POLITECNICO DI TORINO
Repository ISTITUZIONALE

Active Learning with Noisy Labelers for Improving Classification Accuracy of Connected Vehicles

(Article begins on next page)

27 July 2024

Active Learning with Noisy Labelers for Improving Classification Accuracy of Connected Vehicles

Alaa Awad Abdellatif, *Member, IEEE,* Carla Fabiana Chiasserini, *Fellow, IEEE,*
Francesco Malandrino, *Senior Member, IEEE,* Amr Mohamed, *Senior Member, IEEE,*
and Aiman Erbad, *Senior Member, IEEE*

Abstract—Machine learning has emerged as a promising paradigm for enabling connected, automated vehicles to autonomously cruise the streets and react to unexpected situations. Reacting to such situations requires accurate classification for uncommon events, which in turn depends on the selection of large, diverse, and high-quality training data. In fact, the data available at a vehicle (e.g., photos of road signs) may be affected by errors or have different levels of resolution and freshness. To tackle this challenge, we propose an active learning framework that, leveraging the information collected through onboard sensors as well as received from other vehicles, effectively deals with scarce and noisy data. Given the information received from neighboring vehicles, our solution: (i) selects which vehicles can reliably generate high-quality training data, and (ii) obtains a reliable subset of data to add to the training set by trading off between two essential features, i.e., quality and diversity. The results, obtained with different real-world datasets, demonstrate that our framework significantly outperforms state-of-the-art solutions, providing high classification accuracy with a limited bandwidth requirement for the data exchange between vehicles.

Index Terms—Data selection, labeling quality, labelers selection, connected automated vehicles, online learning.

I. INTRODUCTION

The development of automated vehicles and Intelligent Transportation Systems (ITS) has recently attracted significant interest and has become one of the main goals in the agenda of research agencies worldwide. An important component of ITS is represented by automated vehicles, which, equipped with artificial intelligence (AI) technologies as well as cameras, sensors, and lidars, can learn, identify, and handle complex situations occurring on a road [1]. Ordinarily, supervised machine learning is predicated on learning from large quantity of data representing past history, e.g., different photos of the same object. However, roads are very dynamic environments, with high variability in typical driving scenarios. Hence, vehicles may face multiple and diverse uncommon situations, such as unexpected maneuvers by neighboring vehicles or movements of pedestrians and bikers, for which limited history is available. In these cases, conventional AI/supervised learning techniques that rely on large amounts of accurately labeled data for the training, cannot provide sufficiently good results.
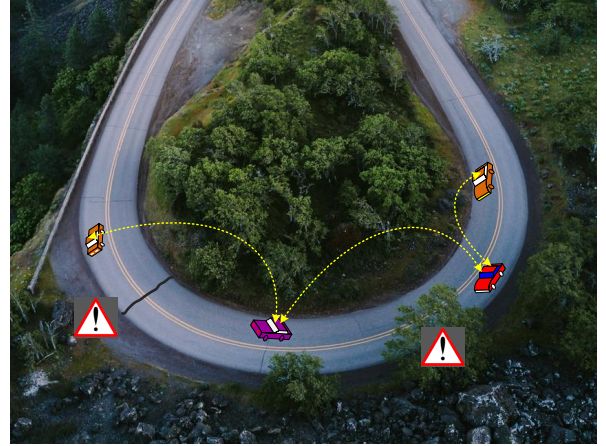
Fig. 1. V2V communications for information exchange about hazardous/unexpected situations.

To overcome such a severe problem, we propose an effective, cooperative solution exploiting the capabilities of Connected and Automated Vehicles (CAVs), based on: (i) leveraging vehicle-to-vehicle (V2V) communication to increase the amount of data available at a vehicle [2], and (ii) adopting online machine learning models to swiftly react to unexpected events. Indeed, thanks to V2V communications, vehicles can leverage not only the data generated by their own onboard sensors and the history available locally (if any), but also those received from their neighboring vehicles. This extra source of information is of particular relevance under non-line-of-sight conditions (see Figure 1), e.g., in the presence of fog or rain, in which built-in cameras may not provide enough, or accurate enough, data on the vehicle's surroundings, or on the trajectory of neighboring vehicles [3]. Additionally, V2V communication can provide a vehicle with information on an event long before the vehicle becomes exposed to it, thus allowing for more time to classify the situation and properly react.

As an example, Volvo trucks and Volvo cars manufacturers enabled their vehicles to share real-time traffic hazard information via their cloud to automatically alert each other. Thus, when a vehicle hazard warning light is switched on, an alert signal is sent to all connected cars and trucks approaching the location of the vehicle whose hazard lights have been activated [4]. Similar automated response systems are employed in platooning scenarios [5].

Our cooperative, on-line learning framework extends this example by collecting, labeling, and integrating the informa-

tion from the neighboring vehicles into the learning model of each vehicle, hence enabling fast detection/prediction of hazardous or unexpected situations.

Among the existing online machine-learning techniques, *active learning* (AL) has emerged as a promising and effective option. AL workflows are based upon actively acquiring *new data* to enhance the learned models. In other words, training data is not static, rather, an active learner can achieve high classification accuracy by selecting over time the most informative data samples and by adding such a subset of data to its training set. Active learning is based on: (i) receiving new data from the available sources – in our cases, the the sensors aboard a vehicle and the vehicle's neighbors –; (ii) selecting the most informative data to add to the training set [6], [7]; (iii) including the newly-acquired data in the training process. Depending upon the concrete scenario and underlying learning technique (e.g., neural networks, SVM...), this can mean either running additional training iterations with the new data, or restart the training from scratch. It follows that both the training set and the learning model are progressively updated, thus improving the learning quality [8]. It is worth noticing that federated learning (FL) [9], which is a popular distributed machine learning methodology allowing multiple nodes to cooperatively train a model without the need to share data, differs from AL in several ways. Indeed, FL requires tight synchronization among nodes and the presence of a centralized parameter server (also called a broker), which renders it unsuitable for highly dynamic scenarios like vehicular ones. Furthermore, AL and FL solve orthogonal problems – the former aims at using newly-arrived data, the latter aims at training a model in a distributed manner – and can be combined when warranted.

However, traditional AL schemes require accurate classifiers, generating ground-truth labels for new data as it arrives. Such an assumption is impractical in many real-time scenarios, including that of connected and automated vehicles. Indeed, vehicles are typically weak labelers, and often have at their disposal noisy data (e.g., generated by cameras in the presence of fog or rain), or data that may have different levels of resolution and freshness. Thus, not only the labels generated by the vehicles' classifiers, but also the data generated or received by a vehicle, may be affected by errors.

