Hybridization of Active Learning and Data Programming for Labeling Large Industrial Datasets

Mona Nashaat
Department of Electrical and Computer Engineering
University of Alberta
Edmonton, Canada
nashaata@ualberta.ca

Aindrila Ghosh
Department of Electrical and Computer Engineering
University of Alberta
Edmonton, Canada
aindrila@ualberta.ca

James Miller
Department of Electrical and Computer Engineering
University of Alberta
Edmonton, Canada
jimm@ualberta.ca

Shaikh Quader
IBM Canada
Toronto, Canada
shaikhq@ca.ibm.com

Chad Marston
IBM USA
USA
cmarston@us.ibm.com

Jean-Francois Puget
IBM France
France
j-f.puget@fr.ibm.com

***Abstract*— Modern machine learning (ML) models are being used heavily in business domains to build effective decision support systems. As a primary requirement, supervised ML models need large labeled datasets. However, obtaining a high volume of labeled training data is both expensive and time-consuming. Researchers have proposed several labeling approaches to avoid manual labeling efforts. Active learning (AL) and Data Programming (DP) are two state-of-the-art techniques used to label datasets. Nevertheless, both approaches have their strengths and weaknesses. For example, AL is computationally expensive to apply on large industrial datasets; and labels generated by DP are often inaccurate and difficult to interpret. To address these challenges, in this paper, we propose a novel hybrid method that integrates the scalability of DP with the user engagement and accuracy of AL. The proposed approach aims at optimizing the labeling process by applying DP to generate initial noisy training data and then use AL to query the user to label only those points that maximize the accuracy of the final labels with a minimum annotation cost. To evaluate the proposed approach, we have used five open source datasets and a real-world business dataset of 1.5 million records. We use traditional active learning and data programming techniques as baselines to compare the performance and annotation cost of our proposed approach. The results show that the proposed method can achieve higher labeling accuracy than data programming. It also can minimize the labeling cost in real-world business scenarios, while delivering a comparable level of performance (accuracy) with active learning.**

Keywords- labeling; machine learning; data programming; active learning; user engagement

I. INTRODUCTION

Organizations have been collecting and managing big industrial data about their operations and business transactions. These datasets are used in building various decision-making systems; as such big data can help businesses improve their decision-making process [1]. Recently, machine learning techniques have been used to derive high-end predictions that can lead organizations to an efficient decision-making process. However, processing and analyzing such significant industrial datasets stipulate new types of technical challenges, especially given the fact that, the quality and the size of training data is the crucial factor that delimits the performance of real-world predictive systems. Also, the recent popularity of complex machine learning techniques such as deep neural networks emphasizes the importance of having high-quality input data, as these techniques can require millions to billions of training data points to perform well. Therefore, having accurate labeled training data is an essential prerequisite for building any of these supervised machine learning models [2]. However, obtaining a large set of labeled training data is a key challenge for machine learning researchers [3, 4]. For most real-world problems, labeled training datasets do not exist. Manual labeling of large industrial datasets is an expensive task, as it is time-consuming and requires domain expertise.

As a result, recently, some approaches have been proposed that aim at generating training data with minimal manual labeling effort. We partition these approaches based on the level of automation in two groups: (i) semi-automated, and (ii) fully-automated, approaches. Semi-automated approaches include Active Learning (AL) [5] and Crowdsourcing [6]. AL tries to minimize the manual labeling effort by asking an oracle or human annotator only to label the instances which are most effective for the model to achieve accurate predictions with low annotation cost. On the other hand, fully-automated approaches do not require any manual effort as they employ programmatic procedures

978-1-5386-5035-6/18/$31.00 ©2018 IEEE

to label datasets. Data Programming (DP) [3] and Distant Supervision [7] are good examples of this group of approaches. The data programming paradigm aims at modeling heuristics, which include weak supervision sources. It uses generative models to refine imperfect training sets that can be used to train discriminative models.

However, both semi- and fully-automatic approaches have their challenges. For example, although DP can automatically label large training datasets, the labels are often noisy [8]. Moreover, debugging these generative models is a challenging task. The errors in the labeling process are hard to detect due to the minimal manual interactions. On the other hand, due to the required level of user involvement, AL can be expensive and might not always be applicable to large datasets [9]. Over the last few years, several researchers [10, 11, 12, 13] have proposed techniques to enhance the scalability of AL. Most of these techniques [10, 11, 12] focus on reducing the number of queries to the oracle. However, research [9, 10] shows that the existing techniques face a continuous tradeoff between accuracy, scalability, and annotation cost. And given the fact that in real-life – Business-to-Business (B2B) scenarios [14] – datasets consist of millions, or even billions, of training instances, this trade-off is often highly problematic. **This trade-off between the accuracy and scalability of AL for real-world B2B datasets with millions of records remains an open research question**.

In this paper, we present a new approach, in which we employ AL within the DP process. It aims at injecting domain expertise in the labeling phase without overwhelming the user to label unnecessary data-points. The proposed method aims at improving the accuracy of the generated noisy training data. It also optimizes user engagement with the labeling process along with the annotation cost, by querying the oracle to only label points, which the generative model has difficulty in labeling. It benefits from the scalability of data programming while bringing the domain experts back in the loop using active learning. Moreover, the integration of the domain experts in the data programming pipeline not only helps the experts to understand the level of accuracy of the labels, but it also makes it easier for users to trust the predictions of the final model.

An overall illustration of the proposed approach is shown in Figure 1. As the figure shows, the method starts by applying a set of labeling functions on the unlabeled dataset to generate an initial noisy version of the labels. Afterward, the data points with the noisy labels are examined to construct an unlabeled pool of data to label them using AL. Then, the user is queried about uncertain points; these queried points are then used to create a refined noisy matrix which is more accurate than the initial matrix. This refined set is then used by a generative model to output a set of probabilistic labels. This set is used to train a discriminative model to generate predictions for the desired learning problem.

In order to evaluate the proposed approach, we compare its performance with the performances of active learning and data programming, on the same datasets. In the experiments,

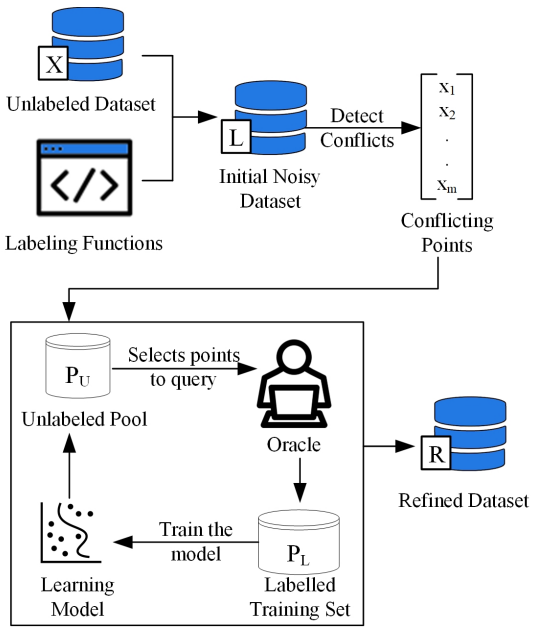

Figure 1. Overview of The Proposed Approach.

we use a real-world large-scale dataset of 1.5 million records provided by our industrial partner IBM, and five other publicly available datasets to investigate the effectiveness of the proposed approach in solving the labeling problem.

