labels, class membership probabilities, etc.). However, annotating instances may be costly and time-consuming since it is often manually executed by annotators.

In general, an annotator is an information or knowledge source, such as a human, the Internet, or a simulation system [9], and can annotate various types of queries. In this survey, we focus on human annotators. Other commonly used terms are oracle [10], expert [11], worker [12], teacher [13], and labeler [14]. A large group of (human) annotators who do not necessarily know each other is also named a crowd [15]. The exact characteristics of a crowd, e.g., the number and heterogeneity of the annotators, depend on the requirements of the crowdsourcing initiative at hand.

Active learning (AL) is a subfield of human-in-the-loop learning [16] and interactive ML [17], [18], which directly

The associate editor coordinating the review of this manuscript and approving it for publication was Eunil Park.

and iteratively interacts with human annotators. It aims at reducing annotation and misclassification cost [19]. Thus, an AL strategy queries annotations for instances from which the classification model is expected to learn the most [20]. As a result, the classification model trained on an actively selected subset of annotated instances reaches in average a superior performance to a model trained on a randomly selected subset. AL strategies have been successfully employed in several applications, e.g., malware detection [21], waste classification [22], classification of medical images [23], and training of robots [24]. However, many of these AL strategies make three central assumptions that limit their practical use [25], and we refer to them as traditional AL:

(1) There is a single omnipresent and omniscient annotator providing the correct annotation for each query at any time. This assumption conflicts with the available options of annotation acquisitions. In particular, crowdsourcing represents a popular way to obtain data annotations [26]. However, on crowdsourcing platforms, e.g., Amazon's Mechanical Turk [27], [28], CloudResearch (formerly TurkPrime) [29], and Prolific Academic [30], multiple error-prone annotators have to be considered [31]. Otherwise, annotation mistakes (e.g., noisy class labels) will degrade the classification model's performance [32], [33].
(2) The cost of an annotation is constant across the queries. This assumption is violated in cases such as biomedical citation screening [11], in which articles are to be classified as relevant or irrelevant for a particular research topic. The time to annotate an article depends on its length, complexity, and the queried annotator [34]. Hence, the cost varies across pairs of articles and annotators.
(3) Each query requests the class label of a specific instance. This assumption ignores the possibility of designing more general and effective queries [25], [35]. Some of these queries also require annotations to be more complex than simple class labels. We avoid confusion regarding the terms label and annotation by defining an annotation as the most general reply to a query, e.g., an annotator could answer a query with "I have no idea!". Correspondingly, a class label is a specific example of an annotation.

Various concepts have been proposed to overcome the limitations above. These include collaborative interactive learning [36], [37] and proactive learning [38], [39]. We summarize their main differences to traditional AL through the three following aspects:

(1) Instead of assuming a single omniscient and omnipresent annotator, they consider (multiple) human, error-prone annotators whose performances can be affected by various factors, e.g., missing expertise, fatigue, and malicious behavior.
(2) Instead of repeatedly querying class labels of instances, they generalize the interaction with human annotators by considering different types of queries and annotations, such as asking an annotator for feedback on an inferred classification rule.
(3) Instead of assuming uniform cost, they take more complex cost schemes regarding annotations and misclassifications into account.

In this survey, we provide an overview of existing AL strategies taking at least one of the three aspects into account and refer to them as real-world AL. We limit the scope by including only strategies for classification in the pool-based AL setting [19] because it is the most researched AL field. However, many implications of this survey go beyond this scope and are emphasized in the outlook. Based on these prerequisites, this survey makes the following contributions:

- We formalize the objective of a real-world AL strategy as the optimal annotation sequence to a cost-sensitive classification problem.
- We propose a taxonomy of existing cost types, interaction schemes, annotator performance models, and selection algorithms to compare different real-world AL strategies.
- We give a comprehensive comparison of about 60 real-world AL strategies and analyze them regarding their handling of error-prone annotators, usage of query and annotation types, consideration of imbalanced misclassification and annotation cost, and query-annotator selection.
- We identify five unsolved challenges in the real-world AL setting and formulate them as future research directions.

We structure this survey's main body according to Fig. 1 that gives an overview of the main topics reviewed in this survey. The four sections III–VI are accompanied by a respective tabular literature overview of real-world AL strategies, including detailed analyses in this survey's appendices as supplementary material. Based on these literature overviews, we formulate challenges in the setting of real-world AL and beyond in Section VII. We conclude this survey in Section VIII.

II. REAL-WORLD ACTIVE LEARNING

In this section, we introduce the problem setting of real-world AL. A real-world application illustrates a possible scenario violating the assumptions of traditional AL and thus indicating the need for real-world AL strategies. In the context of this application, we also explain the notation used throughout this survey. Moreover, we formalize the objective of real-world AL as the optimal solution to a cost-sensitive classification problem and present learning cycles finding greedy approximations of this solution.

A. MOTIVATING APPLICATION

An example of a practical use case requiring the employment of a real-world AL strategy is the classification of low-voltage grids described in [40], [41]. They connect most consumers, e.g., households, to the electrical power system, and an illustration of such a grid is given in Fig. 2.

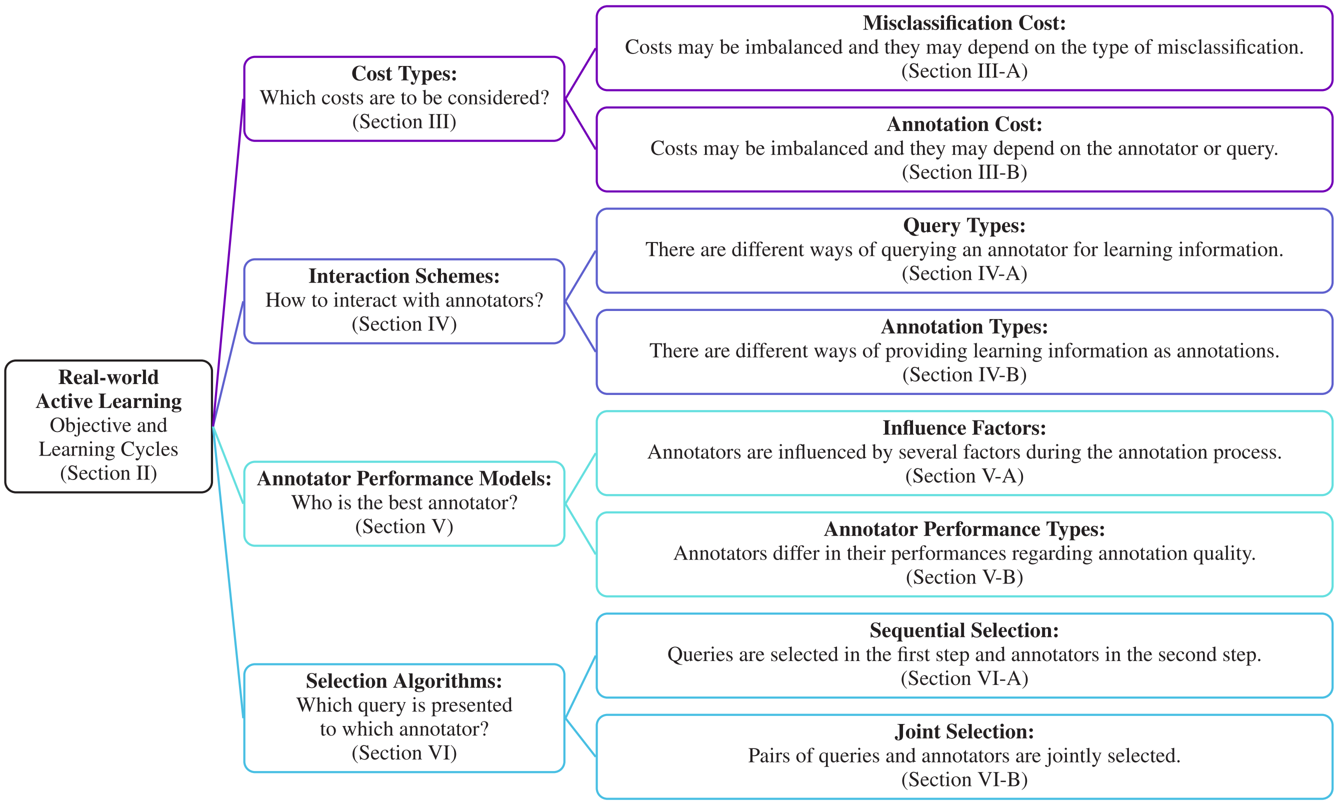


FIGURE 1. Overview of real-world AL and structure of this survey's main body: The nodes of the tree name the different topics of real-world AL identified in this survey. Additionally, they provide a brief summary of each topic and reference the specific section with more details.

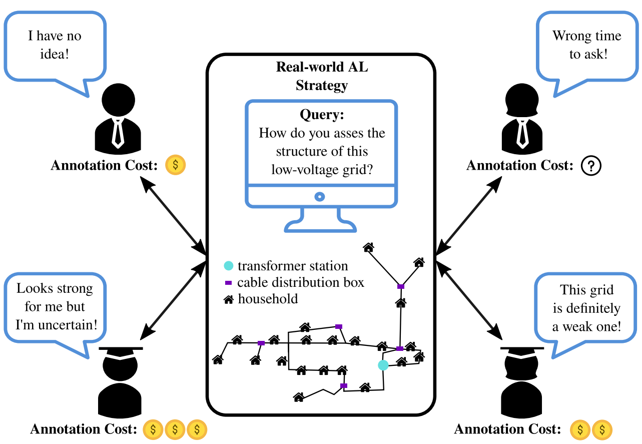


FIGURE 2. Example of a real-world AL use-case with four annotators.

Formerly, the power system was designed to transport energy from a few central generators (plants) to the consumers. However, recent developments are characterized by an increasing number of installed distributed generators, particularly photovoltaic generators, in low-voltage grids [42]. These distributed generators may provoke, e.g., an overload of electrical components, and violate critical voltage values within a grid. Assessing the hosting capacity of low-voltage grids for distributed generation supports the responsible distribution system operator in deciding for which low-voltage grid an investment in the infrastructure could be most beneficial such that its sustainable operational reliability is guaranteed [40].

In this context, a significant challenge is the high complexity of low-voltage grids. Therefore, multiple annotators with heterogeneous background knowledge are requested to provide annotations, such as strong, weak, etc., classifying the hosting capacities of low-voltage grids (cf. Fig. 2). To do so, the annotators have access to a grid diagram and corresponding tabular information. The provided annotations, i.e., ordered classes and confidence assessments in this case, are prone to error because of missing expertise, for example. Moreover, the annotations are expensive because the annotators have to investigate the grids to generate an annotation regarding the hosting capacity. A real-world AL strategy can save time and money by training a classification model to categorize each possible low-voltage grid's hosting capacity automatically.

As a result, a representative question of this survey would be: *"How to design a real-world AL strategy for solving problems such as the classification of low-voltage grids?"*.

B. FORMALIZATION OF PROBLEM SETTING

An instance is described by a D-dimensional feature vector $\mathbf{x} = (x_1, \ldots, x_D)^\mathrm{T}, D \in \mathbb{N}$. It is drawn from the distribution $\Pr(X) = \Pr(X_1, \ldots, X_D)$ defined over a D-dimensional feature (input) space Ω_X, where X_d denotes the random variable of the feature $d \in \{1, \dots, D\}$ and X is used as short-cut for the D dimensional random variable representing all features. The observed multi-set of identically and independently distributed instances is given by $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \subseteq \Omega_X$. For example, the instance of the low-voltage grid illustrated in Fig. 2 may take the form

$$\mathbf{x}_n = \begin{pmatrix} x_{n1} \\ x_{n2} \\ \vdots \\ x_{nD} \end{pmatrix} = \begin{pmatrix} \texttt{\# transformer stations} \\ \texttt{\# cable distribution boxes} \\ \vdots \\ \texttt{\# households} \end{pmatrix}. \tag{1}$$

Each instance $\mathbf{x}_n$ belongs to a true class $y_n \in \Omega_Y$ sampled from the categorical distribution $\Pr(Y \mid X = \mathbf{x}_n)$ with Y denoting the random variable of the true class labels. In total, there are $|\Omega_Y| = C \in \mathbb{N}_{\geq 2}$ classes. The multi-set of true class labels for the observed instances in $\mathcal{X}$ is denoted as $\mathcal{Y} = \{y_1, \dots, y_N\}$. Regarding the classification of low-voltage grids, the class labels would have an ordinal structure ranging from a `very weak` ($Y = 1 \in \Omega_Y$) to a `very strong` ($Y = 5 \in \Omega_Y$) hosting capacity.

As pointed out, there is no omniscient and omnipresent annotator in most applications. In the context of real-world AL, we work with (multiple) error-prone annotators who we summarize in the set $\mathcal{A} = \{a_1, \dots, a_M\}$. Each annotator can be queried to provide annotations. An annotation is not restricted to be a specific class label $y \in \Omega_Y$, but all kinds of annotations are allowed, e.g., confidence scores [43], probabilistic labels [44], or rejecting to answer a query [45]. The space of possible annotations is summarized by the set Ω_Z, e.g., $\Omega_Z = [0, 1]$ if probabilistic class labels are expected for a binary classification problem. The (multivariate) random variable for the annotations of annotator a_m is denoted as Z_m.

A query cannot only ask for the class label of a specific instance $\mathbf{x}_n$, but more general queries such as "Do instance $\mathbf{x}_n$ and instance $\mathbf{x}_m$ belong to the same class?" can be formulated [46]. To learn from queries and annotations, a classification model requires appropriate mathematical representations of them. An exemplary representation of the query given above would be $q = \{\mathbf{x}_n, \mathbf{x}_m\}$. The mathematical representations of all possible queries are summarized in a set $\mathcal{Q}_\mathcal{X}$, which depends on the underlying classification problem and the set of observed instances $\mathcal{X}$ [47]. Due to this dependency, we can interpret the queries as random events, and Q denotes the associated random variable. In most cases, a query asks for the class label of a specific instance such that we can define $\mathcal{Q}_\mathcal{X} = \mathcal{X}$ as query set.