In this paper, we tackle the above challenges by proposing an AL framework for connected vehicles, which selects the optimal set of labelers as well as a subset of the locally-generated and received data, to be used for classification. More specifically, our main contributions are as follows:

1) *Vehicles' interaction:* we investigate three ways in which vehicles can leverage the information exchanged through V2V communications for online training, namely, by sharing labels, data, or a combination of the two. For each of these operational modes, we study the impact on the classification accuracy as well as on the network load. Importantly, although in this work we focus on V2V communications, vehicle-to-everything (V2X) communications could be leveraged as well, with the additional benefit of potentially enlarging the set of neighboring nodes with which a vehicle can interact;

2) *Characterization of data quality:* we propose a method to define the information quality, including two main steps: (i) label integration, to generate an aggregate label for the acquired data, and (ii) data quality assessment, to measure the quality of the acquired data based on labelers' accuracy, data freshness, and affinity of the corresponding labels with respect to the aggregate label;

3) *Labelers' selection:* we propose a reputation-based scheme to evaluate neighboring vehicles quality, and select those considered to be the most reliable labelers.

4) *Data selection:* we define a data selection scheme, which accounts not only for the labelers' quality, but also for the trade-off between data quality and diversity, so as to obtain a maximally diverse set of data with high quality;

5) *Performance evaluation:* we evaluate the proposed framework and compare it against state-of-the-art solutions, using three real-world datasets. Our results demonstrate the effectiveness of the proposed approach in improving the classification performance.

The rest of the paper is organized as follows. Sec. II discusses the related work and highlights the novelty of our study. Sec. III describes the system model, while Sec. IV presents the proposed AL framework, along with the label integration strategies. Sec. V presents the proposed schemes for labelers and data selection. Sec. VI discusses the scenarios and the data traces we used for our performance evaluation, and it shows the obtained results and the gain with respect to existing solutions. Finally, Sec. VII concludes the paper.

II. RELATED WORK

Leveraging V2V communication along with the advances of AI technologies, CAVs can learn the driving environment, make optimal decisions, and share traffic information with other vehicles and the network infrastructure [10]. AI technologies have been very successful in providing accurate, real-time activity classification. However, one of the main limitations of AI is the need for a large amount of accurately labeled data for the training (supervised learning) [11].

As far as AL is concerned, most of the existing works address the problem of noisy (or imperfect) labels in binary classification [12], [13], while very few tackle the multiclass (i.e., multi-label) case. Among the latter ones, [14]–[16] investigate the classification performance of crowdsourced data, where labeling can be done by volunteers or non-expert labelers. One of the crucial problems in crowdsourcing is data quality. As shown in [17], [18], when data/labelers are not selected carefully, the acquired data can be very noisy due to many reasons such as varying degrees of competence, individual biases, and misleading behavior. Moreover, the cost of acquiring a large amount of labeled data is non-trivial [19].

Such challenges have motivated the research for innovative ways to enhance the quality of data acquired from different labelers. In particular, [14] presents an AL-based, deep learning technique, leveraging volunteered geographic information to overcome the lack of big datasets; therein, a customized loss function is specifically defined to effectively deal with noisy labels and avoid performance degradation. [16] enhances

the performance of supervised learning with noisy labels in crowdsourcing systems by applying the majority voting label integration method and selecting the data referring to the same event whose label is sufficiently close to the resulting value. The work in [15] studies the problem of imbalanced noisy labeling in crowdsourcing systems, where the available labeled data is not evenly distributed across the different classes. First, it performs label integration and data selection based on data uncertainty and class imbalance level, then it classifies unlabeled data using the trained model and adds them and the associated labels to the training set. However, this work focuses on label integration and instance selection only, while neglecting the quality of the collected data (e.g., in terms of data freshness) as well as that of the labelers.

A different application is targeted in [8] where AL is used for incremental face identification. This study aims to build a classifier that progressively selects and labels the most informative data, and then it adds the newly labeled data to the training set. Furthermore, AL is combined with self-paced learning (SPL) – a recently developed learning scheme that gradually incorporates from easy to more complex data into the training set, with easy data being those with high classification confidence. Note that all of the above works consider specific classifiers (or loss functions), which cannot be easily incorporated in other learning techniques [20]. Thus, finding a label integration and data selection strategy that can be integrated with a generic multi-class classification scheme, is still an open problem.

With regard to cooperative applications leveraging V2V communications, it is worth noticing that several car manufacturers have already enabled their vehicles to share real-time hazard signals and to automatically alert each other [4]. The integration of V2V communication with machine learning to improve road safety has also received significant interest. In particular, [21] studies the impact of communication loss on 3D object detection exploiting a deep-learning approach. In [22], Gaussian process regression is used to estimate the age of the vehicles' data and proactively allocate, e.g., transmission power and resource blocks for reliable and low-latency V2V communication. [23] deals with the vehicle type recognition problem, in which labeling a sufficient amount of data is very time consuming. The solution presented in [23] exploits fully labeled web data to reduce the labeling time of surveillance data through deep transfer learning; also, only images of unlabeled vehicles with high uncertainty and diversity are selected to be queried.

Finally, a preliminary version of our study has been presented in [24], which introduces an efficient data selection scheme for online learning. Although [24] tackles the trade-off between data quality and diversity, it does not account for the reliability of neighboring vehicles. In this paper, we extend [24] to address the problem of weak or noisy labelers, by developing a labelers' selection scheme and, by doing so, selecting the optimal data subset for online training. Furthermore, we enhance the performance evaluation of the proposed framework, as well as the comparison against state-of-the-art solutions, by using additional real-world datasets.

Novelty. Our work is the first to address the problem of scarce and noisy data available to vehicles for classifying unexpected events. Moreover, unlike other studies, we assess the data quality level accounting for many factors, including freshness. Finally, the methods we propose for label integration as well as for labelers and data selection are general enough to be incorporated in different classification techniques or loss functions.

III. SYSTEM MODEL

We introduce below the main components of the system under study, along with the used notation. The road topology is divided into discrete *segments* $i \in \mathcal{I}$, while time is continuous. We denote with t the generic time instant and with t_{ij} the time at which vehicle $j \in \mathcal{V}$ is found at segment $i \in \mathcal{I}$.

Vehicles. We denote with $\mathcal{V}$ the set of all vehicles, and with $\mathcal{E}(t)$ the set of edges that exist at time t, i.e., pairs of vehicles within radio range of each other. Given an *ego vehicle* $v_0 \in \mathcal{V}$, we indicate by $\mathcal{N}_{v_0}(t)$ the set of neighbors of v_0 at time t, i.e., vehicles $v \in \mathcal{V}: (v_0, v) \in \mathcal{E}(t)$.