This paper is structured as follows: Section 2 presents the methodology of the proposed method, while the experimental results are presented in Section 3. Section 4 discusses related work, while Section 5 concludes the paper.

II. METHODOLOGY

In this section, first, we introduce the background that is related to this research. Following that, we discuss the details of the proposed method. We divide this section into three subsections. In the first two subsections, we discuss active learning and data programming. We primarily focus on the issues of applying these techniques to real-world business situations and explain the motivation behind developing the hybrid method to improve the performance of the labeling process. In the third, subsection, we present the proposed method.

A. Active Learning

Active learning is a special type of semi-supervised learning. AL engages the users in the loop by asking them to label information to enhance the training performance of an underlying classifier [15]. In this research, we focus on pool-based AL in which the process starts with a small set of labeled data (the seed) and a pool of unlabeled data points. Then, the algorithm iteratively queries the user (the oracle) to label specific points from the unlabeled pool which are expected to enhance the performance of the underlying classifier. The queried points are then moved to the labeled pool and used to train the classifier. The process finishes by

either reaching a target performance criterion or exceeding a predefined annotation budget. AL algorithms decide which data instances should be annotated using their query strategies. The query strategy is a critical component of the AL process, as it evaluates the effectiveness of labeling each instance. There are many general query strategies that have been developed in the past; they can be grouped into the following categories [15]: (i) uncertainty sampling, (ii) query-by-committee (QBC), (iii) expected model change, (iv) expected error reduction, (iv) variance reduction, and (v) density-weighted methods.

Uncertainty sampling [16] is one of the most common and effective query strategies; it queries the user about the samples about which the learner is most uncertain. There are many ways to measure uncertainty like least confident, smallest margin, and entropy. Query-by committee strategies [17] also query the oracle about the most uncertain samples. However, they use a committee of classifiers and query the instance about which the committee members disagree. In an expected model change [15], the learner queries the instances that would result in a better model if they were labeled. Similarly, expected error reduction strategies [18] select instances that would result in a model with less generalization error. Variance reduction strategies [15] also try to minimize the generalization error by reducing the output variance. In density-weighted methods [15], the samples are selected from dense areas of the input space.

Even though AL can result in accurate predictive models with minimum labeling effort, research [10, 11, 12, 13, 19] has shown that active learning suffers from challenges of its own. Firstly, in pool-based active learning [10, 11, 12], in order to query the oracle with the most uncertain data-points, AL needs to rank the unlabeled data points in every iteration. Therefore, when the amount of unlabeled data is large, AL gets very expensive, especially with high dimensional data [10, 11]. Secondly, AL works with a small number of annotated instances in order to label a much larger amount of unlabeled data. Therefore, this class-imbalance between the labeled and unlabeled datasets can impact the performance of AL, and result in significant time consumption, especially with large datasets [10, 13]. Thirdly, researchers [19, 20] have argued that the classification of labels with AL often does not scale up to a large number of classes. Finally, AL query strategies tend only to consider data points about which the classifier is most uncertain. However, these data points usually present a small portion of the unlabeled pool, while the majority of the data consists of points with high confidence scores. Therefore, considering high confidence data points could help in reducing the number of the data points to be ranked, and hence, increase the scalability of the process [11].

There has been much research that aims at resolving the issues mentioned above of AL. For example, to optimize the selection efforts of the most uncertain samples, and improve the scalability of the AL process, Huang et al. [10] have organized the unlabeled data into a multi-layer data pool. Then, using dimensionality reduction techniques, data points with the least confidence were selected from the top layers of the pool. In order to make a better ranking, Fang et al. [12] have reframed the AL process into a reinforcement learning problem to decide whether or not to select a data-point for querying the oracle. Among other research that aims at reducing the annotation effort of AL, Shuyang et al. [13] have applied K-medoids clustering to the unlabeled segments, and only present these clusters to the oracle for labeling. Alongside, Jain et al. [19] have used a probabilistic variant of the K-Nearest Neighbor method to improve the scalability of AL for multi-class problems and increase the model discriminating capabilities. Wang et al. [11] on the other hand, have divided the unlabeled data into low-confidence and high-confidence samples, where the high-confidence samples were assigned with pseudo-labels with no human interactions and only a portion of the low-confident samples were presented to the oracle.

Although the abovementioned research attempts to enhance the scalability of AL and reduce the annotation cost, none of these works has been evaluated using massive high dimensional datasets with millions of records. For instance, Wang et al. [11] have reduced the annotation cost up to 15% with AL on 30,607 samples, Fang et al. [12] could reduce up to 30% of the annotation costs on a training set of 200 sentences. Similarly, Shuyang et al. [13] could reduce the labeling cost up to 50% but on a dataset with 8,732 records. In short, among all the existing techniques [10, 11, 12, 13, 19, 21] that aim at reducing the annotation cost of AL, a maximum of 60,000 records [11] was used in evaluating these techniques. Moreover, some of these approaches [11, 13, 19] are only applicable to specific domains. For example, the methods proposed by Wang et al. [11] and Jain et al. [19] are only applicable to image classification. Also, the offered technique by Shuyang et al. [13] is only suitable for sound event classification. Moreover, some methods [23] suffer from the high computational cost for high-dimensional feature space or a large number of unlabeled samples [24]. In the approach proposed by Fang et al. [12], reframing the AL process into a reinforcement learning problem requires feeding the learner with many pairs of query sequences and the resulting end-rewards. Hence, according to Liu et al. [22], with a large number of records in the unlabeled dataset, the approach [12] will need a large number of training episodes.

Therefore, we think that the question of these approaches performing with equal efficiency on datasets with millions of records in the business domain still represents an unanswered research question, which derives the primary motivation and research questions of this work.

RQ1: Do AL approaches scale to industrial-sized datasets?

B. Data Programming

Data programming [3] is a paradigm that is used to generate labeled training datasets automatically. The approach starts by allowing the end users to express weak supervision strategies or domain heuristics in the form of labeling functions. Labeling functions are small scripts that are written by domain experts to automatically label subsets

of the data. Since, these functions are created from sources such as heuristics, rules-of-thumb, ontologies, etc., they conflict, and the generated set of labels are often incorrect. Later in the process, a generative model is used to iteratively refine these labels along with the labeling functions. The final set of labels is used to train a noise- aware discriminative model in order to produce the final predictions for the learning problem.