The task of a real-world AL strategy is to generate a sequence scheduling the execution of the annotation process, which we assume to consist of countable distinct (time) steps. In other words, a sequence answers the question: "Which annotator has to answer which query at which time step?". Accordingly, we define a sequence as a function $\mathcal{S} : \mathbb{N} \to \mathcal{P}(\mathcal{Q}_\mathcal{X} \times \mathcal{A})$, such that $(q_l, a_m) \in \mathcal{S}(t)$ induces an annotation of query q_l by annotator a_m at time step $t \in \mathbb{N}$. The annotation behavior of an annotator can be modeled through a conditional distribution $\Pr(Z_m \mid Q = q_l, t)$ from which $z_{lm}^{(t)} \in \Omega_Z$ is drawn as annotation of annotator a_m for query q_l, i.e., $z_{lm}^{(t)} \sim \Pr(Z_m \mid Q = q_l, t)$. As a result, annotators are not compulsorily deterministic in their decisions. Still, decisions might also change throughout the annotation process, e.g., if an annotator gets tired during the annotation process [48]. An annotation process executed until the beginning of the time step t according to a sequence $\mathcal{S}$ leads to a data set

$$\mathcal{D}(t) = \Big\{(q_l, a_m, z_{lm}^{(t')}) \;\Big|\; t' \in \mathbb{N} \wedge t' < t \wedge (q_l, a_m) \in \mathcal{S}(t') \wedge z_{lm}^{(t')} \sim \Pr(Z_m \mid Q = q_l, t')\Big\} \tag{2}$$

consisting of triplets of a query, an annotator, and an annotation. We define the end of a sequence $\mathcal{S}$ as the last time step at which an annotation has been performed, i.e., where the selection is not empty:

$$t_\mathcal{S} = \max(\{t \mid \mathcal{S}(t) \neq \emptyset \wedge t \in \mathbb{N}\}). \tag{3}$$

On a data set $\mathcal{D}(t)$, a classification model described by its parameters $\boldsymbol{\theta}$ can be trained. We denote the resulting parameters of the classification model by $\boldsymbol{\theta}_{\mathcal{D}(t)}$. For example, these parameters would correspond to weights in the case of a neural network [49] taken as a classification model. The trained classification model predicts class labels for given instances, where the prediction for an instance $\mathbf{x} \in \Omega_X$ is denoted by $\hat{y}(\mathbf{x} \mid \boldsymbol{\theta}_{\mathcal{D}(t)}) \in \Omega_Y$. In many cases, the classification model can predict the class label of an instance and estimate the probabilities of class memberships. In this case, we denote the estimated class membership probability that a given instance $\mathbf{x}$ belongs to class y by $\Pr(Y = y \mid X = \mathbf{x}, \boldsymbol{\theta}_{\mathcal{D}(t)})$.

C. OBJECTIVE

Given the formalized problem setting and generalizing the objective definitions in [38], [50] toward all query and annotation types including complex cost schemes, we formulate the objective of real-world AL as determining the optimal annotation sequence for a cost-sensitive classification problem:

$$\mathcal{S}^* = \underset{\mathcal{S} \in \Omega_S}{\arg\min} \left[\mathrm{MC}(\boldsymbol{\theta}_{\mathcal{D}(t_\mathcal{S}+1)} \mid \boldsymbol{\kappa}) + \mathrm{AC}(\mathcal{D}(t_\mathcal{S} + 1) \mid \boldsymbol{\nu})\right] \quad \text{subject to the constraints } \mathcal{C}, \tag{4}$$

where Ω_S denotes the set of all potential sequences. MC and AC are the misclassification and annotation cost, respectively. We expect them to be on the same scale. Otherwise, extra normalization might be necessary. The optimal annotation sequence $\mathcal{S}^*$ minimizes the total cost while satisfying all constraints $\mathcal{C}$. A common constraint is a maximum annotation budget $B \in \mathbb{R}_{>0}$, i.e., $\mathcal{C} = \{\mathrm{AC}(\mathcal{D}(t_\mathcal{S} + 1) \mid \boldsymbol{\nu}) \leq B\}$. The total cost is decomposed into MC and AC, where the vector $\boldsymbol{\kappa}$ encodes given hyperparameters for computing the MC, e.g, a cost matrix, and the vector $\boldsymbol{\nu}$ represents the hyperparameters for computing AC, e.g., wages of the annotators. We provide a more detailed discussion on different cost schemes in the setting of real-world AL in Section III.

Since it is difficult to find the optimal annotation sequence $\mathcal{S}^*$ given by Eq. 4 in advance [38], an AL strategy aims to approximate the optimal sequence through a greedy approach. Therefore, the annotation sequence $\mathcal{S}$ is defined iteratively at run time by executing a cycle where one iteration corresponds to a single time step. We start with the description of such a cycle for traditional AL. Subsequently, we restructure it to fit the setting of real-world AL.

D. TRADITIONAL ACTIVE LEARNING CYCLE

In traditional AL, an omniscient and omnipresent annotator $\mathcal{A} = \{a_1\}$ is assumed to be available [19]. Moreover, a query expects the class label of an instance such that the set of queries can be represented by $\mathcal{Q}_\mathcal{X} = \mathcal{X}$ and the set of annotations is given by the set of classes, i.e., $\Omega_Z = \Omega_Y$. Traditional AL strategies differ between the labeled (annotated) set $\mathcal{L}(t) = \{(\mathbf{x}_n, y_n) \mid (\mathbf{x}_n, a_1, y_n) \in \mathcal{D}(t)\}$ and the unlabeled (non-annotated) set $\mathcal{U}(t) = \{\mathbf{x}_n \mid \mathbf{x}_n \in \mathcal{X} \wedge (\mathbf{x}_n, y_n) \notin \mathcal{L}(t)\}$ obtained after executing the $(t-1)$-th iteration cycle. The main idea is to develop a strategy intelligently selecting instances from the unlabeled pool $\mathcal{U}(t)$ to which the annotator a_1 assigns true class labels. Due to the omniscience of this annotator a_1, the annotation distribution satisfies $\Pr(Z_1 = y \mid X = \mathbf{x}_n, Y = y, t) = 1$ for all iteration cycles $t \in \mathbb{N}$, classes $y \in \Omega_Y$, and observed instances $\mathbf{x}_n \in \mathcal{X}$.

Fig. 3 summarizes the entire selection procedure as a cycle. (1) At the start of the iteration cycle t, the traditional AL strategy selects an unlabeled instance $\mathbf{x}_{n^*}$ from the unlabeled pool: $\mathcal{U}(t+1) = \mathcal{U}(t) \setminus \{\mathbf{x}_{n^*}\}$. (2) Subsequently, the instance is presented to the omniscient annotator $\mathcal{A} = \{a_1\}$ who provides its true class label y_{n^*}. The resulting instance-label pair is inserted into the labeled pool: $\mathcal{L}(t+1) = \mathcal{L}(t) \cup \{(\mathbf{x}_{n^*}, y_{n^*})\}$, (3) on which the classification model is retrained by updating its parameters $\boldsymbol{\theta}_{\mathcal{L}(t)} \rightarrow \boldsymbol{\theta}_{\mathcal{L}(t+1)}$. (4) At the end of the cycle, the traditional AL strategy decides whether to continue or to stop learning. This decision is made by a so-called stopping criterion [51]–[53], which is part of ongoing research and not within this survey's scope.

The selection of an instance is based on a so-called utility measure $\phi : \mathcal{X} \rightarrow \mathbb{R}$ [54] estimating the utilities of the observed instances $\mathcal{X}$ regarding the classification model to be trained. In general, the unlabeled instance with the maximum utility is selected in iteration cycle t:

$$\mathbf{x}_{n^*} = \underset{\mathbf{x}_n \in \mathcal{U}(t)}{\arg\max} \left[\phi(\mathbf{x}_n \mid \boldsymbol{\theta}_{\mathcal{L}(t)})\right]. \qquad (5)$$

There are many approaches computing instances' utilities. In the following, we briefly describe two fundamental concepts:

- The simplest concept of utility measures is *uncertainty sampling* (US) [55], which usually requires an instance's class membership probabilities estimated by the classification model to be trained. Alternatively, distances to decision boundaries [56] are used as proxies of them. US ranks all instances in the unlabeled pool $\mathcal{U}(t)$ based on an uncertainty measure and queries the label for the instance with the maximum uncertainty regarding its class information. A common uncertainty measure is the entropy H [57] of the class distribution such that an instance's utility is computed as

$$\phi_{\mathrm{US}}(\mathbf{x}_n \mid \boldsymbol{\theta}_{\mathcal{L}(t)}) = H[\Pr(Y \mid X = \mathbf{x}_n, \boldsymbol{\theta}_{\mathcal{L}(t)})]. \qquad (6)$$

- The decision-theoretic framework *expected error reduction* (EER) [58] estimates the performance of the classification model. Therefor, EER assumes that the instances in the unlabeled pool $\mathcal{U}(t)$ form a validation set. For each unlabeled instance, the classification model's expected error is computed on this validation set by retraining the classification model with each combination of the given unlabeled instance and its possible class label. The multiple retraining procedures of the classification model lead to high computational complexity. The resulting estimate of the negative expected error defines the utility measure. Correspondingly, EER selects the instance leading to the minimum estimated error.

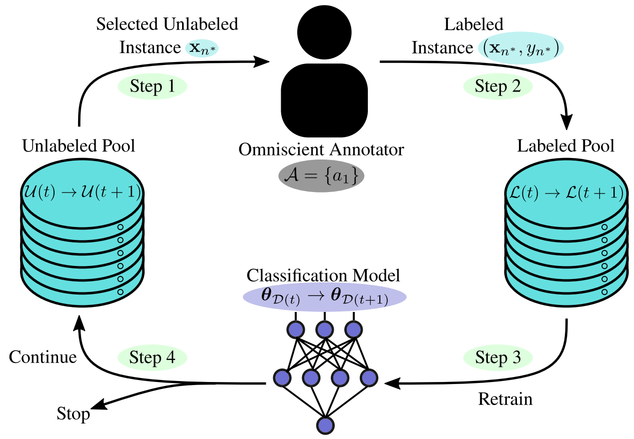


FIGURE 3. Traditional AL cycle according to Settles [19].

One of the main challenges regarding the design of utility measures is the exploration-exploitation trade-off. On the one hand, we aim to select instances near the classification model's decision boundary to refine it (exploitation). On the other hand, we aim to select instances in unknown regions (exploration) [59]. More advanced AL strategies balance this trade-off by considering distances to decision boundaries, densities, class distribution estimates [60]–[62], or using a Bayesian approach [63].

In batch mode AL [23], we must consider the diversity of instances since multiple instances (batch of instances) are selected in each learning iteration cycle. However, a detailed analysis of instance utility measures in the traditional AL setting is beyond the scope of this survey, and a more detailed discussion on them is given in [19], [20], [54], [64].

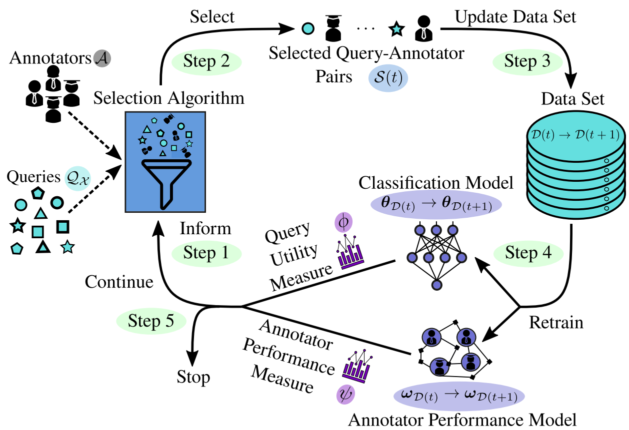


FIGURE 4. Proposed real-world AL cycle.

E. REAL-WORLD ACTIVE LEARNING CYCLE

The traditional AL cycle depicted in Fig. 3 has to be adjusted to fit the setting of real-world AL. Our resulting cycle, including a real-world AL strategy's elements, i.e., query utility measure, annotator performance measure, and selection algorithm, is shown in Fig. 4. (1) At the start of the iteration cycle t, a classification and annotator model, both trained on the current data set $\mathcal{D}(t)$, provide information regarding a query utility and an annotator performance measure. (2) Based on both measures, a selection algorithm specifies a set of query-annotator pairs $\mathcal{S}(t) \subseteq \mathcal{Q}_\mathcal{X} \times \mathcal{A}$. Each pair $(q_l, a_m) \in \mathcal{S}(t)$ initiates an annotation of query q_l by annotator a_m. (3) The annotations are inserted into the data set: $\mathcal{D}(t+1) = \mathcal{D}(t) \cup \{(q_l, a_m, z_{lm}^{(t)}) \mid (q_l, a_m) \in \mathcal{S}(t) \wedge z_{lm}^{(t)} \sim \Pr(Z_m \mid Q = q_l, t)\}$. (4) Then, the classification and annotator model are retrained on the updated data set ($\boldsymbol{\theta}_{\mathcal{D}(t)} \to \boldsymbol{\theta}_{\mathcal{D}(t+1)}$ and $\boldsymbol{\omega}_{\mathcal{D}(t)} \to \boldsymbol{\omega}_{\mathcal{D}(t+1)}$). (5) At the end of the iteration cycle, the real-world AL strategy decides whether to stop or to continue learning.

A **query utility measure** $\phi : \mathcal{Q}_\mathcal{X} \to \mathcal{R}_\phi$ is an element being already part of the traditional AL setting. However, in the real-world AL setting, not only the class labels of non-annotated (unlabeled) instances can be queried, but more general queries can be selected for annotation. This also includes a re-annotation of instances, known as repeated labeling [65], re-labeling [66], or backward instance labeling [67]. Hence, the strict distinction into a non-annotated (unlabeled) set $\mathcal{U}(t)$ and an annotated (labeled) set $\mathcal{L}(t)$ is often not adequate anymore. As a result, the utility measure ϕ needs to be adapted to quantify the utility of more general queries. Another adaption concerns the form of the output of the utility measure. Instead of computing a single score per query, i.e., $\mathcal{R}_\phi \subseteq \mathbb{R}$, a utility measure may provide a more general description for each query, e.g., a distribution, which can then be combined with annotator performance estimates [68], [69]. We provide an overview of real-world AL strategies with query utility measures for specific query and annotation types in Section IV.