The classification task. In our scenario, each vehicle has its own active learning model running locally, which combines locally- and remotely-generated information. This information comes from onboard cameras and Advanced Driver Assistance Systems (ADAS); in the following, we refer to both sensor readings and features extracted from such information as *data*. The high-level purpose of the adaptive learning system we consider is to *classify* the collected data, associating each of them with a *label*. Labels, i.e., the output classes of the data, are the final output we get from the learning model after training it. The classification accuracy then expresses the proximity of the labels in the dataset to the ground truth. We denote with x_{ij} the data observed by vehicle $j \in \mathcal{V}$ while traveling in segment $i \in \mathcal{I}$, and with $y_{ij} \in \mathcal{L}$ the associated label ($\mathcal{L}$ is the set of all possible labels). Data observed by the ego vehicle and the locally-generated labels thereof are indicated by x_{i0} and y_{i0}, respectively. The combination of data, label, and time (x_{ij}, y_{ij}, t_{ij}) is referred to as *sample*.

Modes. As already mentioned and illustrated in Figure 2, vehicles can receive information from their neighbors. Specifically, we compare the following three *modes*:

- **labels**: the ego vehicle receives only the set $\mathcal{Y}_i = \{y_{i0}, y_{i1}, \ldots, y_{iJ}\}$ of labels generated by its neighbors $\{1, 2, \ldots, J\} \in \mathcal{N}_{v_0}(t)$;
- **data**: the ego vehicle receives only the set $\mathcal{X}_i = \{x_{i0}, x_{i1}, \ldots, x_{iJ}\}$ of data observed by its neighbors $\{1, 2, \ldots, J\} \in \mathcal{N}_{v_0}(t)$;
- **samples**: the ego vehicle receives both labels and data.

It is important to observe that modes significantly differ in the network usage they imply. Indeed, labels can be orders of magnitude smaller than the data they are based upon; hence, in terms of communication bandwidth, the "labels" mode is potentially much more efficient than the "data" and "samples" ones. Moreover, we remark that the proposed AL framework can be easily extended to consider Vehicle-to-everything (V2X) communications. In this case, depending on the adopted communication mode (e.g., V2I (vehicle-to-infrastructure), V2N (vehicle-to-network), V2D (vehicle-to-device), or V2V), the ego vehicle can identify the type of
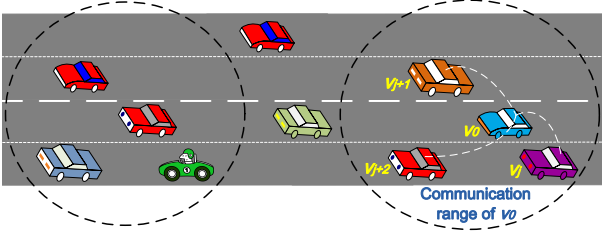
Fig. 2. The considered vehicles network with time-varying topology.

cooperative labelers and share with them its information, without affecting its active learning framework.

IV. ACTIVE LEARNING FRAMEWORK

In this section, we first introduce the proposed AL framework and the performance metrics we consider. Then, we discuss the basic majority voting method for labels integration, in addition to two more sophisticated methods: weighted majority voting and weighted average.

A. Methodology and performance metrics

Our framework, depicted in Figure 3, includes five main stages, as described below.

Offline learning: We consider that each vehicle has a certain amount of history, composed of M samples, through which it can initially train its learning model; as mentioned, M may be very small. With reference to such off-line dataset, we define the accuracy of the generic vehicle labeler j as,

$$A_j = \frac{\sum_{m=1}^{M} I(y_{mj}, y_m)}{M} \quad (1)$$

where y_{mj} is the label generated by j for sample m, y_m is the estimated label, and $I(u, w)$ is an indicator function, such that $I(u, w) = 1$ if $u = w$, and 0 otherwise.

Online labeling: Let us now consider that an event takes place at time t_{i0} while the ego vehicle v_0 is in road segment i. The vehicle acquires some information through its onboard sensors and, possibly, extracts some features (as mentioned, we refer to such new information as *data*); then v_0 labels such data through the learning model, obtaining *sample* (x_{i0}, y_{i0}, t_{i0}). Depending on the adopted operational mode, the ego vehicle shares with its neighbors, labels, data, or samples. Note that, in the data operational mode, v_0 has to label the received information using its own training model.

Label integration: After receiving the information from its neighboring vehicles, v_0 computes an aggregated label for the acquired data, using one of the label integration strategies reported below. Clearly, in label and sample mode, v_0 leverages the labels received from other vehicles, while in data mode it exploits the labels that v_0 itself has obtained for the locally generated data and for the received data. We denote the aggregated label with $\bar{y}_{i0}$; moreover, we define quality indicator q_{i0}, which, as detailed later, accounts for the accuracy (as labelers) of the vehicles A_j from which v_0 receives information, as well as for the data freshness. The samples referring to the situation in road segment i, to which the ego vehicle is exposed, are therefore described as

$(\mathcal{X}_{i0}, \bar{y}_{i0}, q_{i0})$, where $\mathcal{X}_{i0}$ is the data referring to such an event available at v_0.

Labeler selection The received labels/samples from the neighboring vehicles have a significant impact on the obtained accuracy at v_0. Thus, it is important to select the best neighboring vehicles that v_0 should consider. To do so, we propose a labelers selection scheme, where reputation values of diverse labelers are computed by subjective logic model. After monitoring the behavior of the neighboring vehicles, the ego vehicle v_0 evaluates their reputation values, based on past interactions, and selects a subset of labelers $\mathcal{S}_0 = \{v_1, v_2, \ldots, v_J\} \subset \mathcal{N}_{v_0}(t)$ with the highest reputation. It is assumed that all vehicles will have the same evaluation criteria to generate reputation values for the neighboring vehicles.

Data selection and classification: The ego vehicle selects the most appropriate set of samples to update its learning model. The goal of our data selection scheme is to find a maximally diverse collection of samples (with respect to all possible labels, i.e., data classes) in which each sample has as high quality as possible. Hence, the proposed scheme allows for selecting the highest quality data/samples from the highest reputation neighboring vehicles, subject to a diversity requirement. Using the selected samples, v_0 updates its training set, repeats the training[1] and performs classification. We remark that the on-line retraining typically happens at uncommon or rare situations, and it is executed by using a much smaller amount of data compared to offline training; thus it is requires much less time and it implies much less overhead than the initial (offline) training.