DP has been evaluated [3] using four different datasets with a maximum size of 2.03 million records and a minimum of 129,000 records. The evaluation shows that although DP can handle dataset with 2 million records, the precision and recall values vary massively when applied on large-scale datasets. That is, while the precision of the generated labels for 256,000 records is about 84%, for a different dataset with 2.03 million records, it is only 50% [3]. Also, the recall values vary from 43.4% for 256,000 records to 29.2% in the case of the dataset with 2.03 million records. This could be traced back to the noise level in the generated labels, as with more noise in large training datasets, the model is expected to behave poorly when evaluated on test data.

Moreover, DP being a completely automatic process, it does not allow the end-user to evaluate and understand the level of noise in the output of the labeling functions. Weak-supervision [3, 8] that is used to generate the initial labels in DP is a technique that scales for huge datasets, as it enables cheaper sources for creating labeled data. However, it produces noisy labels. Therefore, the generative model needs gold-labels (i.e., hand labeled data) to refine the noise in the generated noisy labels. However, if the amount of noise in the dataset is too high, the number of ground truth (gold) labels required to achieve the desired refinement will increase [25]; obtaining additional hand-labeled data can be both expensive and time-consuming.

Additionally, the generative models are often not as accurate as discriminative models [26], as they use multiple labeling functions, each with a different level of accuracy for the same dataset. This can result in the generative model being mis-specified. In order to address the mis-specification of the generative models, researchers have proposed Socratic learning [26] which is a technique that iteratively debugs the labels that are used to train the generative model. Socratic learning identifies the crucial features in the dataset that were overlooked by the initial labeling functions and automatically debugs the labels. However, being a fully automated process, Socratic learning outputs can be hard for domain experts to interpret and understand why the generative models are failing. In order to increase the interpretability of the Socratic learning process, Verma et al. [27] have proposed the Flipper framework that allows end-users to understand the decisions made by Socratic learning and the rationale behind these decisions. Although the Flipper framework enhances the user-engagement with the data-programming process, and Socratic learning enhances the debugging of noise in the labeled dataset, Socratic learning uses the disagreement between the generative and discriminative models to refine the noise in the generated training data. It does not utilize any ground truth data or domain expertise in the process. **Moreover, the scalability of these techniques is yet to be tested in real-world situations.**

RQ2: Do current DP approaches scale to industrial-sized datasets while achieving a satisfactory level of performance?

C. Proposed Method

In order to address and resolve the shortcomings of applying active learning and data programming to real-world situations with large-scale, high dimensional datasets, we propose a hybrid method, where we employ active learning within the data programming pipeline.

RQ3: Assuming RQ1 and RQ2 are negative; can a hybrid method be constructed to resolve the scalability question?

Assuming we have a population of inputs X of size D (x_1, x_2, …x_i, …x_d) with unknown true labels Y of size D (y_1, y_2, …y_i, …y_d), where $y_i \in \{-1,1\}$. Data programming permits the users to write labeling functions, which allow them to express weak supervision sources such as patterns and heuristics. With a group of K labeling functions denoted as (Λ_1, Λ_2, …Λ_k), applying each labeling function Λ_j on each data point x_i creates a weakly supervised label for x_i. Therefore, each labeling function can result in Λ_j: X→{-1, undefined, 1}, where undefined denotes labels that could not be assigned by the labeling function Λ_j. The result of applying all the labeling functions to the unlabeled data points is a noisy label matrix L where:

$$L_{i,j} = \Lambda_j(x_i) \text{ where } 1 \le i \le D \text{ and } 1 \le j \le K \quad (1)$$

Data programming [28] constructs generative models as a factor graph. It encodes the generative model G using three factors, which are labeling propensity, labeling accuracy and labeling functions pairwise correlations, these factors can be formally defined as:

$$\begin{aligned} \emptyset^{lab}{}_{i,j}(\Lambda, Y) &= \mathbf{1}\{\Lambda_{i,j} \neq undefined\} \\ \emptyset^{Acc}{}_{i,j}(\Lambda, Y) &= \mathbf{1}\{\Lambda_{i,j} = y_i\} \\ \emptyset^{Corr}{}_{i,j,k}(\Lambda, Y) &= \mathbf{1}\{\Lambda_{i,j} = \Lambda_{i,k}\} \text{ where } j,\ k \in C \end{aligned} \quad (2)$$

where C is a set of labeling function pairs (j, k) selected by the generative model to be modeled as correlated (dependent) [3]. Since the labeling functions are weakly generated, they do not perform accurately, they often overlap and conflict. Therefore, in the hybrid method, we try to increase the labeling accuracy factor by resolving the pairwise disagreements between the labeling functions, the pairwise disagreements can be defined as:

$$\emptyset^{dis}{}_{i,j,k}(\Lambda, Y) = \mathbf{1}\{\Lambda_{i,j} \neq \Lambda_{i,k}\} \quad \text{where } j,\ k \in C \quad (3)$$

Then, the hybrid method creates a set of unlabeled points P_U (x_1, x_2, …x_i, …x_u) of size U, where $P_U \subseteq X$, $\forall x_i \in P_U$ $\{x_i | \emptyset^{dis}{}_{i,j,k}(\Lambda, Y) = 1\}$.

As the generative model does not have access to the true labels Y, it learns from the agreements and disagreements of the labeling functions. In the hybrid method, we try to inject domain expertise at this point by asking the user to refine the

disagreements between the labeling functions using AL. Therefore, AL selects points from the pool P_U after estimating their informativeness and queries the users for their labels. There are query strategies that can be employed including uncertainty sampling, QBC and random sampling. The hybrid method applies uncertainty sampling as the default query strategy. We have selected uncertainty sampling and QBC as they are the two most commonly used query strategies [29]. We chose uncertainty sampling as the default query strategy as it shows superiority over other query strategies in our experiments; see section 4.3. Uncertainty sampling only queries the instances about which the model is least confident. The strategy iteratively ranks the pool and considers data point with the least confident score.

The number of data points in P_U is much smaller than the total number of the unlabeled dataset U<<D. As a result, the ranking time is highly reduced. Additionally, to minimize the manual effort, we allow the user to set a predetermined budget of AL ($B_{Labeling}$) to limit the total number of queried instances. Hence, the AL process ends when either all instances in P_U are queried, or the maximum number of queried instances defined by $B_{Labeling}$ is reached.