A **annotator performance measure** $\psi : \mathcal{Q}_\mathcal{X} \times \mathcal{A} \to \mathcal{R}_\psi$ represents a novel element compared to traditional AL strategies and is defined through an annotator model. Similar to a classification model, an annotator model has parameters $\boldsymbol{\omega}_{\mathcal{D}(t)}$ learned from a data set $\mathcal{D}(t)$. Its main task concerns the estimation of the performance $\psi(q_l, a_m \mid \boldsymbol{\omega}_{\mathcal{D}(t)}) \in \mathcal{R}_\psi$ of an annotator a_m regarding a query q_l [68], e.g., the probability for providing a correct annotation. In most cases, $\psi(q_l, a_m \mid \boldsymbol{\omega}_{\mathcal{D}(t)})$ is a point estimate, i.e., $\mathcal{R}_\psi \subseteq \mathbb{R}$, but there are also annotator models estimating probability distributions over annotator performances [68], [69]. Moreover, an annotator model may account for improvements and deteriorations of annotators' performances, e.g., when an annotator learns or gets exhausted. The annotator performance may also be affected by collaboration mechanisms between the annotators, e.g., the best annotator is asked to teach the worst annotator [70]. We provide an overview of annotator performance measures in Section V.

A real-world AL strategy is completed by a **selection algorithm** as the final element. It updates the annotation sequence $\mathcal{S}$ by selecting query-annotator pairs in each iteration cycle t. This selection is specified by choosing a subset of query-annotator pairs $\mathcal{S}(t) \subseteq \mathcal{Q}_\mathcal{X} \times \mathcal{A}$. Therefor, it assesses potential query-annotator pairs through the query utility and annotator performance measure. If the set $\mathcal{S}(t)$ contains multiple queries, we face similar challenges as in batch mode AL, e.g., selecting diverse queries. We provide an overview of selection algorithms in Section VI.

AC and MC are modeled in AL literature by designing cost-sensitive variants of query utility measures, annotator performance measures, or selection algorithms. We provide an overview in Section III.

III. COST TYPES

MC and AC are the most crucial cost types in the real-world AL setting, and we will summarize typical schemes of them in this section. There exist several additional types of cost when solving a classification problem, e.g., cost of computation (e.g., renting a graphics processing unit) and cost of test (e.g., getting the results of a blood test). They are described as a taxonomy in [71]. At the end of this section, we present a literature overview of real-world AL strategies explicitly modeling MC and/or AC.

A. MISCLASSIFICATION COST

Mistakes of the classification model induce MC (the first summand in Eq. 4). In the literature, we identified three cost schemes and describe them in increasing order of complexity in the following:

Uniform MC: Each classification error is charged at an equal cost. The classification model's performance is inversely proportional to the misclassification rate [72], i.e., the proportion of misclassified instances. This cost scheme is the simplest one and is assumed by traditional AL strategies.

Class-dependent MC: This cost scheme is probably the most common one in cost-sensitive classification [73], [74]. The cost of a classification error is defined by means of a cost matrix/table $\mathbf{C} \in \mathbb{R}_{\geq 0}^{C \times C}$, where an entry $\mathbf{C}[y, y']$ in row y and column y' denotes the cost of predicting the class label $\hat{y}(\mathbf{x}_n \mid \boldsymbol{\theta}_{\mathcal{D}(t)}) = y'$, when the instance $\mathbf{x}_n$ actually belongs to class $y_n = y$. Our grid classification example could use the mean absolute error on class numbers as a typical cost measure for ordinal classes [75]. It would be implemented through $\mathbf{C}[y, y'] = |y - y'|$. In some applications, the cost matrix is extended by adding an extra column representing cases where the classification model is too uncertain and rejects predicting a class label (known as reject option [76]).

Instance-dependent MC: Costs of classification errors depend on specific characteristics of instances. An example is fraud detection, where the amount of money involved in a particular case has an essential impact on MC [77]. For our grid classification example, it would be more expensive if many households were affected by overloading a low-voltage grid. Consequently, the feature `# households` in Eq. 1 is to play a central role when computing the cost of misclassifying a low-voltage grid.

MC can be computed as the expectation regarding the true (but unknown) joint distribution $\Pr(X, Y)$ of instances and class labels [76]. For example, class-dependent MC with a cost matrix as hyperparameter, i.e., $\boldsymbol{\kappa} = \mathbf{C}$, is computed according to

$$\mathrm{MC}(\boldsymbol{\theta}_{\mathcal{D}(t)} \mid \mathbf{C}) = \underset{\Pr(X=\mathbf{x},\, Y=y)}{\mathbb{E}} \left[\mathbf{C}[y, \hat{y}(\mathbf{x} \mid \boldsymbol{\theta}_{\mathcal{D}(t)})]\right]. \tag{7}$$

In practice, the exact computation of MC is often infeasible due to the limited size of test data. Furthermore, in the real-world AL setting, its estimation based on a separate set of instances is challenging because of a sampling bias (arising from the active data acquisition) [78] and the lack of known ground truth class labels (arising from the error-proneness of the annotators). Nevertheless, some real-world AL strategies take imbalanced, i.e., class- or instance-dependent, MC into account.

B. ANNOTATION COST

AC (the second summand in Eq. 4) arises from the work effort of the annotators who have to invest time to decide on appropriate annotations for the posed queries. The exact specification of AC depends on the underlying cost scheme. In the literature, we identified four different schemes and describe them in increasing order of complexity in the following:

Uniform AC: The cost of obtaining an annotation is constant for each query and independent of the queried annotator. Correspondingly, the AC is proportional to the number of acquired annotations. This cost scheme is the simplest one and is frequently used. In particular, it is often employed in crowdsourcing environments, where the requester sets a constant pay rate per query. This means the qualification of an annotator and the time spent on annotating a query have no impact on the AC.

Annotator-dependent AC: In this cost scheme, the AC explicitly depends on the queried annotator. This setting is typical when annotators with different qualifications receive different earnings per query, e.g., annotators with different levels of expertise. Of course, there is typically no guarantee that expensive annotators provide more accurate annotations [14].

Query-dependent AC: Since there may be more or less difficult queries, the cost of annotating a query may depend on the query itself. For example, assessing the hosting capacity of a large and complex low-voltage grid may require more time than assessing a small and simple grid. Another example is the annotation of voice mails, where the duration of a voice mail is used as a proxy of the AC, e.g., 0.01 US dollar per second [50]. For the classification of documents, the number of words or characters in a document is often correlated to the AC [34]. Additionally, the query type affects the AC. For example, comparing two instances and deciding whether both belong to the same class is often easier than assigning an instance to one of many classes [79], [80].

Query- and Annotator-dependent AC: If the query and annotator-dependent cost schemes are considered, the AC varies across the pairs of query and annotator [81]. This cost scheme fits scenarios in which annotators are paid according to their individual hourly wages and the annotation time depends on the query [11].

The exact computation of AC depends on the underlying scheme. If we exemplary assume annotator-dependent AC with $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_M)$ and $\nu_m \in \mathbb{R}_{>0}$ representing the payment of querying annotator a_m, we would obtain

$$\mathrm{AC}(\mathcal{D}(t) \mid \boldsymbol{\nu}) = \sum_{m=1}^{M} \nu_m \cdot N_m^{(t)}, \tag{8}$$

$$N_m^{(t)} = \sum_{(q,a,z) \in \mathcal{D}(t)} \delta(a \doteq a_m), \tag{9}$$

where $\doteq$ denotes a Boolean comparison and the indicator function $\delta : \{\text{false}, \text{true}\} \rightarrow \{0, 1\}$ returns one if the argument is true and zero otherwise. Correspondingly, $N_m^{(t)} \in \mathbb{N}$ is the number of annotations provided by annotator a_m until the start of step t. In certain scenarios, such an exact specification of the AC in advance of querying is difficult. This is when the annotation time is the major cost factor. Therefore, an AL strategy is required to estimate the AC before querying an annotator.

C. LITERATURE OVERVIEW

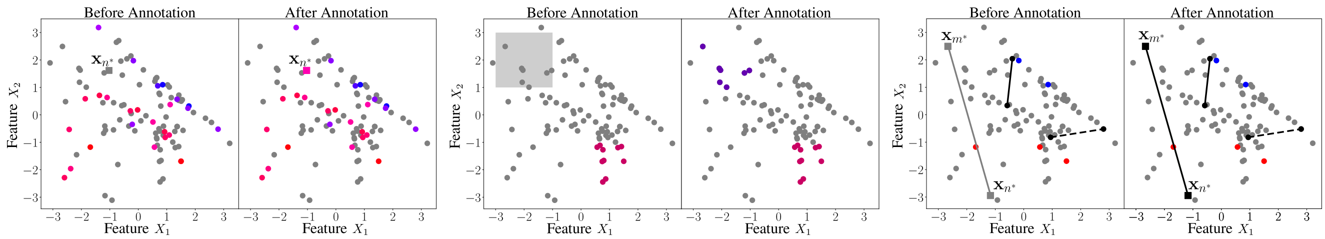
Table 1 gives a literature overview of real-world AL strategies explicitly modeling imbalanced MC or AC. The first part of this table lists strategies being MC-sensitive, i.e., class-dependent or instance-dependent. The second part summarizes strategies taking imbalanced AC into account, i.e., annotator- and/or query-dependent. Each strategy is categorized according to its cost scheme, the type of classification (binary vs. multi-class), and its predefined or estimated required cost information (cost matrix, annotation time, etc.). Additionally, we provide a brief description of each strategy's

TABLE 1. Part 1: Literature overview of cost-sensitive real-world AL strategies.

Strategy	Cost Scheme	Classification	Cost Information
Misclassification Cost (MC)			
Margineantu [82], Kapoor et al. [50], Joshi et al. [79, 80]	class-dependent MC	multi-class	cost matrix (predefined)
	These strategies compute the expected reduction of MC on the annotated and/or non-annotated set. For this purpose, they simulate the annotation of an instance and its addition to the classification model's training set. We can interpret these strategies as cost-sensitive variants of EER.		
Liu et al. [83]	class-dependent MC	multi-class	cost matrix (predefined)
	This strategy extends traditional US by making use of self-supervised training. Therefor, a cost-sensitive classification model is trained on instances annotated by annotators and instances annotated by a cost-insensitive classification model.		
Chen and Lin [84]	class-dependent MC	multi-class	cost matrix (predefined)
	The first variant of this strategy computes the maximum expected MC of an instance. In contrast, the second variant computes the cost-weighted minimum margin between the two predictions with the lowest estimated MCs.		
Krempl et al. [85]	class-dependent MC	binary	cost ratio of false negative vs. false positive (predefined)
	This strategy computes the density-weighted expected MC reduction in an instance's neighborhood within the feature space. Therefor, it simulates the annotation of an instance and its addition to the classification model's training set.		
Käding et al. [86]	class-dependent MC	multi-class	cost function (predefined)
	This strategy computes the classification model's expected change by simulating the annotation of an instance.		
Nguyen et al. [87]	class-dependent MC	binary	cost matrix (predefined)
	This strategy computes the expected MC reduction when obtaining an annotation from an error-prone annotator and from an infallible expert. Therefore, it employs a cost-sensitive variant of EER.		
Huang and Lin [88]	class-dependent MC	multi-class	cost matrix (predefined)
	This strategy is based on a cost embedding approach, which transfers the MC information into a distance measure of a latent space. Utilities are defined as expected MCs that are represented through distances in the latent space.		
Min et al. [89]	class-dependent MC	multi-class	cost matrix (predefined)
	This strategy queries only annotations for instances with MCs being higher than their respective ACs. The MCs are estimated through a cost-sensitive k-nearest neighbor model.		
Wu et al. [90], Wang et al. [91]	class-dependent MC	binary	cost matrix (predefined)
	These strategies employ a density-based clustering technique to construct a master tree of instances. In an iterative process, this master tree is subdivided into blocks and for each block an estimated MC-optimal number of instances are annotated.		
Krishnamurthy et al. [92, 93]	instance-dependent MC	multi-class	cost of predicting a class label for an instance (estimated)
	These strategies query MC information per instance and class from an annotator. Their idea is to query the actual MC information for the class label, for which an instance has the largest estimated MC range.		
Annotation Cost (AC)			
Zheng et al. [14], Chakraborty [94]	annotator-dependent AC	mutli-class	cost of querying an annotator (predefined)
	These strategies solve an optimization problem to specify a subset of annotators with low ACs and high performances.		
Moon and Carbonell [95], Huang et al. [96]	annotator-dependent AC	multi-class	cost of querying an annotator (predefined)
	These strategies compute the annotator performance per AC unit to prefer annotators with high performances and low ACs.		
Nguyen et al. [87]	annotator-dependent AC	multi-class	cost ratio of querying crowd worker vs. expert (predefined)
	This strategy normalizes the utility of querying an expert or crowd worker by their respective ACs to find a trade-off between expensive but correct expert annotations and cheap but error-prone crowd worker annotations.		
Margineantu [82], Donmez and Carbonell [38, 39]	query-dependent AC	multi-class	cost of annotating a query (predefined)
	These strategies subtract the query's individual AC from its utility to prefer highly useful queries with low ACs. These ACs are assumed to be known in advance for each query or to follow a predefined model.		
Kapoor et al. [50]	query-dependent AC	multi-class	cost of annotating an instance with unknown class label (estimated)
	This strategy assumes that the AC of a queried instance depends on its class. Since the true class labels of non-annotated instances are unknown, the strategy uses the estimated class membership probabilities to compute the expected AC of querying an instance's annotation.		
Joshi et al. [79, 80]	query-dependent AC	multi-class	number of comparisons per query (estimated)
	These strategies use multiple comparison queries to reveal the class label of a non-annotated instance. The expected number of comparisons required to reveal an instance's class label is used as an AC proxy and subtracted from an instance's utility.		
Tsou and Lin [97]	query-dependent AC	multi-class	cost of annotating a query (estimated)
	This strategy builds a decision tree throughout the AL process. It computes the average AC of the already annotated instances in each leaf of this tree. These AC estimates are used to normalize the query utilities of the instances in the respective leaves.		

TABLE 1. *(Continued.)* Part 2: Literature overview of cost-sensitive real-world AL strategies.