B. Label integration and quality definition

We consider and compare the following label integration methods for the computation of the aggregate label $\bar{y}_{i0}$.

Majority Voting (MV). The simplest and most popular label integration method is MV [15], which assumes no prior knowledge on the labelers' accuracy or data freshness. In MV, $\bar{y}_{i0}$ is computed as:

$$\bar{y}_{i0}^{(MV)} = \operatorname*{argmax}_{l \in \mathcal{L}} \sum_{j=0}^{|\mathcal{S}_0|} I(y_{ij} = l). \quad (2)$$

Given $\bar{y}_{i0}^{(MV)}$, a sample quality indicator, $q_{i0}^{MV} \in [0, 1]$, is:

$$q_{i0}^{MV} = \frac{1}{|\mathcal{S}_0| + 1} \left(\max_{l \in \mathcal{L}} \sum_{j=0}^{|\mathcal{S}_0|} I(y_{ij} = l) \right). \quad (3)$$

Besides neglecting data freshness, MV's performance is acceptable only when more than 50% of the labelers have high accuracy, which does not always hold in complex real-world scenarios [15]. Thus, in what follows, we propose alternative methods that aim at overcoming MV's weaknesses.

Weighted Majority Voting (WMV). We now define a probability of correctness, p_{ij}, representing the probability

---

[1]As better detailed in Sec. VI-A, we compare several different ML techniques in our performance evaluation. Some of them (e.g., neural networks) can leverage additional data to refine an already-trained model, while others (e.g., tree learning) cannot. For sake of uniformity, we therefore restart the training from scratch for all techniques.
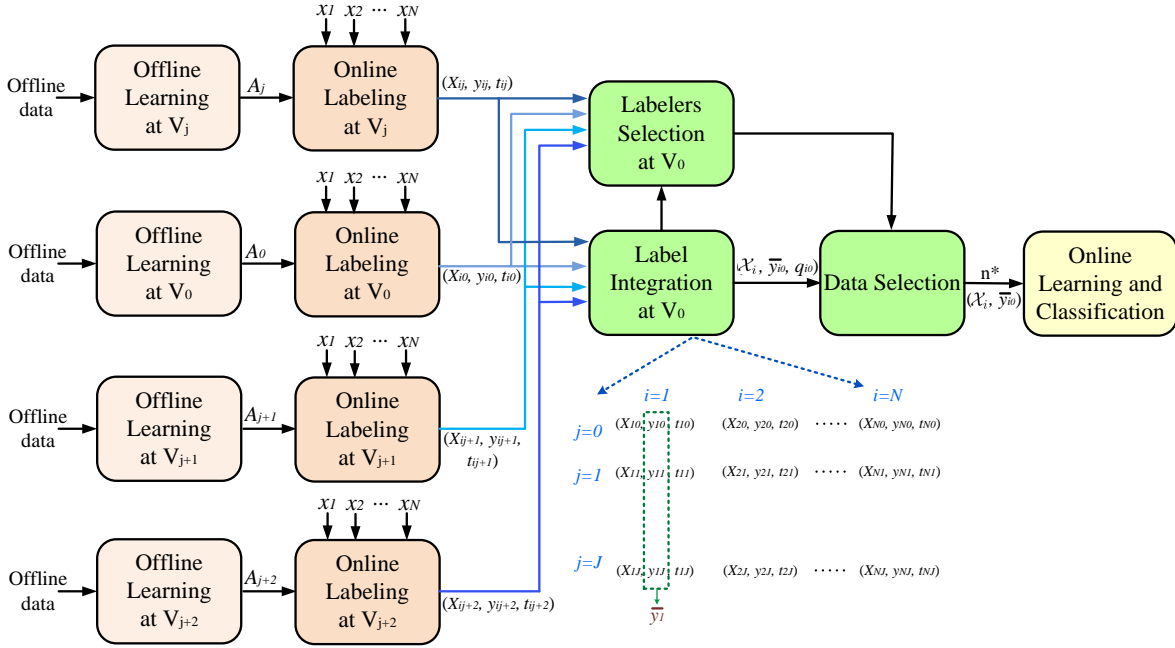
Fig. 3. Diagram representing the proposed AL framework, highlighting the different stages performed by the vehicles and the corresponding data flow.

with which v_0 receives a correct information from v_j. Such a probability depends on the labeler's accuracy and data freshness:

$$p_{ij} = f_{ij} \cdot A_j, \tag{4}$$

with the data freshness being defined as:

$$f_{ij} = \begin{cases} \exp[-(t_{i0} - t_{ij})] & t_{i0} > t_{ij} \\ 0 & t_{i0} \le t_{ij} \end{cases} \tag{5}$$

and taking on values ranging from 0 (totally stale data) to 1 (absolutely fresh data) [25]. Considering that p_{ij}'s are independent with respect to j [18], we write:

$$\bar{y}_{i0}^{(WMV)} = \underset{l \in \mathcal{L}}{\arg\max} \, \mathbb{P}(\bar{y}_{i0}^{(WMV)} = l | y_{i0}, \cdots, y_{iJ}). \tag{6}$$

Following the standard hypothesis testing procedure [18] and assuming for ease of presentation binary classification with equal priors, i.e., $\mathbb{P}(\bar{y}_{i0}^{(WMV)} = 1) = \mathbb{P}(\bar{y}_{i0}^{(WMV)} = -1)$, the aggregate label $\bar{y}_{i0}^{(WMV)}$ is given by:

$$\bar{y}_{i0}^{(WMV)} = \begin{cases} 1, & \rho_1 > \rho_2 \\ 0 & \rho_1 = \rho_2 \\ -1, & \rho_1 < \rho_2 \end{cases} \tag{7}$$

where $\rho_1 = \prod_{\{j:y_{ij}=1\}} p_{ij}$, and $\rho_2 = \prod_{\{j:y_{ij}=-1\}} p_{ij}$. Hence, the quality indicator of WMV is defined as

$$q_{i0}^{WMV} = \max_{l \in \mathcal{L}} \prod_{\{j:y_{ij}=l\}} p_{ij}. \tag{8}$$

Weighted Average (WA). The WA method relies on defining a weighting coefficient accounting for both the labelers' accuracy and data freshness: $\lambda_{ij} = a \cdot f_{ij} + b \cdot A_j$, where a and b are constants representing the importance of f_{ij} and A_j, respectively. Then, the aggregate label is defined as

$$\bar{y}_{i0}^{WA} = \underset{l \in \mathcal{L}}{\arg\max} \sum_{j=0}^{|\mathcal{S}_0|} \exp(\lambda_{ij}) \cdot I(y_{ij} = l). \tag{9}$$

We remark that the exponential function in (9) is used to weight more the labels associated with high classification confidence, thereby making it more descriptive than a simple average. The quality indicator of WA is then given by:

$$q_{i0}^{WA} = \max_{l \in \mathcal{L}} \sum_{j=0}^{|\mathcal{S}_0|} \exp(\lambda_{ij}) \cdot I(y_{ij} = l). \tag{10}$$

To assess the performance of our label integration methods, we define the Labeling Accuracy (LA) for the ego vehicle as

$$LA_0 = \frac{\sum_{i=1}^{\Omega} I(\bar{y}_{i0}, \gamma_i)}{\Omega} \tag{11}$$

where Ω is the size of the testing dataset for the event currently occurring in road segment i, and γ_i is the ground truth.