AL outputs a set of pairs $(X, Y)_{AL}$ $((x_1, y_1), (x_2, y_2), \ldots (x_i, y_i), \ldots (x_n, y_n))$ where $Y_{AL} \in$ true labels set Y, and X_{AL} is the corresponding queried instances. The hybrid method then refines the noisy label matrix L using $(X, Y)_{AL}$, as:

$$L_{refined\ i,j} = \begin{cases} y_i & if\ (x_i, y_i) \in Y_{AL} \\ L_{i,j} & otherwise \end{cases} \quad where \quad j = 1, 2, ..K \quad (4)$$

The refinement process of the label matrix L tries to maximize the empirical probability of labeling functions Λ_j and Λ_k agreeing. As the empirical probability can be formally defined as:

$$\widehat{P}_{j,k} = \frac{a}{D} \quad (5)$$

where a is the number of agreements between Λ_j and Λ_k, and D is the total number of the input data points. By providing true labels for both Λ_j and Λ_k in $L_{refined}$, a is increased, and hence, the empirical probability increases.

Moreover, the hybrid method allows the user to provide values for the true labels. Therefore, it maximizes the probability that the generated label by the labeling functions is correct, which is $P(Ø^{Acc}{}_{i,j}(\Lambda, Y) = 1)$.

Then, the data programming process continues by applying the generative model $\pi_\varphi(L_{refined}, Y)$ that utilizes $L_{refined}$ to generate a set of probabilistic labels. As the generative model learns from the agreements and disagreements of the labeling functions, using the refined supervised labels can increase the learned accuracy of the generative model. The generative model can be formally defined [26] as,

$$G : \pi_\varphi\left(L_{refined},\ Y\right) = \frac{1}{Z_\varphi}\ e^{\ \varphi^T\ L_{refined}\ Y} \quad (6)$$

where $Z_Ø$ is a partition function to ensure π is a distribution, Ø presents the average accuracy of labeling functions. $Z_Ø$ is a (probability) normalizing constant and is formulated in an analogous fashion to partition functions found in statistical mechanics [30]. The formulation of G (from [26]) is in many ways a restatement of standard statistical mechanics where we seek to evaluate the microstates of particles [31]. Hence, the formulation has significant justification across a broad set of problems.

The generative model assigns a probabilistic label for each instance in the training set. Afterward, the output of the generative model can be used to train a noise-aware discriminative model for obtaining final predictions. The complete algorithm of the proposed method is shown in Algorithm 1.

The proposed method aims at maximizing the accuracy of the labeling functions, which enhances the quality of the generated noisy labels. Even though other approaches [26, 28, 32] have been proposed to denoise L, none of these techniques has explored eliciting domain expertise in this process. Therefore, active learning is applied to help in injecting domain expertise in the labeling phase while reducing the frequency of unnecessary manual labeling. It is argued that domain expertise is required to model real problems rather than merely providing generic solutions. That is, the domain expertise encodes the actual problem into the solution.

Algorithm 1 Proposed Method

Input: Unlabeled Input data set X, query strategy q, Predefined budget $B_{Labeling}$ of running Active learning, Ground truth labels G_T.

Output: Labels for the dataset X in the form of confidence score C_s.

1: Create a set of labeling functions $\Lambda = \{\Lambda_1, \Lambda_2, \ldots \Lambda_k\}$.
2: Create a sparse matrix L by applying Λ to X.
3: Create pairwise disagreements factor $Ø^{dis}(\Lambda, Y)$.
4: Initialize empty pool P_U.
5: **Repeat**
6: **If** $Ø^{dis}{}_{i,j,k}(\Lambda,Y) = 1$ **then** add x_i to P_U.
7: $i \leftarrow i+1$.
8: **Until** $i > D$.
9: Initialize an empty set of pairs $(X, Y)_{AL}$.
10: **Repeat**
11: Pick points $Q_i \in P_U$ using q.
12: Ask the user to label the queried points with y_i.
13: Remove Q_i from P_U and add (Q_i, y_i) to $(X, Y)_{AL}$
14: **Until** P_U is empty or $B_{Labeling}$ is reached.
15: Use $(X, Y)_{AL}$ to refine L into $L_{refined}$.
16: With the $L_{refined}$ and ground truth labels G_T as input, train the generative model to output a confidence score C_s for each data-point in X.
17: Train a discriminative model with C_s.
18: Evaluate the discriminative model using new test data.

III. EXPERIMENTAL RESULTS

In this section, we demonstrate the experimental results. The experiments are twofold. First, we compare the proposed method with data programming (RQ2 and RQ3).

The results are evaluated based on the performance metrics of the generative and the discriminative models. Moreover, because the datasets used in the experiments contain ground truth labels, we estimate the labeling accuracy. In the second part, we compare the proposed method with different active learning strategies (RQ1 and RQ3). The evaluation in this part is based upon an estimation of the performance metrics of the predictive model and the number of queried data points needed to reach this performance level.

This section is divided into four subsections. In the first part, we discuss the datasets which are used in the experiments. Next, we discuss the experimental settings. In the third part, we compare the proposed method with data programming. The last section compares between the proposed method and active learning.

A. Datasets

This is collaborative research with IBM Analytics research team. Hence, for our experiments, we have used a real-world dataset, "Renewal Sales," that contains highly-anonymized renewal records from IBM Corporation. The dataset approximately includes 1.3 million records. These records primarily describe historical transactions of customers renewing their software subscriptions. Due to the sensitivity and financial value of such data, we only have limited access to a completely anonymized version of the actual data. Each data point presents information of software license and subscription granted for customer sites. Beside basic anonymized customer information, transaction revenue, and site identifier, each record contains information about the renewal history of the customer, such as the number of previous renewals and cancellation (non-renewals) completed by the customer, along with the final renewal status for each transaction. The dataset is used in a binary classification problem with two classes, namely Renewal '1' and Non-Renewal '-1'.

The tabular dataset has been customized by the IBM research team for being used in a Business-to-Business (B2B) scenario [14] by an organizational renewal team. In this scenario, the dataset is used to build a model to predict whether a customer will renew their contract or not so that the renewal team can take actions based on this prediction.

As this is only a single dataset, the hybrid method is also evaluated using five publicly available datasets to investigate its effectiveness in solving the labeling problem in real-world situations. The first dataset is the Bank dataset [33], which has almost 45,000 records for marketing campaigns based on phone calls. The classification goal is to predict if the phone call would result in a successful subscription to the campaign. The second dataset is the Online News Popularity dataset [34], which is used to predict the number of shares in social networks for online articles. The third one is the Occupancy Detection dataset [35] with 20,000 records for predicting room occupancy. The fourth is the Credit Card Clients dataset [36]. The dataset has 30,000 records and is used in a classification problem to predict the default credit card payments. These datasets were selected randomly – availability sampling. Sampling applied the single restriction what all datasets with physical, chemical, biological, etc. basis were considered invalid, so to avoid datasets which may contain features based upon universal facts. All datasets utilized are considered to only include human-made artifacts.