Strategy	Cost Scheme	Classification	Cost Information
Annotation Cost (AC)			
Settles et al. [81], Haertel et al. [98], Tomanek and Hahn [99], Wallace et al. [100]	query-dependent AC	multi-class	annotation time per query (estimated)
	These strategies estimate the annotation time per query. Therefore, they use either historical data in form of logged annotation times or employ prior knowledge regarding a domain, e.g., the number of words when annotating text. The estimated annotation times are considered by normalizing query utility or employing a linear rank combination of utilities and annotation times.		
Wallace et al. [11]	annotator-, query-dependent AC	multi-class	annotation time per query (estimated) + cost per time unit (predefined)
	This strategy computes ACs by multiplying the annotation times (estimated through the number of words in a document) with the respective salaries (predefined) of the annotators. Annotators with low estimated ACs are more often queried.		
Arora et al. [34]	annotator-, query-dependent AC	multi-class	annotation time per query-annotator pair (estimated)
	This strategy estimates the annotation time as a function of the annotator and the query. Therefor, it uses features to describe the query and the annotator in combination with historical data in form of logged annotation times.		



(a) Example of an instance query. (b) Example of a region query. (c) Example of a comparison query.

FIGURE 5. Illustration of query types within the feature space: For a binary classification problem, the two-dimensional instances of an artificially generated set $\mathcal{X} \subset \mathbb{R}^2$ are plotted according to their feature values. Probabilistic annotations are depicted by using the corresponding proportions of the red and blue colors.

main idea. A more in-depth analysis of them is provided in the appendices of this survey as supplementary material.

IV. INTERACTION SCHEMES

Interaction with human annotators forms an essential part of AL. In this survey, we focus on the AL typical query-annotation-based interaction. For this purpose, we provide an overview of different query types and annotation types based on the literature. At the end of this section, we present a literature overview of existing real-world AL strategies using different combinations of queries and annotations as interaction schemes.

A. QUERY TYPES

The set of possible queries $\mathcal{Q}_{\mathcal{X}}$ specifies how a real-world AL strategy can interact with the available annotators $\mathcal{A}$. Depending on the underlying classification problem, there are different possibilities to design these queries. In the literature, we identified the following three most common query types:

Instance queries ask for information on a specific instance $\mathbf{x}_{n^*}$ and are the most common query type. Next to class labels, a query may request additional information. Concrete examples are presented in [44], [101], where annotators are asked for confidence scores interpreted as proxies of an instance's class membership probabilities. An illustration is given in Fig. 5(a) by marking the selected instance $\mathbf{x}_{n^*}$ for which the class membership probability for the blue class is expected as an annotation.

Region queries do not query information regarding a specific instance, but ask annotators to provide information about an entire region in the feature space [102]. For this purpose, the query is to be formulated in an appropriate and human-readable representation [103]. A common way to achieve this requirement involves formulating premises of sharp or possibilistic classification rules by defining conditions on the value ranges of features [104]. An example of such a region query is depicted by the gray rectangle in Fig. 5(b). Although a region query provides class information about many instances, this type of query differs from batch mode AL, where each instance of a selected batch is annotated individually [23].

Comparison queries enhance the learning process by obtaining relative information between instances [47]. For example, the comparison query, illustrated in Fig. 5(c), compares two instances $\mathbf{x}_{n^*}$ and $\mathbf{x}_{m^*}$ by requesting whether they belong to the same class or not [46], [105]. Regarding the ordinal grid classification example, another conceivable comparison query may ask which of two grid instances has a superior hosting capacity.

Going beyond these three query types, we will present our own proposals for query types as future research directions in Section VII.

B. ANNOTATION TYPES

Usually, the type of an annotation depends on the query itself. In this survey, we differentiate between the following three annotation types:

Distinct annotations are the simplest form of annotations. They represent categorical information without the scope of interpretation. Most AL strategies use them to encode class labels. Other AL strategies expect a simple `yes` or `no` as a distinct annotation [46], [47]. Furthermore, they can encode a sorting of instances in case of a comparison query [106].

Soft annotations allow for the representation of continuous information. They are often inaccurate and subjective. Many AL strategies use them to obtain information on the confidence of a provided class label by requesting a numerical value in a continuous confidence interval [43], [101]. Another example is the use of probabilistic labels as gradual annotations [44], [107], which enhanced the classification performance for certain tasks, e.g., in the medical domain [108].

Explanatory annotations are the most informative type of annotations. Instead of only communicating a distinct or soft decision, an explanatory annotation also explains why a certain decision has been made. An exemplary explanation would be: "The instance $\mathbf{x}_n$ does not belong to the positive class because its feature value x_{nd} is too low." [109].

C. LITERATURE OVERVIEW

A query mostly requests information of a specific kind. Accordingly, the query and annotation types are closely coupled. Table 2 gives a literature overview of existing combinations of queries and annotations as interaction schemes. The query "To which class does instance $\mathbf{x}_n$ belong?" known already from the traditional AL setting is excluded. A more in-depth analysis of the real-world AL strategies in Table 2 with a focus on their query utility measures is provided in the appendices of this survey as supplementary material.

V. ANNOTATOR PERFORMANCE MODELS

The error-proneness of annotators poses a major challenge in real-world AL [36]. In this section, we discuss the typical factors influencing the performance of error-prone annotators. Moreover, we identify three different types of annotator performance. At the end of this section, we present a literature overview of existing annotator performance models.

A. INFLUENCE FACTORS

We refer to "annotator performance" as a general term for the quality of the annotations obtained from an annotator. There is no clear definition of this term, but there exist several concrete interpretations, e.g., label accuracy [123], confidence [101], uncertainty [70], reliability [124], etc. Such an interpretation is closely coupled to the annotation type and the expected optimal annotation of a query.

The annotator performance may be affected by various factors [125], [126], and the most prominent ones identified in the AL literature are given in the following:

The **domain knowledge** of annotators has an essential impact on their performances [127]. Insufficient knowledge leads to a deterioration of the annotator performance. In complex tasks, such as assessing the hosting capacity of a low-voltage grid, a certain level of domain knowledge is indispensable.

The **query difficulty** affects the probability of obtaining an optimal annotation [12], [128], [129]. For example, in recognition of hand-written digits, it is often more challenging to differentiate between the digits 1 and 7 than discriminating between the digits 1 and 8 [44]. Next to the subject of a query, also its type can be crucial for the performance of an annotator [80].

The **ability for a reliable self-assessment** of annotators plays a central role, particularly in scenarios where queries ask for confidence scores as annotations [101]. Although empirical studies [11], [130] have shown that annotators can reliably estimate their performances in some domains, the Dunning-Kruger-effect [131] states that, in particular, unskilled annotators provide not only erroneous annotations, but they also cannot realize their mistakes. This effect has also been confirmed in a large-scale crowd-sourcing study [132].

The **motivation or level of interest** of an annotator may influence the elaborateness during the annotation process. For example, in a crowdsourcing study analyzed in [127], more interested annotators performed superiorly.

The **payment** of an annotator may have a significant impact on the annotator performance, such that well-paid annotators provide more high-quality annotations. In a crowdsourcing environment, the improvement of the annotation quality has been confirmed by increasing the pay from 0.10$ to 0.25$ per query [127].

An annotator has to be **concentrated** when annotating a query [133]. Otherwise, annotation mistakes arise because of missing mindfulness or tiredness.

A constant stream of queries of the same type may be annoying for the annotator [134]. Therefore, the **way of interaction** between the AL strategy and an annotator may influence the annotation results and needs to be designed appropriately. For example, different interaction schemes can lead to different degrees of an annotator's enjoyability, as experimentally shown in [135].

The **learning aptitudes** of annotators are also crucial for their performances. For example, one could teach the annotators to provide high-quality annotations [125].

The **collaboration** between annotators is interlinked with their performances. Incorporating mechanisms for collaboration can strongly improve the annotation quality [136].

B. ANNOTATOR PERFORMANCE TYPES

Modeling and quantifying the influence of each of the previously listed factors on annotator performance is infeasible. Instead, existing annotator models abstract from these factors to estimate annotator performance. In the literature, we identified three different types of annotator performances. Therefor, we generalize the class label noise taxonomy,

TABLE 2. Literature overview of combinations of queries and annotations employed by real-world AL strategies.

Strategy	Query	Annotation
Instance Queries		
Hu et al. [110]	Does instance $\mathbf{x}_n \in \mathcal{X}$ belong to concept $\mathcal{K} \subset \Omega_Y$?	distinct annotations: $\Omega_Z = \{\texttt{yes}, \texttt{no}\}$
Bhattacharya and Chakraborty [111]	To which class in $\{y^{(1)}, \ldots, y^{(n)}\} \subset \Omega_Y$ does instance $\mathbf{x}_n \in \mathcal{X}$ belong?	distinct annotations: $\Omega_Z = \Omega_Y$
Cebron et al. [112]	To which class does instance $\mathbf{x}_n \in \mathcal{X}$ not belong?	distinct annotations: $\Omega_Z = \mathcal{P}(\Omega_Y)$
Donmez and Carbonell [38, 39], Wallace et al. [11], Fang and Zhu [113], Zhong et al. [45], Käding et al. [86]	Provided that you are confident: What is the class label of instance $\mathbf{x}_n \in \mathcal{X}$?	distinct annotations: $\Omega_Z = \Omega_Y \cup \{\texttt{uncertain}\}$
Donmez and Carbonell [38, 39], Ni and Ling [101], Calma et al. [44]	What is the class label of instance $\mathbf{x}_n \in \mathcal{X}$ and how confident are you?	soft annotations: $\Omega_Z = \Omega_Y \times \Omega_C$ where Ω_C denotes the set of possible confidence scores
Song et al. [43]	How confident are you that instance $\mathbf{x}_n \in \mathcal{X}$ belongs to the positive class?	soft annotations: $\Omega_Z = [-1, 1]$ with $z \in \Omega_z$ indicating the confidence that $\mathbf{x}_n$ belongs to the positive class
Biswas and Parikh [109]	Does instance $\mathbf{x}_n \in \mathcal{X}$ belong to class $y \in \Omega_Y$? If this is not the case, can you explain the reason?	explanatory annotations: $\Omega_Z = \{\texttt{yes}\} \cup \Omega_E$ with Ω_E representing the set of explanations
Teso and Kersting [114]	Does instance $\mathbf{x}_n \in \mathcal{X}$ belong to class $y \in \Omega_Y$ because of explanation $e \in \Omega_E$?	explanatory annotations: $\Omega_Z = \{\texttt{yes}\} \cup \Omega_Y \cup \Omega_E$ with Ω_E representing the set of explanations
Region Queries		
Druck et al. [115], Settles [116]	For which classes is a positive feature value $X_d > 0$ highly indicative?	distinct annotations: $\Omega_Z = \mathcal{P}(\Omega_Y)$ with $z \subseteq \Omega_Y$ indicating the set of possible classes
Du and Ling [102]	What is the proportion of positive instances in the region described by the constellation of categorical features, e.g., $X_1 \doteq 1 \wedge X_4 \doteq 0 \wedge X_8 \doteq 0$?	soft annotations: $\Omega_Z = [0, 1]$ with $z \in \Omega_Z$ indicating the proportion of positive instances
Du and Ling [10], Luo and Hauskrecht [117, 118, 103], Rashidi and Cook [104], Haque et al. [119]	What is the proportion of positive instances in the region described by the feature constellation, e.g., $X_1 \in [0, 2] \wedge X_2 \leq 10 \wedge X_3 \doteq 3$?	
Comparison Queries		
Fu et al. [46, 105], Joshi et al. [79, 80]	Do instance $\mathbf{x}_n \in \mathcal{X}$ and instance $\mathbf{x}_m \in \mathcal{X}$ belong to the same class?	distinct annotations: $\Omega_Z = \{\texttt{yes}, \texttt{no}\}$
Xiong et al. [120]	Is instance $\mathbf{x}_n \in \mathcal{X}$ more similar to instance $\mathbf{x}_m \in \mathcal{X}$ than instance $\mathbf{x}_o \in \mathcal{X}$?	distinct annotations: $\Omega_Z = \{\texttt{yes}, \texttt{no}, \texttt{uncertain}\}$
Kane et al. [47], Xu et al. [121], Hopkins et al. [122]	Is instance $\mathbf{x}_n \in \mathcal{X}$ more likely to belong to the positive class than instance $\mathbf{x}_m \in \mathcal{X}$?	distinct annotations: $\Omega_Z = \{\texttt{yes}, \texttt{no}\}$
Qian et al. [106]	What is the decreasing order of the instances $\{\mathbf{x}_n, \mathbf{x}_m, \mathbf{x}_o\} \subset \mathcal{X}$ regarding their similarities to instance $\mathbf{x}_p \in \mathcal{X}$?	distinct annotations: Ω_Z consists of all possible ordering of the available instances, e.g., $(\mathbf{x}_m, \mathbf{x}_o, \mathbf{x}_n) \in \Omega_Z$

presented by Frénay and Verleysen [137], to the setting of real-world AL by including queries and annotations instead of instances and classes. The resulting statistical taxonomy of annotator performance types is presented in Fig. 6. There are four random variables depicted as nodes: Q is the query, Z is the optimal annotation, Z_m is the annotation provided by annotator a_m, and P_m is the variable indicating the performance of the annotator a_m. In the simplest case, P_m is a binary variable to represent whether an annotator provides the optimal annotation ($P_m = 1$) or not ($P_m = 0$). We denote observed variables by shading the corresponding nodes, whereas the other nodes represent latent variables. The variable t is a deterministic parameter denoting the time. Arrows represent statistical dependencies, e.g., the optimal annotation always depends on the underlying query. The dashed arrow between the annotator performance variable P_m and the time t indicates an optional dependency. If this dependency is considered, the annotator performance is time-varying [48]. Otherwise, it is assumed to be persistent. In the first case, the annotator performance is constant during the entire annotation process. In the latter case, the annotator performance may increase due to the learning progress of an annotator [70], [81] or may decrease because of exhaustion or emerging boredom [135]. We provide more details regarding the three annotator performance types in the following:

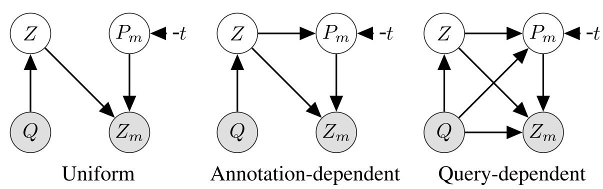


FIGURE 6. Proposed statistical models of annotator performance types.

Uniform annotator performance: The annotator performance depends only on the characteristics of the annotator. As a result, the query itself or the query's optimal annotation has no influence. An example is given in Fig. 7, where an annotator has the constant probability of 90% to recognize a hand-written digit correctly.