We remark that the WMV and WA methods account for the labelers' quality and freshness; also, WA leverages an exponential function with a weighting coefficient to magnify the effect of high-quality labelers, which improves the performance compared to MV and WMV.

V. LABELERS AND DATA SELECTION

In this section, we detail the two core procedures of our framework, i.e., labelers selection and data selection.

A. Labelers' selection

Since we are dealing with noisy/unreliable sources of information (i.e., labelers), we opt to leverage subjective logic to select the most reliable labelers. Subjective logic is a type of probabilistic logic that accounts for uncertainty of diverse sources to model and analyze the systems involving relatively unreliable sources [26]. Indeed, we aim to define a reputation function, to be used at each vehicle, to model the interactions

with different vehicles. The proposed reputation scheme considers two types of interactions, namely, positive interactions and negative interactions. The former means that a vehicle believes that the received information from the neighboring vehicle (or labeler) is true, i.e., its label is consistent with the aggregate label $\bar{y}_{i0}$. The latter means that the received information may be misleading or unreliable, since it conflicts with the remaining information acquired from other labelers.

Considering the ego vehicle v_0 and a neighboring vehicle v_j, the trustworthiness or local opinion (as in subjective logic [27]) of v_0 about v_j is formally defined as a function of $b_{0j}, d_{0j}, u_{0j}, a_{0j}$. Herein, b_{0j}, d_{0j}, and u_{0j} represent the belief, disbelief, and uncertainty of v_0 in v_j, respectively, while a_{0j} is the base rate. The belief is the probability that the received information from v_j is true, disbelief is the probability that the received information is false, and uncertainty is the confidence in the obtained knowledge about v_j. The base rate is the prior probability in the absence of belief or disbelief. Hence, based on the subjective logic model in [27], [28], we have that:

$$b_{0j}, d_{0j}, u_{0j}, a_{0j} \in [0, 1], \text{ and } b_{0j} + d_{0j} + u_{0j} = 1. \quad (12)$$

In our framework, we define the reputation value of v_j at v_0, as follows:

$$R_{0j} = \delta \cdot b_{0j} + (1 - \delta) \cdot a_{0j}, \quad \text{with} \quad (13)$$

$$b_{0j} = (1 - u_{0j}) \frac{\alpha_{0j}}{\alpha_{0j} + \beta_{0j}}. \quad (14)$$

In (13) and (14), we have:
- $\alpha_{0j} = \sum_{i=1}^{N_j} I(y_{ij} = \bar{y}_{i0})$ is the number of positive interactions between v_j and v_0, and $\beta_{0j} = \sum_{i=1}^{N_j} I(y_{ij} \neq \bar{y}_{i0})$ is the number of negative interactions, where N_j is the number of labels, data, or samples received from v_j at v_0;
- $\delta = \min \left\{ \frac{\alpha_{0j} + \beta_{0j}}{\theta}, 1 \right\}$, where θ is the number of interactions that reveals high prior knowledge about v_j;
- $a_{0j} = A_j$;
- the uncertainty u_{0j} is mainly obtained by the communication quality of the link between v_0 and v_j.

Accordingly, after receiving the information from the neighboring vehicles, v_0 updates the reputation values of different labelers, using (13). Hence, the labelers with high reputation can be selected.

After calculating the reputation of diverse labelers, we address the tradeoff between the labeling quality and network load for the selected subset of labelers $\mathcal{S}_0$ at v_0. Let φ_0 be the average quality of the acquired samples; e.g., when the WA method is used for label integration, φ_0 is defined as:

$$\varphi_0 = \frac{\sum_{i \in \mathcal{Z}} q_{i0}^{WA}}{|\mathcal{Z}|}, \quad (15)$$

where $\mathcal{Z}$ is the set of the latest consecutive road segments along which the ego vehicle has traveled. The network load is defined as $B_0 = \psi \cdot \sum_{j=1}^{|\mathcal{S}_0|} N_j$, where ψ is the size of the exchanged message between the neighboring vehicles. Both objectives (i.e., labeling quality and network load) are normalized with respect to their maximum values, then they are combined into the following utility function:

$$U_0 = \eta \cdot \tilde{\varphi}_0 - (1 - \eta) \cdot \tilde{B}_0 \quad (16)$$

where $\tilde{\varphi}_0$ and $\tilde{B}_0$ are the normalized quality and load, and $0 \leq \eta \leq 1$ is a constant trading off the labeling quality and network load. Accordingly, we define the optimal selected subset of labelers at v_0 as

$$\mathcal{S}_0^* = \underset{\mathcal{S}_0 \in \mathcal{N}_{v_0}(t)}{\arg\max} U_0. \quad (17)$$

To efficiently obtain $\mathcal{S}_0^*$, we propose the Labelers' Selection (LS) algorithm reported in Algorithm 1. The LS algorithm leverages the monotonicity property of the optimal set, in order to decrease the complexity of searching for the optimal subset of labelers. Such a property is proved by the lemma below.