Also, since, these are all binary classification problems; this adds a threat to validity to the evaluation. Hence, as an example of multiclass classification problems, the last dataset is the MNIST dataset, which consists of hand-written digits images with 10-classes, from '0', to '9' was added.

B. Experiments Setup

In these experiments, we compare the proposed method with data programming and different active learning strategies. Table 1 shows the experiments settings. Regarding data programming, the table shows, for each dataset, the number of created labeling functions (LF), the rate of instances with positive labels (% Pos) in case of binary classifications problems. The table also shows the number of training candidates for each problem.

Although data programming creates probabilistic training labels for any discriminative model, most existing research [3, 8, 25] has used popular deep learning models. However, as we work mainly with datasets used in the business domain, we had to investigate a wide range of classification algorithms in order to find the best algorithm for each problem. In case of the renewal sales dataset and the bank dataset, we used tree boosting algorithms implemented in XGBoost [37]. In the News dataset and MNIST datasets, we used linear Support Vector Machine (SVM) classifier [38]; while we chose logistic regression for the rest of the problems. This has the added benefit of demonstrating the resilience of the approach to the choice of the classifier.

Because labeling functions affect the final model accuracy, we tried to accommodate as many weak supervision sources as possible, while maintaining an acceptable performance level for each function. Therefore, all the labeling functions that were used in the experiments have accuracy values over than 60%. In the case of the renewal dataset, we invited six renewal representatives to guide us in writing the labeling functions. They guided us into writing four labeling functions that are based upon their domain expertise. Regarding the other datasets, we used pattern-based heuristics to create the labeling functions for each dataset [25, 27, 39, 40]; it is believed that this is in-line with best practice when no access exists to domain experts.

Table 1 also shows active learning settings for the datasets used in the experiments. For each dataset, the table shows the attributes number, the size of the initial labeled set (initial seed), and the size of both training and test sets. In the beginning, all the data points in the training set are used as the unlabeled pool. Subsequently, after each iteration, the test set is used to evaluate the model's effectiveness.

The stopping criteria in active learning are determined after examining the learning curve for the underlying classifier. We stopped the active learning process when the curve shows no significant performance improvement with additional iterations.

TABLE I. ACTIVE LEARNING AND DATA PROGRAMMING SETTINGS

Dataset	# of instances	# of attributes	Data Programming			Active Learning		
			# LFs	*% Pos.*	*# Candidates*	*Initial seed*	*Train set size*	*Test set size*
Renewal Sales	1,354,704	11	4	73.06	1,083,763	67,735	839,917	447,052
Bank	45,211	17	3	11.70	36,169	2,260	28,031	14,920
News	39,797	61	5	49.34	31,716	1,989	24,675	13,133
Credit Card	30,000	24	5	22.12	24,001	1,500	18,600	9,900
Occupancy Detection	20,560	7	3	23.10	16,448	1,028	12,747	6,785
MNIST	70,000	784	5	-	56,000	3500	43,400	23,100

Regarding the proposed method, we set a maximum labeling cost of 7% of the total training set size. This value is determined based on the renewal sales dataset; the labeling budget is set after consulting domain expertise (i.e., renewal sales representatives) and estimating the cost of manual labeling. We also used the same labeling budget for the rest of the problems, because we could not estimate the labeling cost in these cases, and the chosen maximum labeling cost is far less than the queried instances percentage required by AL in each case. We compare the proposed method with several AL strategies, including uncertainty sampling, QBC, and random sampling. For the random strategy results, we repeated each experiment ten times and computed the arithmetic mean.

C. *Hybrid Method vs. Data Programming*

Both the hybrid method and data programming have been applied to the six datasets using the same set of labeling functions. The same labeling functions are used in order to evaluate the effectiveness of the proposed method in enhancing the accuracy of these functions. Table 2 shows the predictive performance metrics of the generative and discriminative models. For the generative models, the table shows precision (P), recall (R), and F1 measure. The table also presents the same measures for the discriminative model along with Matthews correlation coefficient (MCC). MCC is considered as one of the best measures for describing the confusion matrix and the classifier performance in binary classifications problems [41, 42]; other measures, such as accuracy, can be not useful in situations when the two classes highly vary in sizes. And because most of the datasets used in the experiments have strongly unbalanced classes, we considered MCC measure when evaluating the performance of the discriminative model in each problem. The table also presents the labeling accuracy computed based on the ground truth labels. The labeling accuracy is measured as the ratio between the number of correctly labeled instances to the training set size. Fundamentally, no perfect mechanism exists for encoding a confusion matrix, and hence, we present a wide array of measures to avoid measurement bias.

The table shows that the generative model in the proposed method achieved higher precision, recall and F1 values in all the problems. As the generative model performance depends on the labeling functions performance, this empirically demonstrates the effectiveness of the proposed model in resolving the disagreement between the labeling functions. In some problems, such as Bank, MNIST, and News datasets, the proposed approach managed to improve the F1 scores by 25%, 21.5%, and 14.7% respectively. Moreover, the proposed method achieved higher recall, precision, MCC, and F1 measure for discriminative model performance in all the problems. Based on MCC, the results show that the proposed method could improve the classification performance in Renewal Sales dataset by 15.5% compared to the data programming approach. It also managed to achieve 84.1% labeling accuracy, while the labeling accuracy of data programming is 68%, hence an increase ratio of 23.7%.

The results show that the proposed method achieved better labeling accuracy than data programming in all datasets. However, the smallest increase ratio obtained by the proposed method was in the Credit Card dataset. The labeling accuracy of data programming is 31.2%, and the proposed method managed to increase it up to 34%. After investigating this case, we observed that the labeling functions accuracies were initially poor. Since, enhancing the labeling accuracy depends on the initial accuracies of the labeling functions and the maximum labeling cost allowed, in some datasets, the proposed method could not achieve a significant enhancement within the permitted labeling cost. This could be a reflection on the lack of domain knowledge we are able to encode about the credit card scenario. Overall, the proposed method increased the labeling accuracy by an average of 19% across all datasets.

D. *Hybrid Method vs. Active Learning*

six datasets; Figure 2 shows a sample of the results for both the renewal sales and the bank datasets. The curves show the relation between the percentage of queried instances and the classifier accuracy. The learning curves stabilized after querying 15% and 12% of the unlabeled pool in Renewal sales dataset and the bank dataset respectively. Since we observe that uncertainty sampling achieved the highest accuracy in all datasets, we choose uncertainty sampling as the default query strategy in the hybrid method. The performance metrics obtained from applying uncertainty sampling to all the problems are summarized in Table 3. The table shows the classifier Precision (P), Recall (R), (MCC) [41, 33], accuracy (Accuracy), and the number of queried instances required to reach the corresponding MCC level. While the proposed method managed to achieve better MCC in all the problems, it achieved the highest enhancements in Bank and Occupancy Detection datasets with an improvement ratio of 21.9% and 14.3% respectively.