Optimal Annotation:	1	7	7
Query: Which digit is shown in the image?			

Annotator Performance:			
Uniform:	90%	90%	90%
Annotator-dependent:	90%	70%	70%
Query-dependent:	90%	95%	40%

FIGURE 7. Illustration of annotator performance types.

Annotation-dependent annotator performance: The annotator performance depends next to the annotator's characteristics on the optimal annotation for a query. An example is given in Fig. 7, where an annotator is better at identifying the digit 1 (constant correctness probability of 90%) than the digit 7 (constant correctness probability of 70%) in images of hand-written digits.

Query-dependent annotator performance: The annotator performance depends on the annotator's characteristics, the query, and the optimal annotation. An example is given in Fig. 7, where an annotator has a low probability to correctly identify the third digit as 7 because it can be misinterpreted as the digit 2.

C. LITERATURE OVERVIEW

During the AL process, the performances of the annotators are estimated by annotator models. Table 3 provides a literature overview, including a categorization of those models. Next to the assumptions regarding the type of annotator performance, we use several other factors to categorize different annotator models. In particular, the query and annotation types described in Section IV are essential properties of an annotator model. However, to the best of our knowledge, existing annotator models focus on instance queries such that no column for the query type is present in Table 3. As a further category, we differentiate between the assumed relation of the annotators. In the case of multiple annotators, they are either independent or collaborative. If a model can work with a single annotator, the term single is denoted for this category. Furthermore, we indicate in Table 3 whether an annotator model allows for the integration of prior knowledge regarding the performances of annotators. Additionally, we provide a brief description of each annotator model's main idea. A more in-depth analysis of these annotator models is provided in the appendices of this survey as supplementary material.

VI. SELECTION ALGORITHMS

The selection of query-annotator pairs is based on a selection algorithm. It uses the query utility measure ϕ and the annotator performance measure ψ as basis to specify $\mathcal{S}(t) \subseteq \mathcal{Q}_{\mathcal{X}} \times \mathcal{A}$ as the set of query-annotator pairs in each AL iteration cycle $t \in \mathbb{N}$. In this context, we differentiate between two types of selection algorithms, explained in the following. At the end of this section, we present a literature overview of existing selection algorithms.

A. SEQUENTIAL SELECTION

Sequential selection of queries and annotators is made in two steps. In the first step, one or multiple (in the case of batch mode AL) queries with the highest utilities are selected. In a second step, corresponding annotators are selected and assigned to the respective queries, e.g., a predefined number of the annotators with the highest estimated performances per query [139]. Ideally, the selected annotators lead to low AC while providing high accuracy annotations. The main motivation for a sequential selection is to emphasize useful queries by selecting them in advance of the annotators. Moreover, the issue of annotator selection reduces to determining a ranking of the annotators regarding a selected query. As a result, not the exact but only the relative differences between the performances of the annotators are crucial for the annotator selection.

B. JOINT SELECTION

Selecting queries without considering the annotators' performances can result in low-quality annotations because there is no guarantee that at least one annotator has a sufficient performance regarding a selected query [45]. This problem can be resolved by applying a selection algorithm jointly selecting queries and annotators. For this purpose, the query utility and the annotator performance measure are to be combined appropriately, e.g., by taking their product [96]. Compared to the sequential selection of queries and annotators, the joint selection comes with higher computational complexity. Instead of computing the annotator performance estimates only for the selected queries, the annotator performance estimates are required for each possible query. Moreover, exact estimates regarding the annotator performance are more crucial since the annotator performance estimates are directly integrated into the selection criterion. If these estimates are unreliable, not only the annotator selection will be negatively affected but also the query selection.

C. LITERATURE OVERVIEW

Table 4 provides an overview of selection algorithms employed by existing real-world AL strategies, which select query-annotator pairs. Next to the differentiation between a sequential and joint selection of queries and annotators, the number of selected queries and annotators per learning cycle is of interest. Selecting only a single query-annotator pair is often easier than selecting a batch of query-annotator pairs. In the latter case, the selection algorithm must ensure that queries are diverse. Otherwise, redundant information is queried. Moreover, multiple annotators are to be distributed across queries. To differentiate between both settings, we denote either single or batch for the query and annotator selection categories in Table 4. A few selection algorithms consider criteria beyond annotator performance and query

TABLE 3. Part I: Literature overview of annotator performance models employed by real-world AL strategies.

Strategy	Annotation Type	Temporal Annotator Performance	Annotator Relation	Prior Knowledge
Uniform Annotator Performance				
Donmez et al. [123], Zheng et al. [14]	distinct: class labels	persistent	independent	yes
	This annotator model estimates the true class label of an instance by means of majority voting. Following the interval estimation method [138], the majority votes are then used to evaluate the upper bound of the fraction of correctly annotated instances as performance estimate for each annotator.			
Donmez et al. [48]	distinct: class labels	time-varying	independent	yes
	This annotator model models the performance of each annotator as a time-varying latent state sequence. For this purpose, it assumes that the change in the annotator performance from one to the next state follows a Gaussian distribution with a zero-mean and a known variance, which is shared among all annotators.			
Long et al. [139, 140], Long and Hua [141]	distinct: binary class labels	persistent	independent	no
	These annotator models, based on a probabilistic model with Gaussian processes [142], estimate a single performance value per annotator by comparing the provided annotations to the estimated true annotations. The performance value of an annotator indicates the probability that this annotator assigns the correct class label to an instance.			
Annotation-dependent Annotator Performance				
Wu et al. [143]	distinct: binary class labels	persistent	independent	yes
	This annotator model, based on the logistic regression model proposed by Raykar et al. [144], estimates the performance of an annotator in dependence of an instance's (unknown) true class label. Using the maximum a posteriori criterion, the model's training follows the expectation-maximization algorithm [145], which iteratively estimates the true class labels (expectation-step) to evaluate the probability of a correct annotation for each class-annotator pair (maximization-step).			
Rodrigues et al. [146]	distinct: binary class labels	persistent	independent	no
	This annotator model, based on a Gaussian processes [142] framework and expectation propagation [147], estimates the class-dependent specificity and sensitivity of each each annotator by comparing the provided annotations to the estimated true annotations.			
Moon and Carbonell [95]	distinct: class labels	persistent	independent	no
	This annotator model expects an initial set of instances annotated by each annotator. The true class labels of these instances are estimated through majority voting. Subsequently, the performance of an annotator is computed as the annotation accuracy, i.e., estimated fraction of correct annotations, per class.			
Nguyen et al. [87]	distinct: class labels	persistent	independent	yes
	This annotator model differs between infallible experts and error-prone crowd workers. The performance of the latter ones is estimated by comparing their provided class labels with the expert class labels. For this purpose, the model computes a confusion matrix including a Bayesian prior for the group of crowd workers.			
Query-dependent Annotator Performance				
Wallace et al. [11]	distinct: binary class labels and `uncertain`	persistent	independent	yes
	This annotator model relies on domain information in form of annotators' pay grades. Therefore, it assumes that the pay grades of the annotators are highly correlated with their performances.			
Donmez and Carbonell [38, 39]	soft: class labels and confidence scores	persistent	independent	no
	This annotator model uses the annotators' confidence scores as proxies of their performances. Using k-means clustering [148], an annotator is queried to annotate the $k \in \mathbb{N}$ instances closest to the respective k cluster centroids. It is assumed that instances belonging to a cluster, whose centroid has a high-confidence annotation, will be accurately annotated by the corresponding annotator.			
Du and Ling [10]	distinct: binary class labels	persistent	single	no
	Since the classification model is trained under a single annotator's supervision, this annotator model assumes that the classification model behaves similarly to the annotator. As a result, the annotator performance estimates near the classification model's decision boundary are lower than in regions where the classification model is certain.			
Yan et al. [68, 149]	distinct: binary class labels	persistent	independent	no
	This annotator model, based on a logistic regression model proposed in [150], estimates the performance of an annotator in dependence of an instance and its true class label. Using the maximum likelihood criterion, the model's training follows the expectation-maximization algorithm [145], which iteratively estimates the true class labels (expectation-step) and uses them to evaluate the annotator performance for each instance-annotator pair (maximization-step).			
Ni and Ling [101]	soft: binary class labels and confidence scores	persistent	single/independent	no
	This annotator model uses the confidence scores provided by the annotators as proxies of their performances. For non-annotated instances, these scores are estimated using the (inverse-)distance-weighted k-nearest neighbors rule [151]. Accordingly, an annotator's performance for an instance is defined as the weighted mean confidence score of its k-nearest neighbors being already annotated by this annotator.			

TABLE 3. *(Continued.)* **Part II: Literature overview of annotator performance models employed by real-world AL strategies.**

Strategy	Annotation Type	Temporal Annotator Performance	Annotator Relation	Prior Knowledge
Query-dependent Annotator Performance				
Fang et al. [70]	distinct: binary class labels	time-varying	collaborative	no
	This annotator model interprets the performance as uncertainty of an annotator regarding high-level concepts, e.g., sports, politics, and culture in case of document classification. These concepts are latent variables and modeled through a Gaussian mixture model [152]. An instance may belong to multiple concepts. Using the maximum likelihood criterion, the model's training follows the expectation-maximization algorithm [145], which iteratively estimates the true class labels (expectation-step) and takes them as basis for evaluating an annotator's uncertainty in annotating an instance (maximization-step).			
Fang and Zhu [113]	distinct: binary class labels and `uncertain`	persistent	single/independent	no
	This annotator model expects the annotator to provide `uncertain` as annotation, if the annotator does not know an instance's true class label. Using this information, the model characterizes the performance of the annotator by training a classifier to estimate the probability whether an instance will not belong to the annotator's uncertain knowledge set.			
Fang et al. [153, 154]	distinct: binary class labels	persistent	independent	no
	This annotator model assumes that the performance of an annotator depends on a high-level representation of an instance's features and their true class label. This dependency is indirectly modeled by introducing a latent variable for the expertise of each annotator. The expertise of an annotator is then computed as weighted linear combination of the instance's high-level features.			
Zhao et al. [12]	distinct: binary class labels	persistent	independent	no
	This annotator model estimates the annotator performance through two latent variables, namely, the query difficulty and the query-independent expertise of an annotator. For example, for annotators with high expertise or for easy queries, the probability of providing the true class label is high. The query difficulty and annotator expertise are latent and therefore iteratively estimated through the expectation-maximization algorithm [145].			
Zhong et al. [45], Käding et al. [86]	distinct: class labels and `uncertain`	persistent	single/independent	no
	These annotator models allow an annotator to provide `uncertain` as annotation in case of a lack of knowledge regarding an instance's class membership, otherwise she/he provides a class label. The instances annotated with class labels (positive class) and the ones annotated with `uncertain` (negative class) form a binary classification problem. They are used to train an annotator model, i.e., a support vector machine [56] in [45] and Gaussian processes [142] in [86]. It predicts whether an annotator has sufficient knowledge to annotate an instance (positive class) or not (negative class).			
Huang et al. [96]	distinct: class labels	persistent	independent	no
	This annotator model expects an initial set of instances with true class labels and annotations of each annotator. The model assumes that an annotator has a similar performance on similar instances. Therefore, it estimates an annotator's performance for an instance by computing the annotation accuracy regarding the instance's nearest neighbors in the initial set.			
Yang et al. [155]	distinct: class labels	persistent	independent	no
	This annotator model learns a low-dimensional embedding for each annotator to capture the annotator's expertise regarding latent topics. Additionally, an embedding for each instance is learned as representation by the latent topics. Both embeddings are combined to estimate the performance of an annotator. Since these embeddings are latent variables, they are learned through the expectation-maximization algorithm [145].			
Chakraborty [94]	distinct: class labels	persistent	independent	no
	This annotator model expects an initial set of instances with true class labels and annotations of each annotator. Since the true class labels are known in this set, the mistakes of each annotator can be determined on this set. A binary logistic regression classifier is then trained for each annotator separately. The trained logistic regression model of an annotator estimates her/his performance as the probability of obtaining a correct annotation for a certain instance.			
Herde et al. [69]	distinct: class labels	persistent	independent	yes
	This annotator model estimates the performance of an annotator for a certain instance in form of a Beta distribution. This distribution is parameterized by the number of estimated false and true annotations in the local neighborhood of an instance. A false or true annotation of an annotator is identified by comparing the annotations of a single annotator to the predictions of a classifier trained with the annotations of the other annotators.			

utility, e.g., a collaboration between annotators. We denote these criteria accordingly in Table 4. Additionally, we provide a brief description of each selection algorithm's main idea. A more in-depth analysis of them is provided in the appendices of this survey as supplementary material.

VII. FUTURE RESEARCH DIRECTIONS

This section proposes some future research directions resulting from analyzing the real-world AL strategies discussed in the previous sections. We structure them into three categories to distinguish between challenges that strongly relate to this survey and those that go partially beyond it. Although we define these addressable challenges separately, they are not entirely solvable without taking a holistic view.

A. ACTIVE LEARNING FOR CLASSIFICATION

1) MULTI-CRITERIA COST FUNCTIONS

The majority of existing real-world AL strategies minimizes the number of queries and misclassifications. However, in real-world applications, the ACs are often unknown in advance and may be query- and annotator-dependent. Furthermore, the computation of the MC is related to the

TABLE 4. Literature overview of selection algorithms employed by real-world AL strategies.