Lemma 1: *The optimal set $\mathcal{S}_0^*$ is monotonic, i.e., if we have $j \in \mathcal{S}_0^*$ and $i \notin \mathcal{S}_0^*$ then we must have $R_{0j} > R_{0i}$.*

Proof: Suppose that a labeler i with reputation R_{0i} has been selected from a set of optimal labelers $\widehat{\mathcal{S}}_0(i)$, while a labeler j with reputation R_{0j} has been selected from a non-optimal set of labelers $\mathcal{S}_0(j)$. Given that $R_{0j} > R_{0i}$, the labeling accuracy of labeler j must be greater than the accuracy of labeler i, based on (13). From (10) and (15), we can infer that $\varphi_0(j) > \varphi_0(i)$, hence $U_0(j) > U_0(i)$. This contradicts our first assumption that $\widehat{\mathcal{S}}_0(i)$ is the optimal set of labelers, according to the definition given in (17). Hence, the optimal set S_0^* must include labeler j that has high reputation, while adding any other labelers with lower reputation instead of j would result in a non-optimal set. ∎

The above lemma can significantly reduce the search complexity for the optimal set $\mathcal{S}_0^*$ by first ordering all labelers in descending order with respect to their reputation R_{0j}. Then, for each possible selected labelers set size $1, 2, \cdots, |\mathcal{N}_{v_0}(t)|$, we obtain the optimal set by sequentially adding labelers with high reputation, and till when including one more labeler leads to a decrease in U_0. We highlight that, after defining the optimal subset of labelers $\mathcal{S}_0^*$, their information will be considered in the following procedure to obtain a maximally diverse set of data with high quality.

B. Data subset selection

The objective of our data selection algorithm, named Quality-Diversity Selection (QDS), is to obtain a subset of high-quality data to be added to the online training set, so as to maximize the model classification accuracy. We highlight that, unlike most of the existing quality-based schemes to data selection that result in a reduced samples' diversity, our approach efficiently trades off quality and diversity, thus it significantly improves the performance of the AL framework. Furthermore, the QDS algorithm not only determines *which* samples should be selected but also *how many* should be added to the training set.

The diversity score is measured based on the entropy of the selected samples [29]: $H(\mathcal{X}) = -\sum_{k=1}^{K} \chi_k \log_2 \chi_k$, where

---

Algorithm 1 Labelers' Selection (LS) Algorithm

1: **Input:** $A_j, (\mathcal{X}_{i0}, \mathcal{Y}_{i0}, \mathcal{T}_i), \forall i \in \{1, \cdots N\}$
2: Compute $\bar{y}_{i0}$ and q_{i0} using a label integration method (e.g., MV, WMV, WA)
3: Obtain the reputation value for each labeler $j, \forall j \in \{1, \cdots |\mathcal{N}_{v_0}(t)|\}$ using (13)
4: Rank all labelers in descending order with respect to their reputation values
5: Initially: set $j = 1, \mathcal{S}_0(j) = \{v_1\}$, and compute $U_0(j)$ using (16)
6: **for** $j = 2 : |\mathcal{N}_{v_0}(t)|$ **do**
7:     Add labeler j to the selected subset of labelers $\mathcal{S}_0(j)$
8:     Compute $U_0(j)$ using (16)
9:     **if** $U_0(j) < U_0(j-1)$ **then**
10:         $\mathcal{S}_0^* = \mathcal{S}_0(j-1)$
11:         Break   ▷ No further addition is worth it
12:     **else**
13:         $\mathcal{S}_0^* = \mathcal{S}_0(j)$
14:     **end if**
15: **end for**
16: **return** $\mathcal{S}_0^*$

χ_k is the fraction of samples belonging to class k, and K is the number of classes defined for the classification task. The labeling quality of the selected samples is defined as in (15), where n is the number of selected samples. Accordingly, the sample selection is conducted in two steps:

- **Selecting class k^*:** the class of samples to target is chosen so as to maximize diversity, i.e., $k^* = \mathrm{argmax}_k H(\mathcal{X})$. The idea is indeed that the more diverse samples are selected, hence the more balanced their distribution across the different classes, the more informative they will be.
- **Selecting the samples:** given class k^*, the samples with the best quality are selected such that the labeling quality is maximized, i.e., $\mathcal{X}^* = \mathrm{argmax}_\mathcal{X} \varphi_0(\mathcal{X})$.

The selected samples are added to the on-line training set and the classification accuracy of the AL framework is checked by labeling the data in the testing set. If the obtained classification accuracy $\widehat{\alpha}$ is below the desired predefined value α, one more sample is selected, till accuracy α is reached. The proposed QDS algorithm is summarized in Algorithm 2, where n^* is the number of selected samples.

Algorithm 2 Quality-Diversity Selection (QDS) Algorithm

1: **Input:** $\mathcal{S}_0^*, (\mathcal{X}_{i0}, \mathcal{Y}_{i0}, \mathcal{T}_i), \bar{y}_{i0}, q_{i0}, \forall i \in \{1, \cdots N\}$.
2: Identify the selected class k^*
3: Given k^*, select samples with maximum quality $\mathcal{X}^*$
4: Add selected samples to the online training set $\mathcal{O}$
5: Compute $\widehat{\alpha}$
6: **if** $\widehat{\alpha} > \alpha$ **then**
7:     $n^* = |\mathcal{O}|$
8:     Break   ▷ n^* is obtained
9: **else**
10:     Go to step 2.
11: **end if**
12: **return** $n^*, \mathcal{O}$

VI. Performance Evaluation

In this section, we first present the simulation environment that is used to derive our results. Then, we assess the performance of the proposed AL framework compared to state-of-the-art techniques. In particular, our results mainly focus on label integration assessment, data and labelers selection assessment, as well as studying the effect of data freshness on the obtained classification accuracy.

A. Simulation environment

In our performance evaluation, we use three datasets, called *vehicles*, *Antwerp*, and *MNIST*. The *vehicles* dataset [30] includes a set of photos of four types of vehicles (namely, a double decker bus, a Cheverolet van, a Saab 9000, and an Opel Manta 400). The images of the *vehicles* dataset were processed with the BINATTS image processing system, extracting a combination of scale-independent features through a combination of classical moment-based measures and heuristic ones, like circularity, rectangular, and compactness. Thus, in the *vehicles* dataset, the data refers to the photos of the vehicles, while the labels are the types of vehicles.

The *Antwerp* dataset [31] represents the traffic demand of the ring road of Antwerp during a typical day. This dataset includes the vehicles' speed and the per road-section density (i.e., the number of vehicles currently in a section), along with global indicators of the transportation system. All these features (i.e., data) are reported every 60 seconds. We use this dataset to detect the road density, which is divided into six classes (or labels). The *MNIST* dataset [32] is a large database of handwritten digits, which is widely used for the training and testing of different supervised machine learning schemes. The data here refers to the photos of the handwritten digits – one field per pixel –, while the labels are the types of digits (i.e., from 0 to 9).

Each dataset has been divided into three sets: offline training set, online training set, and testing set. For the label integration assessment, we use the three datasets and devote to each of them a separate plot showing the labeling accuracy as a function of the size of the offline training set. Then, to avoid redundancy, we use the *MNIST* dataset for the data selection and data freshness assessment, while we leverage the *Antwerp* dataset for the labelers' selection assessment.