TABLE II. DATA PROGRAMMING RESULTS IN TERMS OF GENERATIVE MODEL PERFORMANCE (PRECISION (P), RECALL (R), AND F1 MEASURE (F1)), THE ACCURACY OF THE GENERATED LABELS (LABELLING ACCURACY), AND DISCRIMINATIVE MODEL PERFORMANCE (PRECISION (P), RECALL (R), MCC, AND F1 MEASURE (F1))

Dataset	Proposed Method								Data Programming							
	Generative Model			*Labelling*	*Discriminative Model*				*Generative Model*			*Labelling*	*Discriminative Model*			
	P	***R***	***F1***	***Accuracy***	***P***	***R***	***MCC***	***F1***	***P***	***R***	***F1***	***Accuracy***	***P***	***R***	***MCC***	***F1***
Renewal Sales	0.921	0.875	0.897	0.841	0.887	0.902	0.898	0.894	0.863	0.738	0.796	0.680	0.860	0.754	0.778	0.803
Bank	0.873	0.815	0.843	0.767	0.869	0.841	0.848	0.855	0.655	0.694	0.674	0.626	0.839	0.712	0.740	0.770
News	0.903	0.792	0.844	0.565	0.899	0.969	0.950	0.932	0.746	0.726	0.736	0.486	0.877	0.891	0.887	0.884
Credit Card	0.848	0.765	0.805	0.340	0.891	0.877	0.880	0.884	0.815	0.691	0.748	0.312	0.852	0.648	0.689	0.736
Occupancy Detection	0.938	0.806	0.867	0.683	0.891	0.934	0.923	0.912	0.817	0.778	0.797	0.603	0.869	0.815	0.828	0.841
MNIST	0.873	0.929	0.900	0.590	0.882	0.952	0.934	0.916	0.722	0.760	0.741	0.457	0.857	0.818	0.828	0.837

The table also shows that the proposed method achieved higher precision values in all the problems, with the highest precision value of 0.98 obtained in the Renewal Sales dataset. The proposed approach managed to enhance the accuracy values in Renewal Sales and Occupancy Detection with a percentage increase of 2.1% and 3.6% respectively. Additionally, the proposed method managed to achieve better accuracy values in all the problems, except for the Credit Card problem. In Credit Card problem, the initially inaccurate labeling functions suppressed the proposed method from achieving higher accuracy value. We find this point conforms with our overall conclusion of the importance of injecting domain expertise in the process of generating training dataset, either from the early beginning (i.e., writing accurate labeling functions) or in the process of refining the noisy labels (i.e., resolving the disagreements between the labeling functions). Also, the table shows that the proposed method could significantly reduce the number of queried instances in all the problems. For example, active learning needed to query the user for 15.4% and 12.3% of the training set in Renewal Sales and the Bank dataset in order to achieve accuracy levels of 0.95 and 0.93 respectively. On the other hand, the proposed method achieved higher levels of accuracy (0.97 and 0.96 in both datasets) with only querying 7% of the training set in both cases. The table attests that the proposed method manages to reduce the labeling cost in all the use cases while achieving an acceptable accuracy level and higher values in precision, recall, F1, and MCC.

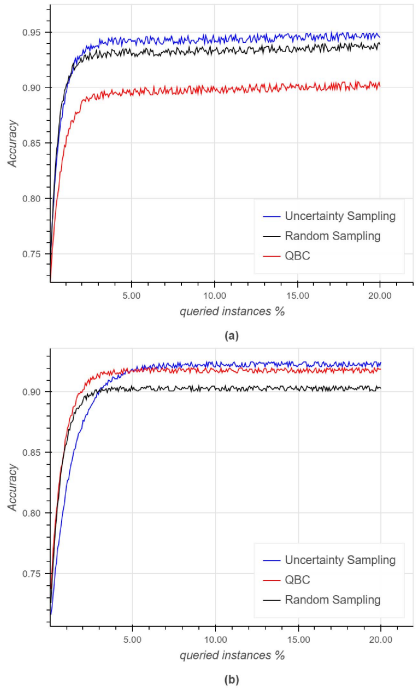

Figure 2. Active Learning Leraning Curves for (a) Renewal Sales dataset (b) Bank dataset

IV. RELATED WORK

The scalability of Machine Learning (ML) has always been a concern for researchers [43, 44, 45]. Over the years, several researchers have enhanced [46, 47] AL strategies to label large training datasets in different domains such as text analysis, image recognition, social network modeling, etc. [48]. For example, Sarawagi et al. [46] have enhanced the traditional AL technique to assist with automatic deduplication of records while integrating data from different sources. Mozafari et al. [47] have used AL to minimize the number of questions asked to the crowd, in crowd-sourced databases.

Moreover, there is ample research [49, 50] that looks into debugging the annotations generated by AL strategies. For example, Lin et al. [49] proposed a method for the automatic and cost-effective relabeling of data-points for AL techniques that use a single annotator; Donmez et al. [50] have focused on the challenge of fallible and fatigable oracles by proposing a decision-theoretic approach for selecting an optimal oracle. However, in contrast to our proposed method, Sarawagi et al. [46] only validated their approach against a maximum of 32,131 records, and have not focused on minimizing the annotation cost of AL. Similarly, the technique proposed by Mozafari et al. [47] was validated across 10,000 data-points. For Lin et al. [49] and Donmez et al. [50], the sizes of the datasets were less than even 5,000 records. Hence, the applicability of these techniques cannot be guaranteed for our dataset with 1.5 million records.

In order to achieve scalability, weak-supervision [51] is a technique that is often used by academics. However, weak-supervision not only suffers from poor performance [52] due to the noise in the generated labels, but they are difficult to interpret.

TABLE III. ACTIVE LEARNING RESULTS IN TERMS OF THE FINAL MODEL PERFORMANCE (PRECISION (P), RECALL (R), MCC, AND ACCURACY) AND THE NUMBER OF INSTANCES QUERIED TO REACH THE CORRESPONDING MCC VALUE.