Strategy	Query Selection	Annotator Selection	Criteria Beyond Utility and Performance
Sequential Selection			
Ni and Ling [101], Wu et al. [143], Rodrigues et al. [146], Fang et al. [153, 154], Zhong et al. [45]	single	single	none
	These strategies select the query with the highest estimated utility. Subsequently, they select the annotator with the highest estimated performance regarding the annotation of this query.		
Wallace et al. [11]	single	single	workload of annotators
	This strategy selects either a non-annotated query with the highest estimated utility or a query for re-annotation. The annotator selection follows a categorical distribution whose parameters reflect a certain objective, e.g., balancing the annotation workload among annotators.		
Zhao et al. [12]	single	single	none
	This strategy selects the query with the highest estimated utility. The annotator selection follows one of two options. On the one hand, an annotator can be selected with a probability proportional to her/his estimated performance. On the other hand, the estimated best annotator is either selected with a predefined probability or a random one.		
Donmez et al. [123]	single	batch	none
	This strategy selects the query with the highest estimated utility. Subsequently, it selects an adaptive number of annotators with the highest estimated performances.		
Zheng et al. [14]	single	batch	none
	This strategy selects the query with the highest estimated utility. In an exploration phase, it assigns an adaptive number of annotators with the highest estimated performances to this query. In the subsequent exploitation phase, a fixed subset of annotators with low ACs and high performances is determined and always selected.		
Fang et al. [70]	single	batch	collaboration between annotators
	This strategy selects the query with the highest estimated utility. Subsequently, it selects not only the annotator with the highest estimated performance but additionally the annotator with the lowest estimated performance. This way, the estimated best annotator can teach the estimated worst annotator.		
Long et al. [139, 140], Long and Hua [141]	single	batch	none
	These strategies select the query with the highest estimated utility. Subsequently, they select a predefined number of annotators with the highest estimated performances.		
Yang et al. [155]	batch	batch	none
	This strategy selects a predefined number of queries with the highest estimated utilities. Subsequently, it assigns to each of these selected queries the respective annotator with the highest estimated performance.		
Joint Selection			
Donmez and Carbonell [38, 39], Moon and Carbonell [95], Huang et al. [96]	single	single	none
	These strategies select the query-annotator pair whose product of estimated query utility and annotator performance is the highest.		
Yan et al. [149]	single	single	none
	This strategy jointly selects a query and annotator by solving a linearly constrained and bi-convex optimization problem. Its goal is to find the optimal trade-off between a highly useful query and a high-performance annotator.		
Yan et al. [68]	single	single	none
	This strategy combines the query utility information and annotator performance information through a mutual information criterion [156] as the joint selection criterion for a query-annotator pair.		
Nguyen et al. [87], Herde et al. [69]	single	single	none
	These strategies jointly select a query and annotator by incorporating the estimated performance of an annotator (group) into the query utility measure quantifying the performance gain of the classification model.		
Chakraborty [94]	batch	batch	query diversity
	This strategy jointly selects a batch of query-annotator pairs by solving a linear programming problem. Its solution balances the trade-off between useful queries, accurate annotators, and a small redundancy between these queries.		

application at hand. Therefore, an AL strategy needs to accept a user-defined objective function as input. This function needs to account for additional criteria, such as balancing the workload between annotators [11].

2) NOVEL QUERY TYPES AND A COMBINATION OF THEM

Present AL strategies focus on collecting novel information relevant to the classification model. However, a query may not only improve the classification model but additionally the queried annotator [157]. For example, a strategy could ask "Are you certain that instance $\mathbf{x}_n \in \mathcal{X}$ belongs to class $y \in \Omega_Y$? Previously, you stated that the similar instance $\mathbf{x}_m \in \mathcal{X}$ belongs to class $y' \in \Omega_Y$?". Such a query may help the annotator to learn from previous annotation mistakes. Moreover, most pool-based AL strategies query class information of instances. However, recently, Liang *et al.* [158] proposed the strategy *active learning with contrastive natural language explanations* (ALICE). It uses queries of the form "How would you differentiate between the class $y \in \Omega_Y$ and class $y' \in \Omega_Y$?" in combination with explanatory annotations.

As a result, ALICE does not need a pool of non-annotated instances but only a small initial training set. Next to novel query types, future strategies may combine different query types to enhance interaction with annotators further.

3) BATCH SELECTION OF DIVERSE QUERIES AND ANNOTATORS

Deep learning model's generalization capabilities depend on a vast amount of data. Therefore, annotating single queries per AL cycle may be inappropriate [159]. Instead, a batch of diverse and useful queries is to be selected per AL cycle. Such a batch maximizes usefulness by avoiding redundancies. In a multi-annotator setting, assigning appropriate annotators to these queries is an additional challenge. For example, assigning all queries in a batch to a single annotator can be harmful because it could bias the performance estimates of the other annotators [146].

4) ADVANCED ANNOTATOR PERFORMANCE ESTIMATION

Existing annotator models are limited in their application due to their assumptions. On the one hand, most of them assume persistent annotator performances and thus disregard, e.g., learning aptitude, collaboration, or signs of fatigue. On the other hand, they do not incorporate background knowledge about the annotators, e.g., interests, skills, level of education, age, etc. Such knowledge may improve the annotator selection [160].

5) REALISTIC EVALUATION

Evaluating real-world AL strategies is more complex than assessing traditional AL strategies. In particular, the simulation of realistic experimental settings represents a challenge. For example, there is a need to collect real-world data sets processed by multiple annotators to verify the performance of AL strategies in multi-annotator settings. When collecting such data sets, it is infeasible to present each possible query to each annotator. Therefore, a further research direction is the simulation of annotators for different query types and with different assumptions regarding their types of performances. Moreover, an AL strategy may be evaluated in a real-world system [161] in addition to simulated experiments on benchmark data sets to verify its effectiveness regarding real-world applications.

B. ACTIVE LEARNING ISSUES BEYOND CLASSIFICATION

Although we focused on AL strategies for classification in this survey, their analysis provides insights beyond a classification setting. If we exemplify object detection in images, similar challenges arise when employing AL strategies. For example, relying on the number of annotated images as AC is not representative. Instead, the number of objects within an image is more appropriate [162] because annotating images with many objects is more time-intensive. Another example for object detection is the handling of error-prone annotators, where the AL strategy has additionally to assess the quality of provided bounding box annotations.

A challenge affecting pool-based AL with multiple annotators is the asynchronous nature of the annotation process [136]. This results from different working speeds of annotators, i.e., some annotators process queries faster than others. Due to this asynchronous nature, the selection of query-annotator pairs must be adaptive regarding the working states of the annotators. This is, in particular, true for stream-based AL.

Techniques of explainable artificial intelligence may improve the interaction between annotators and AL strategies. For example, the ML model can visualize its decision-making process such that an annotator can monitor the model's learning progress to correct wrong decisions [157].

C. ACTIVE LEARNING ISSUES BEYOND ARTIFICIAL INTELLIGENCE

Deploying AL strategies into real-world applications not only raises challenges in the scope of artificial intelligence but also involves research beyond it. One example is graphical user interfaces of the annotation process, which are crucial for the efficiency of the AL process. Studies have shown that an appropriate user interface design strongly decreases the annotation time and thus AC [116], [163]. Another example is the design of queries and annotations from a psychological perspective. On the one hand, queries are to be formulated neutral without a bias toward a specific annotation. On the other hand, annotations are to be comparable, particularly when asking for the annotators' self-assessments.

Another future research direction is integrating AL into further little to no explored application areas to exploit its full potential. For example, it can be employed in material science to actively design experiments in a more systematic way [164] or for automatic program repair [165] to save cost and time. Another example would be the review process in science, where AL can select appropriate reviewers as annotators for articles. Therefor, one could use feedback from authors of past conferences and the reviewers' background knowledge to train annotator models.

VIII. CONCLUSION

At the start of this survey, we pointed out unrealistic assumptions as disadvantages of traditional AL strategies. Based on that, we identified three crucial requirements for real-world AL strategies, i.e., estimating costs, asking alternative queries, and modeling annotator performances. Subsequently, we formalized the objective for classification tasks as the specification of the optimal annotation sequence leading to minimum MC and AC. Additionally, we proposed a novel AL cycle that generalizes the settings of the majority of existing real-world AL strategies. A strategy is part of a learning system in this cycle and comprises a query utility measure, an annotator performance measure, and a selection algorithm. We provided tabular literature overviews of existing real-world AL strategies regarding their cost types, their query- and annotation-based interaction, their handling of error-prone annotators, and their selection of query-annotator pairs. In addition, we analyzed real-world AL strategies in more detail and embedded them in our unifying mathematical notation in the appendices as supplementary material. These analyses resulted in the formulation of future research directions in the field of AL.

ACKNOWLEDGMENT

The authors would like to thank Daniel Kottke, Tuan Pham Minh, Lukas Rauch, Robert Monarch, and the anonymous reviewers for their comments greatly improving this survey.

REFERENCES

[1] L. Haddon, *Information and Communication Technologies in Everyday Life: A Concise Introduction and Research Guide*. Oxford, U.K.: Berg Publishers, 2004.

[2] C. Edwards, "Growing pains for deep learning," *Commun. ACM*, vol. 58, no. 7, pp. 14–16, 2015.

[3] Y. Roh, G. Heo, and S. E. Whang, "A survey on data collection for machine learning: A big data—AI integration perspective," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 4, pp. 1328–1347, Apr. 2021.

[4] P. Larrañaga, D. Atienza, J. Diaz-Rozo, A. Ogbechie, C. Puerto-Santana, and C. Bielza, *Industrial Applications of Machine Learning*. Boca Raton, FL, USA: CRC Press, 2018.

[5] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Comput. Surv.*, vol. 52, no. 1, pp. 1–38, Feb. 2019.

[6] A. I. Kadhim, "Survey on supervised machine learning techniques for automatic text classification," *Artif. Intell. Rev.*, vol. 52, no. 1, pp. 273–292, Jun. 2019.

[7] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina, N. Jaitly, B. Li, J. Chorowski, and M. Bacchiani, "State-of-the-art speech recognition with sequence-to-sequence models," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 4774–4778.

[8] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019.

[9] T. Hanika, M. Herde, J. Kuhn, J. M. Leimeister, P. Lukowicz, S. Oeste-Reiß, A. Schmidt, B. Sick, G. Stumme, S. Tomforde, and K. A. Zweig, "Collaborative interactive learning—A clarification of terms and a differentiation from other research fields," 2019, *arXiv:1905.07264*.

[10] J. Du and C. X. Ling, "Active learning with human-like noisy oracle," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2010, pp. 797–802.

[11] B. C. Wallace, K. Small, C. E. Brodley, and T. A. Trikalinos, "Who should label what? Instance allocation in multiple expert active learning," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2011, pp. 176–187.

[12] L. Zhao, Y. Zhang, and G. Sukthankar, "An active learning approach for jointly estimating worker performance and annotation reliability with crowdsourced data," 2014, *arXiv:1401.3836*.

[13] O. Dekel, C. Gentile, and K. Sridharan, "Selective sampling and active learning from single and multiple teachers," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 2655–2697, 2016.

[14] Y. Zheng, S. Scott, and K. Deng, "Active learning from multiple noisy labelers with varied costs," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2010, pp. 639–648.

[15] E. Estellés-Arolas and F. Gonzalez-Ladron-De-Guevara, "Towards an integrated crowdsourcing definition," *J. Inf. Sci.*, vol. 38, no. 2, pp. 189–200, Apr. 2012.

[16] R. Munro, *Human-in-the-Loop Machine Learning: Active Learning, Annotation, and Human-Computer Interaction*. Shelter Island, NY, USA: Manning Publications, 2019.

[17] A. Holzinger, "Interactive machine learning (iML)," *Informatik-Spektrum*, vol. 39, no. 1, pp. 64–68, Feb. 2016.

[18] S. Teso and O. Hinz, "Challenges in interactive machine learning," *KI Künstliche Intelligenz*, vol. 34, no. 2, pp. 127–130, Jun. 2020.

[19] B. Settles, "Active learning literature survey," Univ. Wisconsin–Madison, Comput. Sci., Madison, WI, USA, Tech. Rep. 1648, 2010.

[20] C. C. Aggarwal, X. Kong, Q. Gu, J. Han, and P. S. Yu, "Active learning: A survey," in *Data Classification: Algorithms and Applications*. London, U.K.: Chapman & Hall, 2014, pp. 571–605.

[21] N. Nissim, R. Moskovitch, L. Rokach, and Y. Elovici, "Novel active learning methods for enhanced PC malware detection in windows OS," *Expert Syst. Appl.*, vol. 41, no. 13, pp. 5843–5857, 2014.

[22] L. Ahmed, K. Ahmad, N. Said, B. Qolomany, J. Qadir, and A. Al-Fuqaha, "Active learning based federated learning for waste and natural disaster image classification," *IEEE Access*, vol. 8, pp. 208518–208531, 2020.

[23] S. C. H. Hoi, R. Jin, J. Zhu, and M. R. Lyu, "Batch mode active learning and its application to medical image classification," in *Proc. 23rd Int. Conf. Mach. Learn. (ICML)*, 2006, pp. 417–424.

[24] M. Herde, D. Kottke, A. Calma, M. Bieshaar, S. Deist, and B. Sick, "Active sorting—An efficient training of a sorting robot with active learning techniques," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–8.

[25] B. Settles, "From theories to queries: Active learning in practice," in *Proc. Workshop Active Learn. Exp. Design*, Sardinia, Italy, 2011, pp. 1–18.

[26] J. Howe, "The rise of crowdsourcing," *Wired Mag.*, vol. 14, no. 6, pp. 1–5, Jun. 2006.

[27] G. Paolacci, J. Chandler, and P. G. Ipeirotis, "Running experiments on Amazon mechanical Turk," *Judgment Decis. Making*, vol. 5, no. 5, pp. 411–419, 2010.

[28] M. Buhrmester, T. Kwang, and S. D. Gosling, "Amazon's mechanical Turk," *Perspect. Psychol. Sci.*, vol. 6, no. 1, pp. 3–5, 2011.

[29] L. Litman, J. Robinson, and T. Abberbock, "TurkPrime.Com: A versatile crowdsourcing data acquisition platform for the behavioral sciences," *Behav. Res. Methods*, vol. 49, no. 2, pp. 433–442, Apr. 2017.

[30] E. Peer, L. Brandimarte, S. Samat, and A. Acquisti, "Beyond the Turk: Alternative platforms for crowdsourcing behavioral research," *J. Experim. Social Psychol.*, vol. 70, pp. 153–163, May 2017.

[31] E. G. Rodrigo, J. A. Aledo, and J. A. Gámez, "Machine learning from crowds: A systematic review of its applications," *Wiley Interdiscipl. Rev., Data Mining Knowl. Discovery*, vol. 9, no. 2, Art. no. e1288, 2019, Art. no. e1288.

[32] X. Zhu and X. Wu, "Class noise vs. attribute noise: A quantitative study," *Artif. Intell. Rev.*, vol. 22, no. 3, pp. 177–210, Nov. 2004.

[33] J. A. Sáez, M. Galar, J. Luengo, and F. Herrera, "Analyzing the presence of noise in multi-class problems: Alleviating its influence with the one-vs-one decomposition," *Knowl. Inf. Syst.*, vol. 38, no. 1, pp. 179–206, 2014.