To model the fact that vehicles may have different *quality* levels (e.g., quality of their sensors, camera, and computational capabilities), each vehicle is assigned a different classification model [33] (one classifier among the following ones: fine tree, medium tree, linear SVM, medium Gaussian SVM, linear discriminant, and weighted KNN), with the best-performing classifiers associated with the highest quality vehicles. It follows that low-quality vehicles are more likely to have incorrect labels. The considered classifiers are run using MATLAB Toolbox.

B. Results

The first aspect we are interested in is the impact of the label integration methods on the labeling accuracy (LA). To this
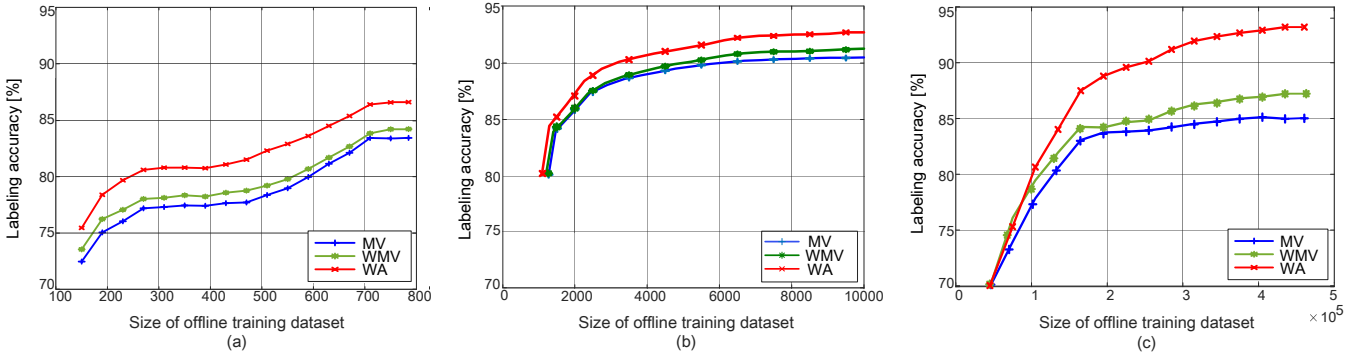
Fig. 4. Labeling accuracy as a function of the size M of the offline training set, for different label integration methods while considering samples mode, using: (a) the *vehicles* dataset, (b) the *MNIST* dataset, and (c) the *Antwerp* dataset.
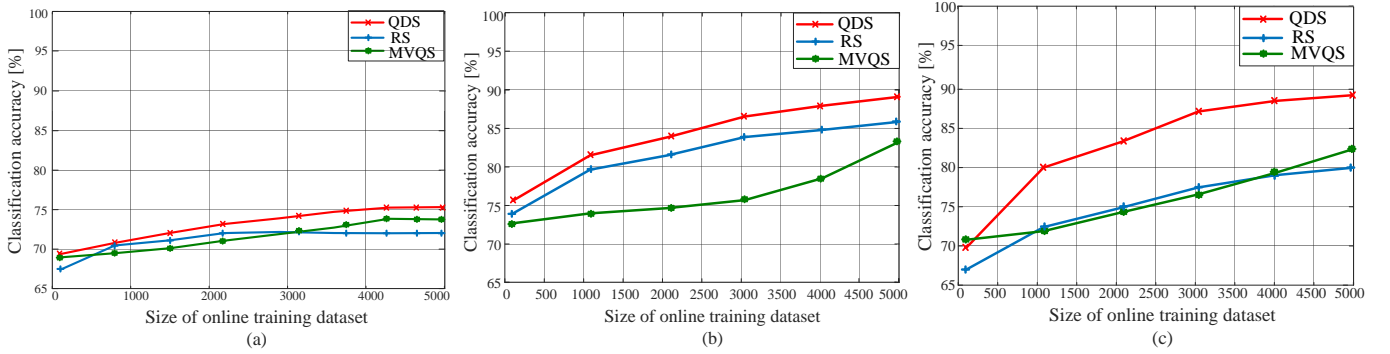

Fig. 5. Classification accuracy as a function of the size of the online training set, for different data selection strategies, under the (a) labels, (b) data, and (c) samples modes.

end, Figure 4 presents the LA as a function of the size M of the offline training set, for different datasets. Unless otherwise specified, we assume that the ego vehicle v_0 is helped by a total of four neighbors. It is possible to observe how a larger training set always corresponds to a better accuracy. More interestingly, the weighted average integration (WA) yields substantially better accuracy than majority voting (MV) and weighed majority voting (WMV). An intuitive explanation is that WA is able to use all available samples, while at the same time accounting for their quality and freshness. This resembles such standard machine learning techniques as dropout and data augmentation, where training on a more challenging dataset improves performance. Indeed, the WA method relies on defining a weighting coefficient that accounts for both the labelers' accuracy and data freshness. Hence, it is used to weight more the labels associated with high classification confidence, thereby making the results more descriptive than a simple average. Based on this result, we use the WA integration method in the remainder of our performance evaluation.

The second aspect we investigate is how the data selection algorithm and the mode of operation influence the performance. Figure 5 compares the proposed approach against a state-of-the-art solution presented in [16], hereinafter referred to as majority voting quality selection (MVQS). The MVQS scheme addresses the problem of weakly labeled crowdsourced data and aims to select for training those data that have the best quality labels, while using the MV method for label integration. Furthermore, we compare our QDS approach against a baseline approach, namely, the random-selection approach

(RS), where the WV method is used for label integration and the samples are selected randomly for training.

The plots in Figure 5 depict the classification accuracy as a function of the online training set size; each curve therein corresponds to a data selection algorithm, and each plot corresponds to a different mode. Comparing the individual lines within each plot, it is possible to observe how our own QDS algorithm consistently outperforms both the MVQS and RS approaches. This result suggests also that samples' quality is not the only factor to account for when assembling a training set, rather labelers' accuracy shall be considered as well. Looking at the three plots, it is clear that the samples mode is associated with higher performance than the data mode, and both outperform the labels mode; consistently with one's intuition, more information – be it labels or data – translates into better performance. Based on this result, in the following we focus on the data mode and samples mode.