Dataset	Proposed Method					Active Learning				
	P	***R***	***MCC***	***Accuracy***	***# queried instances***	***P***	***R***	***MCC***	***Accuracy***	***# queried instances***
Renewal Sales	0.98	0.98	0.90	0.97	65,100.28	0.98	0.96	0.84	0.95	139,500.60
Bank	0.78	0.88	0.80	0.96	2,172.59	0.70	0.70	0.66	0.93	3,724.44
News	0.92	0.94	0.86	0.93	1,912.40	0.90	0.91	0.80	0.90	3,005.20
Credit Card	0.73	0.83	0.72	0.90	1,441.65	0.72	0.79	0.67	0.91	3,089.25
Occupancy Detection	0.71	0.98	0.80	0.93	987.98	0.70	0.83	0.70	0.90	1,270.26
MNIST	0.93	0.98	0.90	0.95	3,363.85	0.89	0.97	0.84	0.92	4,805.50

Several researchers have attempted to enhance weak-supervision techniques [3, 8, 51] and learn from noisy data [25, 51, 52]. For example, in order to utilize the unique expertise of each expert, Guan et al. [51] propose modeling individual experts and learning sample-specific average weights by combining expert opinions. On the other hand, in order to resolve the challenges with writing appropriate labeling functions, Verma et al. [25] have proposed an automated technique that iteratively generates heuristics based on small subsets of the labeled dataset, for which the heuristic is accurate. Veit et al. [52] have proposed a technique conceptually like Data Programming that uses cleanly annotated data to refine noisy annotations in the domain of image classification. Nevertheless, as mentioned by Verma et al. [25], despite being potentially scalable, all these approaches suffer from a trade-off between accuracy and coverage. For example, while the method proposed by Veit et al. [52] could achieve a precision of 57% on a dataset with 240,449 records, the F1 measure of the model proposed by Verma et al. was 71% for a dataset of 200 records.

In order to achieve scalability with enhanced performance (accuracy), several researchers have proposed hybridization [53, 54, 55] of AL with different machine learning techniques. For example, Lin et al. [53] have combined two ML strategies namely, AL and Self-Paced Learning (SPL) in a weak expert recertification scenario, in order to enhance the classification accuracy of SPL and reduce the annotation cost of AL. On the other hand, Wang et al. [54] have combined active learning with semi-supervised learning, where the supervised clustering technique is used to discover hidden patterns in the data, AL is used to label only a portion of the unlabeled data. Finally, Park et al. [55] have proposed a method that combines active learning and adaptive incremental learning to estimate the degree of concept drift in streaming data. Nevertheless, the method offered by Park et al. [55] obtained 79% accuracy with 45,312 records; and the technique proposed by Lin et al. [53] achieved 78% accuracy with 181,901 records. Hence, the application of these methods on B2B datasets with millions of records might not yield similar, or suitable, performance. Moreover, unlike our proposed hybrid method, neither of these approaches focused on reducing the annotation cost of AL with the improved scalability and accuracy.

V. CONCLUSIONS

In this paper, we present a new hybrid method for labeling massive training datasets. The technique employs active learning (AL) within the data programming (DP) process to optimize the user engagement, annotation cost, and scalability of the labeling process. The proposed method allows the users to initially express domain knowledge through labeling functions, and then it optimizes the user engagement to resolve the conflicts between the labeling function while sustaining the annotation cost. The proposed method increases the accuracy of the generative models and the generated labels. We evaluate the proposed method by comparing its performance with traditional AL and DP techniques as our baseline approaches. The experiments used five open source datasets and one real-world business dataset of 1.5 million records provided by our industrial partner, IBM. Our empirical results show that the proposed method outperforms data programming in labeling accuracy and predictive performance in all datasets. It significantly enhances the performance of the generative models by up to 25% (F1 measure) while increasing the accuracy of the generated labels by up to 29% (labeling accuracy). When compared to active learning, the proposed method achieves higher performance in most of the problems and a comparable level of (accuracy) performance in the rest. Across all the datasets, the proposed method maintained the least labeling cost with a percentage decrease up to 53% compared to active learning!

REFERENCES

[1] N. Mohamed and J. Al-Jaroodi, "Real-time big data analytics: Applications and challenges," in 2014 International Conference on High Performance Computing & Simulation, 2014, pp. 305–310.

[2] R. Caruana, N. Karampatziakis, and A. Yessenalina, "An Empirical Evaluation of Supervised Learning in High Dimensions," Proc. the 25th International Conference on Machine Learning, USA, 2008.

[3] A. Ratner, C. De Sa, S. Wu, D. Selsam, and C. Ré, "Data Programming: Creating Large Training Sets, Quickly," 2016.

[4] Y. Sun, E. Lank, and M. Terry, "Label-and-Learn: Visualizing the Likelihood of Machine Learning Classifier's Success During Data Labeling," Proc. the 22Nd International Conference on Intelligent User Interfaces, USA, 2017, pp. 523–534.

[5] M. Elahi, F. Ricci, and N. Rubens, "A survey of active learning in collaborative filtering recommender systems," Computer Science Review, vol. 20, pp. 29–50, 2016.

[6] N. Otani, Y. Baba, and H. Kashima, "Quality Control for Crowdsourced Hierarchical Classification," Proc. IEEE ICDM, 2015.

[7] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, "Distant Supervision for Relation Extraction Without Labeled Data," Proc. ACL-IJCNLP, USA, 2009, pp. 1003–1011.

[8] A. Ratner, S. H. Bach, H. Ehrenberg, J. Fries, S. Wu, and C. Ré, "Snorkel: rapid training data creation with weak supervision," Proc. the VLDB Endowment, vol. 11, no. 3, pp. 269–282. 2017.

[9] G. V. Cormack and M. R. Grossman, “Scalability of Continuous Active Learning for Reliable High-Recall Text Classification,” Proc. the 25th ACM International on Conference on Information and Knowledge Management, USA, 2016, pp. 1039–1048.

[10] E.-C. Huang, H.-K. Pao, and Y.-J. Lee, “Big active learning,” in 2017 IEEE International Conference on Big Data, 2017, pp. 94–101.

[11] K. Wang, D. Zhang, Y. Li, R. Zhang, and L. Lin, “Cost-Effective Active Learning for Deep Image Classification,” IEEE Transactions on Circuits and Systems for Video Technology, 2017.

[12] M. Fang, Y. Li, and T. Cohn, “Learning how to Active Learn: A Deep Reinforcement Learning Approach,” Proc. the 2017 Conference on Empirical Methods in Natural Language Processing, 2017.

[13] Z. Shuyang, T. Heittola, and T. Virtanen, “Active learning for sound event classification by clustering unlabeled data,” in 2017 IEEE International Conference on Acoustics, Speech and Signal Processing, 2017, pp. 751–755.

[14] M. Bohanec, M. K. Borštnar, and M. Robnik-Šikonja, “Integration of Machine Learning Insights into Organizational Learning: A Case of B2B Sales Forecasting,” in Blurring the Boundaries Through Digital Innovation, vol. 19, 2016, pp. 71–85.

[15] B. Settles, “Active Learning Literature Survey,” University of Wisconsin–Madison, Computer Sciences Technical Report, 2009.

[16] D. D. Lewis and W. A. Gale, “A Sequential Algorithm for Training Text Classifiers,” Proc. ACM SIGIR, USA, 1994, pp. 3–12.

[17] H. S. Seung, M. Opper, and H. Sompolinsky, “Query by committee,” Proc. the fifth annual workshop on Computational learning theory, United States, 1992, pp. 287–294.