[34] S. Arora, E. Nyberg, and C. P. Rosé, "Estimating annotation cost for active learning in a multi-annotator environment," in *Proc. NAACL HLT Workshop Act. Learn. Natural Lang. Process. (HLT)*, 2009, pp. 18–26.

[35] D. Angluin, "Queries and concept learning," *Mach. Learn.*, vol. 2, no. 4, pp. 319–342, 1988.

[36] A. Calma, J. M. Leimeister, P. Lukowicz, S. Oeste-Reiss, T. Reitmaier, A. Schmidt, B. Sick, G. Stumme, and K. A. Zweig, "From active learning to dedicated collaborative interactive learning," in *Proc. Int. Conf. Archit. Comput. Syst.*, Nuremberg, Germany, 2016, pp. 1–8.

[37] G. Bahle, A. Calma, J. M. Leimeister, P. Lukowicz, S. Oeste-Reiss, T. Reitmaier, A. Schmidt, B. Sick, G. Stumme, and K. A. Zweig, "Lifelong learning and collaboration of smart technical systems in open-ended environments—Opportunistic collaborative interactive learning," in *Proc. IEEE Int. Conf. Autonomic Comput. (ICAC)*, Jul. 2016, pp. 1–10.

[38] P. Donmez and J. G. Carbonell, "Proactive learning: Cost-sensitive active learning with multiple imperfect oracles," in *Proc. 17th ACM Conf. Inf. Knowl. Mining (CIKM)*, 2008, pp. 619–628.

[39] P. Donmez and J. G. Carbonell, *From Active to Proactive Learning Methods*. Berlin, Germany: Springer, 2010, pp. 97–120.

[40] S. Breker, A. Claudi, and B. Sick, "Capacity of low-voltage grids for distributed generation: Classification by means of stochastic simulations," *IEEE Trans. Power Syst.*, vol. 30, no. 2, pp. 689–700, Mar. 2015.

[41] S. Breker, J. Rentmeister, B. Sick, and M. Braun, "Hosting capacity of low-voltage grids for distributed generation: Classification by means of machine learning techniques," *Appl. Soft Comput.*, vol. 70, pp. 195–207, Sep. 2018.

[42] H. B. Puttgen, P. R. MacGregor, and F. C. Lambert, "Distributed generation: Semantic hype or the dawn of a new era?" *IEEE Power Energy Mag.*, vol. 1, no. 1, pp. 22–29, Jan. 2003.

[43] J. Song, H. Wang, Y. Gao, and B. An, "Active learning with confidence-based answers for crowdsourcing labeling tasks," *Knowl.-Based Syst.*, vol. 159, pp. 244–258, Nov. 2018.

[44] A. Calma, M. Stolz, D. Kottke, S. Tomforde, and B. Sick, "Active learning with realistic data—A case study," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–8.

[45] J. Zhong, K. Tang, and Z.-H. Zhou, "Active learning from crowds with unsure option," in *Proc. Int. Conf. Artif. Intell.*, Buenos Aires, Argentina, 2015, pp. 1061–1067.

[46] Y. Fu, B. Li, X. Zhu, and C. Zhang, "Do they belong to the same class: Active learning by querying pairwise label homogeneity," in *Proc. 20th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2011, pp. 2161–2164.

[47] D. M. Kane, S. Lovett, S. Moran, and J. Zhang, "Active classification with comparison queries," in *Proc. IEEE 58th Annu. Symp. Found. Comput. Sci. (FOCS)*, Oct. 2017, pp. 355–366.

[48] P. Donmez, J. Carbonell, and J. Schneider, "A probabilistic framework to learn from multiple annotators with time-varying accuracy," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2010, pp. 826–837.

[49] A. K. Jain, J. Mao, and K. M. Mohiuddin, "Artificial neural networks: A tutorial," *Computer*, vol. 29, no. 3, pp. 31–44, Mar. 1996.

[50] A. Kapoor, E. Horvitz, and S. Basu, "Selective supervision: Guiding supervised learning with decision-theoretic active learning," in *Proc. Int. Joint Conf. Artif. Intell.*, Hyderabad, India, 2007, pp. 877–882.

[51] J. Zhu, H. Wang, E. Hovy, and M. Ma, "Confidence-based stopping criteria for active learning for data annotation," *ACM Trans. Speech Lang. Process.*, vol. 6, no. 3, pp. 1–24, Apr. 2010.

[52] M. Altschuler and M. Bloodgood, "Stopping active learning based on predicted change of f measure for text classification," in *Proc. IEEE 13th Int. Conf. Semantic Comput. (ICSC)*, Jan. 2019, pp. 47–54.

[53] K. Scharei, M. Herde, M. Bieshaar, A. Calma, D. Kottke, and B. Sick, "Automated active learning with a robot," *Arch. Data Sci., Ser. A*, vol. 5, no. 1, p. 16, 2018.

[54] Y. Fu, X. Zhu, and B. Li, "A survey on instance selection for active learning," *Knowl. Inf. Syst.*, vol. 35, no. 2, pp. 249–283, 2013.

[55] D. D. Lewis and C. Catlett, "Heterogeneous uncertainty sampling for supervised learning," in *Proc. Int. Conf. Mach. Learn.*, San Francisco, CA, USA, 1994, pp. 148–156.

[56] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *J. Mach. Learn. Res.*, vol. 2, pp. 45–66, Mar. 2001.

[57] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, Jul./Oct. 1948.

[58] N. Roy and A. McCallum, "Toward optimal active learning through sampling estimation of error reduction," in *Proc. Int. Conf. Mach. Learn.*, San Francisco, CA, USA, 2001, pp. 441–448.

[59] T. Osugi, D. Kun, and S. Scott, "Balancing exploration and exploitation: A new algorithm for active machine learning," in *Proc. 5th IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2005, pp. 1–8.

[60] P. Donmez, J. G. Carbonell, and P. N. Bennett, "Dual strategy active learning," in *Proc. Eur. Conf. Mach. Learn.*, Warsaw, Poland, 2007, pp. 116–127.

[61] T. Reitmaier and B. Sick, "Let us know your decision: Pool-based active training of a generative classifier with the selection strategy 4DS," *Inf. Sci.*, vol. 230, pp. 106–131, May 2013.

[62] A. Calma, T. Reitmaier, and B. Sick, "Semi-supervised active learning for support vector machines: A novel approach that exploits structure information in data," *Inf. Sci.*, vol. 456, pp. 13–33, Aug. 2018.

[63] D. Kottke, M. Herde, C. Sandrock, D. Huseljic, G. Krempl, and B. Sick, "Toward optimal probabilistic active learning using a Bayesian approach," *Mach. Learn.*, vol. 110, pp. 1199–1231, May 2021.

[64] P. Kumar and A. Gupta, "Active learning query strategies for classification, regression, and clustering: A survey," *J. Comput. Sci. Technol.*, vol. 35, no. 4, pp. 913–945, Jul. 2020.

[65] P. G. Ipeirotis, F. Provost, V. S. Sheng, and J. Wang, "Repeated labeling using multiple noisy labelers," *Data Mining Knowl. Discovery*, vol. 28, no. 2, pp. 402–441, 2014.

[66] C. H. Lin, M. Mausam, and D. S. Weld, "Re-active learning: Active learning with relabeling," in *Proc. AAAI Conf. Artif. Intell.*, Phoenix, AZ, USA, 2016, pp. 1845–1852.

[67] X.-Y. Zhang, S. Wang, and X. Yun, "Bidirectional active learning: A two-way exploration into unlabeled and labeled data set," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 12, pp. 3034–3044, Dec. 2015.

[68] Y. Yan, R. Rosales, G. Fung, F. Farooq, B. Rao, and J. Dy, "Active learning from multiple knowledge sources," in *Proc. Int. Conf. Artif. Intell. Statist.*, La Palma, Canary Islands, 2012, pp. 1350–1357.

[69] M. Herde, D. Kottke, D. Huseljic, and B. Sick, "Multi-annotator probabilistic active learning," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 10281–10288.

[70] M. Fang, X. Zhu, B. Li, W. Ding, and X. Wu, "Self-taught active learning from crowds," in *Proc. IEEE 12th Int. Conf. Data Mining*, Dec. 2012, pp. 858–863.

[71] P. D. Turney, "Types of cost in inductive concept learning," 2002, *arXiv:cs/0212034*.

[72] N. Seliya, T. M. Khoshgoftaar, and J. Van Hulse, "A study on the relationships of classifier performance metrics," in *Proc. Int. Conf. Tools With Artif. Intell.*, 2009, pp. 59–66.

[73] Z.-H. Zhou and X.-Y. Liu, "On multi-class cost-sensitive learning," *Comput. Intell.*, vol. 26, no. 3, pp. 232–257, 2010.

[74] C. Elkan, "The foundations of cost-sensitive learning," in *Proc. Int. Joint Conf. Artif. Intell.*, Seattle, WA, USA, 2001, pp. 973–978.

[75] S. Baccianella, A. Esuli, and F. Sebastiani, "Evaluation measures for ordinal regression," in *Proc. 9th Int. Conf. Intell. Syst. Design Appl.*, 2009, pp. 283–287.

[76] C. M. Bishop, *Pattern Recognition and Machine Learning*. Cham, Switzerland: Springer, 2006.

[77] P. K. Chan, W. Fan, A. L. Prodromidis, and S. J. Stolfo, "Distributed data mining in credit card fraud detection," *IEEE Intell. Syst. Appl.*, vol. 14, no. 6, pp. 67–74, Nov./Dec. 1999.

[78] D. Kottke, J. Schellinger, D. Huseljic, and B. Sick, "Limitations of assessing active learning performance at runtime," 2019, *arXiv:1901.10338*.

[79] A. J. Joshi, F. Porikli, and N. Papanikolopoulos, "Breaking the interactive bottleneck in multi-class classification with active selection and binary feedback," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2995–3002.

[80] A. J. Joshi, F. Porikli, and N. P. Papanikolopoulos, "Scalable active learning for multiclass image classification," *IEEE Trans. Softw. Eng.*, vol. 34, no. 11, pp. 2259–2273, Nov. 2012.

[81] B. Settles, M. Craven, and L. Friedland, "Active learning with real annotation costs," in *Proc. NIPS Workshop Cost-Sensitive Learn.*, Vancouver, CA, Canada, 2008, pp. 1–10.

[82] D. D. Margineantu, "Active cost-sensitive learning," in *Proc. Int. Joint Conf. Artif. Intell.*, Edinburgh, Scotland, 2005, pp. 1622–1623.

[83] A. Liu, G. Jun, and J. Ghosh, "A self-training approach to cost sensitive uncertainty sampling," *Mach. Learn.*, vol. 76, nos. 2–3, pp. 257–270, Sep. 2009.

[84] P.-L. Chen and H.-T. Lin, "Active learning for multiclass cost-sensitive classification using probabilistic models," in *Proc. Conf. Technol. Appl. Artif. Intell.*, Dec. 2013, pp. 13–18.

[85] G. Krempl, D. Kottke, and V. Lemaire, "Optimised probabilistic active learning (OPAL)," *Mach. Learn.*, vol. 100, nos. 2–3, pp. 449–476, Sep. 2015.

[86] C. Kading, A. Freytag, E. Rodner, P. Bodesheim, and J. Denzler, "Active learning and discovery of object categories in the presence of unnameable instances," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4343–4352.

[87] A. T. Nguyen, B. C. Wallace, and M. Lease, "Combining crowd and expert labels using decision theoretic active learning," in *Proc. AAAI Conf. Hum. Comput. Crowdsourcing*, San Diego, CA, USA, 2015, pp. 120–129.

[88] K.-H. Huang and H.-T. Lin, "A novel uncertainty sampling algorithm for cost-sensitive multiclass active learning," in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, Dec. 2016, pp. 925–930.

[89] F. Min, F.-L. Liu, L.-Y. Wen, and Z.-H. Zhang, "Tri-partition cost-sensitive active learning through kNN," *Soft Comput.*, vol. 23, no. 5, pp. 1557–1572, 2019.

[90] Y.-X. Wu, X.-Y. Min, F. Min, and M. Wang, "Cost-sensitive active learning with a label uniform distribution model," *Int. J. Approx. Reasoning*, vol. 105, pp. 49–65, Feb. 2019.

[91] M. Wang, Y. Lin, F. Min, and D. Liu, "Cost-sensitive active learning through statistical methods," *Inf. Sci.*, vol. 501, pp. 460–482, Oct. 2019.

[92] A. Krishnamurthy, A. Agarwal, T.-K. Huang, H. Daumé, and J. Langford, "Active learning for cost-sensitive classification," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1915–1924.

[93] A. Krishnamurthy, H. Daum, and J. Langford, "Active learning for cost-sensitive classification," *J. Mach. Learn. Res.*, vol. 20, pp. 1–50, Jul. 2019.

[94] S. Chakraborty, "Asking the right questions to the right users: Active learning with imperfect oracles," in *Proc. AAAI Conf. Artif. Intell.*, New York, NY, USA, 2020, pp. 3365–3372.