[18] N. Roy and A. McCallum, “Toward Optimal Active Learning through Sampling Estimation of Error Reduction,” in ICML, 2001.

[19] P. Jain and A. Kapoor, “Active learning for large multi-class problems,” in 2009 IEEE CVPR, 2009, pp. 762–769.

[20] S. Ertekin, J. Huang, L. Bottou, and L. Giles, “Learning on the border: active learning in imbalanced data classification,” Proc. the 16th ACM CIKM, 2007.

[21] C. Dima, M. Hebert, and A. Stentz, “Enabling learning from large datasets: applying active learning to mobile robotics,” Proc. IEEE International Conference on Robotics and Automation, USA, 2004.

[22] M. Liu, W. Buntine, and G. Haffari, “Learning How to Actively Learn: A Deep Imitation Learning Approach,” Proc. the 56th Annual Meeting of the Association for Computational Linguistics, 2018.

[23] B. Long, J. Bian, O. Chapelle, Y. Zhang, Y. Inagaki, and Y. Chang, “Active Learning for Ranking through Expected Loss Optimization,” IEEE Transactions on Knowledge and Data Engineering, 2015.

[24] Y. Fu, X. Zhu, and B. Li, “A survey on instance selection for active learning,” Knowledge and Information Systems, pp. 249–283, 2013.

[25] P. Varma and C. Ré, “Reef: Automating Weak Supervision to Label Training Data,” 2018.

[26] P. Varma et al., “Socratic Learning: Augmenting Generative Models to Incorporate Latent Subsets in Training Data,” arXiv:1610.08123.

[27] P. Varma, D. Iter, C. De Sa, and C. Ré, “Flipper: A Systematic Approach to Debugging Training Sets,” Proc. the 2nd Workshop on Human-In-the-Loop Data Analytics, USA, 2017.

[28] S. H. Bach, B. He, A. Ratner, and C. Ré, “Learning the Structure of Generative Models without Labeled Data,” arXiv:1703.00854, 2017.

[29] M. E. Ramirez-Loaiza, M. Sharma, G. Kumar, and M. Bilgic, “Active learning: an empirical study of common baselines,” Data Mining and Knowledge Discovery, vol. 31, no. 2, pp. 287–313, 2017.

[30] J. Klauder and B. Skagerstam, “Coherent States: Applications in Physics and Mathematical Physics,” World Scientific, 1984.

[31] R. J. Baxter, “Exactly solved models in statistical mechanics,” Courier Corporation, 2007.

[32] S. Wu et al., “Fonduer: Knowledge Base Construction from Richly Formatted Data,” Proc. the 2018 International Conference on Management of Data, USA, 2018, pp. 1301–1316.

[33] S. Moro, P. Cortez, and P. Rita, “A data-driven approach to predict the success of bank telemarketing,” Decision Support Systems, 2014.

[34] K. Fernandes, P. Vinagre, and P. Cortez, “A Proactive Intelligent Decision Support System for Predicting the Popularity of Online News,” in Progress in Artificial Intelligence, vol. 9273, 2015.

[35] L. M. Candanedo and V. Feldheim, “Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models,” Energy and Buildings, vol. 112, pp. 28–39, 2016.

[36] I.-C. Yeh and C. Lien, “The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients,” Expert Systems with Applications, vol. 36, no. 2, 2009.

[37] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” Proc. the 22nd ACM SIGKDD, USA, 2016, pp. 785–794.

[38] J. Kremer, K. Steenstrup Pedersen, and C. Igel, “Active learning with support vector machines,” Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 4, no. 4, pp. 313–326, 2014.

[39] P. Varma et al., “Inferring Generative Model Structure with Static Analysis,” arXiv:1709.02477, 2017.

[40] M. Dehghani, A. Severyn, S. Rothe, and J. Kamps, “Avoiding Your Teacher’s Mistakes: Training Neural Networks with Controlled Weak Supervision,” arXiv:1711.00313, 2017.

[41] D. M. Powers, “Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation,” 2011.

[42] D. Chicco, “Ten quick tips for machine learning in computational biology,” BioData Mining, vol. 10, no. 1, 2017.

[43] J. Qiu, Q. Wu, G. Ding, Y. Xu, and S. Feng, “A survey of machine learning for big data processing,” Journal on Advances in Signal Processing, vol. 2016, no. 1, 2016.

[44] H.-F. Yu, P. Jain, P. Kar, and I. S. Dhillon, “Large-scale Multi-label Learning with Missing Labels,” Proc. the 31st International Conference on International Conference on Machine Learning, 2014.

[45] S. Das Bhattacharjee, A. Talukder, and B. V. Balantrapu, “Active learning based news veracity detection with feature weighting and deep-shallow fusion,” in 2017 IEEE Conference on Big Data, 2017.

[46] S. Sarawagi and A. Bhamidipaty, “Interactive deduplication using active learning,” Proc. the 8th ACM SIGKDD, Canada, 2002.

[47] B. Mozafari, P. Sarkar, M. Franklin, M. Jordan, and S. Madden, “Scaling up crowd-sourcing to very large datasets: a case for active learning,” Proc. the VLDB Endowment, pp. 125–136, Oct. 2014.

[48] B. Du et al., “Exploring Representativeness and Informativeness for Active Learning,” IEEE Transactions on Cybernetics, 2017.

[49] C. H. Lin, M. Mausam, and D. S. Weld, “Re-Active Learning: Active Learning with Relabeling.,” in AAAI, 2016, pp. 1845–1852.

[50] P. Donmez and J. G. Carbonell, “Proactive learning: cost-sensitive active learning with multiple imperfect oracles,” Proc. ACM CIKM, USA, 2008.

[51] M. Y. Guan, V. Gulshan, A. M. Dai, and G. E. Hinton, “Who Said What: Modeling Individual Labelers Improves Classification,” arXiv:1703.08774, 2017.

[52] A. Veit, N. Alldrin, G. Chechik, I. Krasin, A. Gupta, and S. Belongie, “Learning From Noisy Large-Scale Datasets With Minimal Supervision,” Proc. IEEE CVPR, 2017, pp. 839–847.

[53] L. Lin, K. Wang, D. Meng, W. Zuo, and L. Zhang, “Active self-paced learning for cost-effective and progressive face identification,” IEEE transactions on pattern analysis and machine intelligence, 2018.

[54] Z. Wang, B. Du, L. Zhang, L. Zhang, and X. Jia, “A Novel Semisupervised Active-Learning Algorithm for Hyperspectral Image Classification,” IEEE Transactions on Geoscience and Remote Sensing, 2017.

[55] C. H. Park and Y. Kang, “An active learning method for data streams with concept drift,” in 2016 IEEE Conference on Big Data, 2016.