[95] S. Moon and J. G. Carbonell, "Proactive learning with multiple class-sensitive labelers," in *Proc. Int. Conf. Data Sci. Adv. Anal. (DSAA)*, Oct. 2014, pp. 32–38.
[96] S.-J. Huang, J.-L. Chen, X. Mu, and Z.-H. Zhou, "Cost-effective active learning from diverse labelers," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 1879–1885.
[97] Y.-L. Tsou and H.-T. Lin, "Annotation cost-sensitive active learning by tree sampling," *Mach. Learn.*, vol. 108, no. 5, pp. 785–807, May 2019.
[98] R. A. Haertel, E. K. Ringger, and J. L. Carroll, "Return on investment for active learning," in *Proc. NIPS Workshop Cost Sensitive Learn.*, Vancouver, BC, Canada, 2008, pp. 1–8.
[99] K. Tomanek and U. Hahn, "A comparison of models for cost-sensitive active learning," in *Proc. Int. Conf. Comput. Linguistics*, Beijing, China, 2010, pp. 1247–1255.
[100] B. C. Wallace, K. Small, C. E. Brodley, J. Lau, and T. A. Trikalinos, "Modeling annotation time to reduce workload in comparative effectiveness reviews categories and subject descriptors active learning to mitigate workload," in *Proc. Int. Health Inform. Symp.*, 2010, pp. 28–35.
[101] E. A. Ni and C. X. Ling, "Active Learning with *c*-certainty," in *Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining*, Kuala Lumpur, Malaysia, 2012, pp. 231–242.
[102] J. Du and C. X. Ling, "Active learning with generalized queries," in *Proc. 9th IEEE Int. Conf. Data Mining*, Dec. 2009, pp. 120–128.
[103] Z. Luo and M. Hauskrecht, "Region-based active learning with hierarchical and adaptive region construction," in *Proc. SIAM Int. Conf. Data Mining*, Calgary, AB, Canada, 2019, pp. 441–449.
[104] P. Rashidi and D. J. Cook, "Ask me better questions: Active learning queries based on rule induction," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2011, pp. 904–912.
[105] Y. Fu, B. Li, X. Zhu, and C. Zhang, "Active learning without knowing individual instance labels: A pairwise label homogeneity query approach," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 4, pp. 808–822, Apr. 2014.
[106] B. Qian, X. Wang, N. Cao, H. Li, and Y.-G. Jiang, "A relative similarity based method for interactive patient risk prediction," *Data Mining Knowl. Discovery*, vol. 29, no. 4, pp. 1070–1093, Jul. 2015.
[107] C. Sandrock, M. Herde, A. Calma, D. Kottke, and B. Sick, "Combining self-reported confidences from uncertain annotators to improve label quality," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.
[108] Q. Nguyen, H. Valizadegan, and M. Hauskrecht, "Learning classification models with soft-label information," *J. Amer. Med. Inform. Assoc.*, vol. 21, no. 3, pp. 501–508, May 2014.
[109] A. Biswas and D. Parikh, "Simultaneous active learning of classifiers & attributes via relative feedback," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 644–651.
[110] P. Hu, Z. C. Lipton, A. Anandkumar, and D. Ramanan, "Active learning with partial feedback," in *Proc. Int. Conf. Represent. Learn.*, New Orleans, LA, USA, 2019, pp. 1–14.
[111] A. R. Bhattacharya and S. Chakraborty, "Active learning with n-ary queries for image recognition," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2019, pp. 800–808.
[112] N. Cebron, F. Richter, and R. Lienhart, "I can tell you what it's not': Active learning from counterexamples," *Prog. Artif. Intell.*, vol. 1, no. 4, pp. 291–301, 2012.
[113] M. Fang and X. Zhu, "Active learning with uncertain labeling knowledge," *Pattern Recognit. Lett.*, vol. 43, pp. 98–108, Jul. 2014.
[114] S. Teso and K. Kersting, "Explanatory interactive machine learning," in *Proc. AAAI/ACM Conf. AI, Ethics, Soc.*, Jan. 2019, pp. 239–245.
[115] G. Druck, B. Settles, and A. McCallum, "Active learning by labeling features," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, vol. 1, 2009, pp. 81–90.
[116] B. Settles, "Closing the loop: Fast, interactive semi-supervised annotation with queries on features and instances," in *Proc. Conf. Empirical Methods Natural Language Process.*, Edinburgh, Scotland, 2011, pp. 1467–1478.
[117] Z. Luo and M. Hauskrecht, "Hierarchical active learning with group proportion feedback," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 2532–2538.
[118] Z. Luo and M. Hauskrecht, "Hierarchical active learning with proportion feedback on regions," in *Proc. Eur. Conf. Mach. Learn.*, Dublin, Ireland, 2018, pp. 464–480.
[119] M. M. Haque, L. B. Holder, M. K. Skinner, and D. J. Cook, "Generalized query-based active learning to identify differentially methylated regions in DNA," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 10, no. 3, pp. 632–644, May 2013.
[120] S. Xiong, Y. Pei, R. Rosales, and X. Z. Fern, "Active learning from relative comparisons," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 12, pp. 3166–3175, Dec. 2015.
[121] Y. Xu, H. Zhang, K. Miller, A. Singh, and A. Dubrawski, "Noise-tolerant interactive learning using pairwise comparisons," in *Proc. Adv. Neural Inf. Process. Syst.*, Long Beach, CA, USA, 2017, pp. 2431–2440.
[122] M. Hopkins, D. Kane, S. Lovett, and G. Mahajan, "Noise-tolerant, reliable active classification with comparison queries," in *Proc. Conf. Learn. Theory*, 2020, pp. 1957–2006.
[123] P. Donmez, J. G. Carbonell, and J. Schneider, "Efficiently learning the accuracy of labeling sources for selective sampling," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2009, pp. 259–268.
[124] M. Li, A. F. Myrman, T. Mu, and S. Ananiadou, "Modelling instance-level annotator reliability for natural language labelling tasks," in *Proc. Conf. North*, 2019, pp. 2873–2883.
[125] F. Daniel, P. Kucherbaev, C. Cappiello, B. Benatallah, and M. Allahbakhsh, "Quality control in crowdsourcing: A survey of quality attributes, assessment techniques, and assurance actions," *ACM Comput. Surv.*, vol. 51, no. 1, pp. 1–40, Apr. 2018.
[126] Y. Jin, M. Carman, Y. Zhu, and Y. Xiang, "A technical survey on statistical modelling and design methods for crowdsourcing quality control," *Artif. Intell.*, vol. 287, Jun. 2020, Art. no. 103351.
[127] G. Kazai, J. Kamps, and N. Milic-Frayling, "An analysis of human factors and label accuracy in crowdsourcing relevance judgments," *Inf. Retr.*, vol. 16, no. 2, pp. 138–178, Jul. 2013.
[128] J. Whitehill, P. Ruvolo, T. Wu, J. Bergsma, and J. Movellan, "Whose vote should count more: Optimal integration of labels from labelers of unknown expertise," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, 2009, pp. 2035–2043.
[129] B. Beigman Klebanov and E. Beigman, "Some empirical evidence for annotation noise in a benchmarked dataset," in *Proc. Hum. Language Technol., Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, Los Angeles, CA, USA, 2010, pp. 438–446.
[130] A. Calma, S. Oeste-Reiß, B. Sick, and J. M. Leimeister, "Leveraging the potentials of dedicated collaborative interactive learning: Conceptual foundations to overcome uncertainty by human-machine collaboration," in *Proc. 51st Hawaii Int. Conf. Syst. Sci.*, 2018, pp. 960–968.
[131] J. Kruger and D. Dunning, "Unskilled and unaware of it: How difficulties in recognizing one's own incompetence lead to inflated self-assessments," *J. Personality Social Psychol.*, vol. 77, no. 6, pp. 1121–1134, 1999.
[132] U. Gadiraju, B. Fetahu, R. Kawase, P. Siehndel, and S. Dietze, "Using worker self-assessments for competence-based pre-selection in crowdsourcing microtasks," *ACM Trans. Comput.-Hum. Interact.*, vol. 24, no. 4, pp. 1–26, Sep. 2017.
[133] A. Calma and B. Sick, "Simulation of annotators for active learning: Uncertain oracles," in *Proc. Workshop Tutorial Interact. Adapt. Learn.*, Skopje, Macedonia, 2017, pp. 49–58.
[134] S. Amershi, M. Cakmak, W. B. Knox, and T. Kulesza, "Power to the people: The role of humans in interactive machine learning," *AI Mag.*, vol. 35, no. 4, pp. 105–120, 2014.
[135] M. Cakmak, C. Chao, and A. L. Thomaz, "Designing interactions for robot active learners," *IEEE Trans. Auto. Mental Develop.*, vol. 2, no. 2, pp. 108–118, Jun. 2010.
[136] J. C. Chang, S. Amershi, and E. Kamar, "Revolt: Collaborative crowdsourcing for labeling machine learning datasets," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, May 2017, pp. 2334–2346.
[137] B. Frenay and M. Verleysen, "Classification in the presence of label noise: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 5, pp. 845–869, May 2013.
[138] L. P. Kaelbling, *Learning in Embedded Systems*. Cambridge, MA, USA: MIT Press, 1993.
[139] C. Long, G. Hua, and A. Kapoor, "Active visual recognition with expertise estimation in crowdsourcing," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 3000–3007.
[140] C. Long, G. Hua, and A. Kapoor, "A joint Gaussian process model for active visual recognition with expertise estimation in crowdsourcing," *Int. J. Comput. Vis.*, vol. 116, no. 2, pp. 136–160, Jan. 2016.

[141] C. Long and G. Hua, "Multi-class multi-annotator active learning with robust Gaussian process for visual recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2839–2847.
[142] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer School on Machine Learning*. Berlin, Germany: Springer, 2003, pp. 63–71.
[143] W. Wu, Y. Liu, M. Guo, C. Wang, and X. Liu, "A probabilistic model of active learning with multiple noisy oracles," *Neurocomputing*, vol. 118, pp. 253–262, Oct. 2013.
[144] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy, "Learning from crowds," *J. Mach. Learn. Res.*, vol. 11, pp. 1297–1322, Apr. 2010.
[145] T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal Process. Mag.*, vol. 13, no. 6, pp. 47–60, Nov. 1996.
[146] F. Rodrigues, F. Pereira, and B. Ribeiro, "Gaussian process classification and active learning with multiple annotators," in *Proc. Int. Conf. Mach. Learn.*, Bejing, China, 2014, pp. 433–441.
[147] T. P. Minka, "Expectation propagation for approximate Bayesian inference," in *Proc. Conf. Uncertainty Artif. Intell.*, Seattle, WA, USA, 2001, pp. 362–369.
[148] R. Xu and D. C. Wunsch, "Survey of clustering algorithms," *IEEE Trans. Neural Netw.*, vol. 16, no. 3, pp. 645–678, Jun. 2005.
[149] Y. Yan, R. Rosales, G. Fung, and J. G. Dy, "Active learning from crowds," in *Proc. Int. Conf. Mach. Learn.*, Bellevue, WA, USA, 2011, pp. 1–12.
[150] Y. Yan, G. Hermosillo, R. Rosales, L. Bogoni, G. Fung, L. Moy, M. Schmidt, and J. G. Dy, "Modeling annotator expertise: Learning when everybody knows a bit of something," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2010, pp. 932–939.
[151] S. A. Dudani, "The distance-weighted k-nearest-neighbor rule," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-6, no. 4, pp. 325–327, Apr. 1976.
[152] S. J. Gershman and D. M. Blei, "A tutorial on Bayesian nonparametric models," *J. Math. Psychol.*, vol. 56, no. 1, pp. 1–12, 2012.
[153] M. Fang, J. Yin, and X. Zhu, "Knowledge transfer for multi-labeler active learning," in *Machine Learning and Knowledge Discovery in Databases*. Prague, Czech Republic: Springer, 2013, pp. 273–288. [Online]. Available: https://link.springer.com/chapter/10.1007%2F978-3-642-40988-2_18
[154] M. Fang, J. Yin, and D. Tao, "Active learning for crowdsourcing using knowledge transfer," in *Proc. AAAI Int. Conf. Artif. Intell.*, Quebec City, QC, USA, 2014, pp. 1809–1815.
[155] J. Yang, T. Drake, A. Damianou, and Y. Maarek, "Leveraging crowdsourcing data for deep active learning an application: Learning intents in Alexa," in *Proc. World Wide Web Conf. World Wide Web (WWW)*, 2018, pp. 23–32.
[156] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Hoboken, NJ, USA: Wiley, 1991, ch. Entropy, Relative Entropy and Mutual Information, pp. 12–49.
[157] B. Ghai, Q. V. Liao, Y. Zhang, R. Bellamy, and K. Mueller, "Explainable active learning (XAL): An empirical study of how local explanations impact annotator experience," 2020, *arXiv:2001.09219*.
[158] W. Liang, J. Zou, and Z. Yu, "ALICE: Active learning with contrastive natural language explanations," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2020, pp. 4380–4391.
[159] O. Sener and S. Savarese, "Active learning for convolutional neural networks: A core-set approach," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–13.
[160] D. E. Difallah, G. Demartini, and P. Cudré-Mauroux, "Pick-a-crowd: Tell me what you like, and i'll tell you what to do," in *Proc. 22nd Int. Conf. World Wide Web (WWW)*, 2013, pp. 367–374.
[161] J. Baldridge and A. Palmer, "How well does active learning actually work? Time-based evaluation of cost-reduction strategies for language documentation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Singapore, 2009, pp. 296–305.
[162] Z. Shen, J. Zhao, M. Dell, Y. Yu, and W. Li, "OLALA: Object-level active learning for efficient document layout annotation," 2020, *arXiv:2010.01762*.
[163] D. P. Papadopoulos, J. R. R. Uijlings, F. Keller, and V. Ferrari, "Extreme clicking for efficient object annotation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4940–4949.
[164] T. Lookman, P. V. Balachandran, D. Xue, and R. Yuan, "Active learning in materials science with emphasis on adaptive sampling using uncertainties for targeted design," *NPJ Comput. Mater.*, vol. 5, no. 1, pp. 1–17, Dec. 2019.
[165] M. Bohme, C. Geethal, and V.-T. Pham, "Human-in-the-loop automatic program repair," in *Proc. IEEE 13th Int. Conf. Softw. Test., Validation Verification (ICST)*, Oct. 2020, pp. 274–285.

MAREK HERDE received the B.Sc. and M.Sc. degrees in computer science from the University of Kassel, Germany, where he is currently pursuing the Ph.D. degree in computer science. His research interests include active learning, deep learning, and methods for learning from error-prone annotators.

DENIS HUSELJIC received the B.Sc. and M.Sc. degrees in computer science from the University of Kassel, Germany, where he is currently pursuing the Ph.D. degree in computer science. His research interests include active learning, deep learning for computer vision, and methods for uncertainty estimation.

BERNHARD SICK (Member, IEEE) received the Diploma, Ph.D., and Habilitation degrees from the University of Passau, Germany. He is currently a Full Professor of the Department Intelligent Embedded Systems at the University of Kassel, Germany. His research interests include data science and machine learning with applications, such as renewable energies, autonomous driving, and physics/materials science. He has authored more than 200 peer-reviewed publications in these areas. He is a member of GI. He received several theses awards, best paper awards, teaching awards, and inventor awards.

ADRIAN CALMA received the B.Sc., M.Sc., and Ph.D. degrees in computer science from the University of Kassel, Germany. He is keen on applying active learning techniques to real-world problems. His research interest includes developing active learning techniques handling error-prone annotators. He is currently fellow of the Department of Intelligent Embedded Systems at the University of Kassel, where he is working on improving precision farming methods with active learning.