The better performance of the data and samples modes comes, however, at the cost of an increased network load, as summarized in Figure 6-(a). Each marker therein corresponds to a combination of mode and training size, and its x- and y-coordinates (respectively) correspond to the network load and the achieved classification accuracy. The figure highlights how different trade-offs between network load and classification accuracy can be pursued and that, in general, the two quantities are strongly correlated. In Figure 5 and Figure 6, the *MNIST* dataset is used, while the network load is calculated by assuming that the size of the exchanged messages between the neighboring vehicles ψ is equal to 2 bytes in the case of
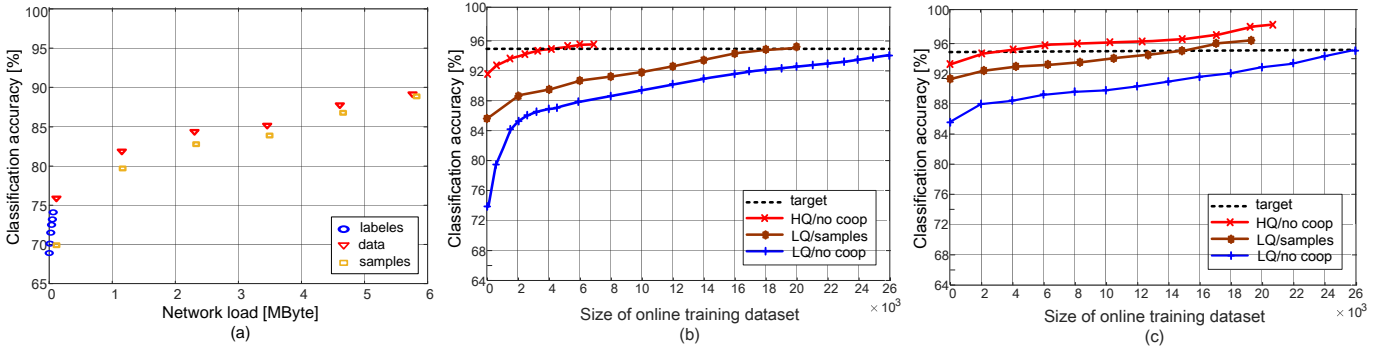
Fig. 6.   (a) Trade-offs between classification accuracy and network load; (b) effect of cooperation on classification accuracy when the size of the offline training set is $M = 500$ and (c) $M = 2,000$, while considering data mode.

labels mode, 180 bytes in the case of data mode, and 182 bytes in the case of samples mode.

The third aspect we are interested in is the issue of cooperation between vehicles, and, in particular, how much, and to whom, cooperation is beneficial. The plots in Figure 6-(b) and Figure 6-(c) depict how, for a fixed size of the offline training set, the quantity of available *online* training data influences the classification accuracy. Different lines in these plots correspond to high-quality (HQ/no coop) and low quality (LQ/no coop) vehicles with no cooperation, as well as to low-quality vehicles operating in samples mode (LQ/samples). It is clear that cooperation yields a substantial performance advantage for low-quality vehicles, which can reach the desired level of accuracy ($\alpha = 0.95$ in the plots) with a substantially smaller number of samples, hence, in a much shorter time.

Next, in Figure 7–Figure 11, we study the effect of the proposed labelers' selection (LS) algorithm on the obtained classification accuracy, using the *Antwerp* dataset, while considering the samples mode. We now assume that the ego vehicle v_0 can be helped by a total of seven neighboring labelers. We further assume that the accuracy of the neighboring labelers are normally distributed according to distribution $\mathcal{N}(\mu, \sigma^2)$, with mean $\mu = 83\%$ and variance $\sigma^2 = 162.6$. Figure 7-(a) and Figure 7-(b) show the classification accuracy and the average labeling quality φ_0, respectively, as the online training set size varies. Each curve in Figure 7 corresponds to a number of selected labelers, while the dotted curve corresponds to the optimal subset of labelers selected by our LS algorithm. Interestingly, Figure 7 highlights that the increase in the number of labelers interacting with the ego vehicle v_0 does not always lead to improving the labeling quality or the classification accuracy. In fact, the classification accuracy reaches a maximum for a certain number of labelers, after which receiving samples from more labelers may harm the performance. The reason is that the newly-added samples may arrive from low-quality labelers, which may confuse the classifier rather than enhancing the learning process. This confirms the importance of selecting only high-quality labelers among the available neighbors. Importantly, our labeler selection algorithm is especially well-suited for this situation, since it allows to select the optimal subset of labelers. Interestingly, by considering both the labeling quality and network load, our scheme could select the minimum number of (high-reputation)
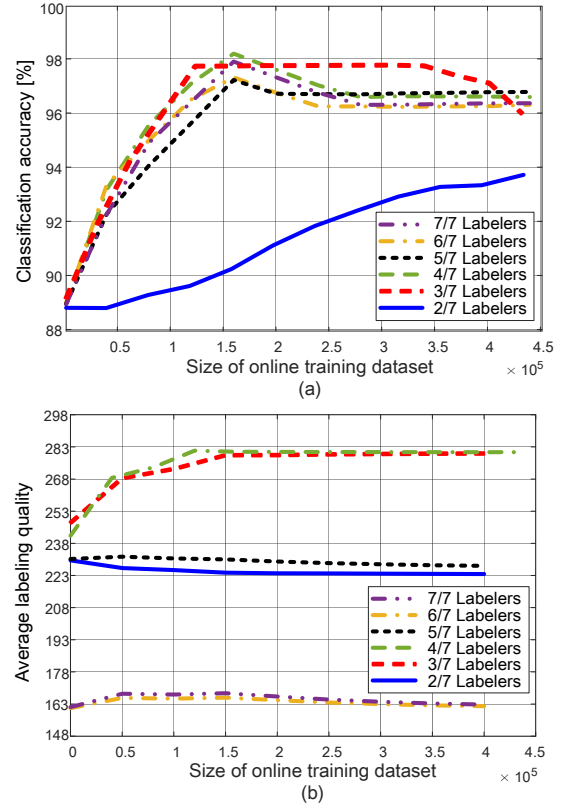

Fig. 7.   Impact of the number and quality of labelers on the obtained: (a) classification accuracy, and (b) average labeling quality, as the size of the online training set varies, while using the *Antwerp* dataset.

labelers (i.e., 3 out of 7) that yields adequate classification performance without overloading the network.

Given the selected optimal subset of labelers (i.e., 3 out of 7), Figure 8 depicts the effect of offline training set size on the obtained performance. This figure highlights that high-reputation labelers need to have at their disposal a sufficient amount of offline training data to generate high-quality labels. Consequently, by receiving high-quality samples from different neighbors, the online training yields a substantial enhancement in the classification accuracy for vehicles with small offline training set.

To further illustrate the gain of the proposed LS algorithm, we compare it in Figure 9 against two baselines: Random
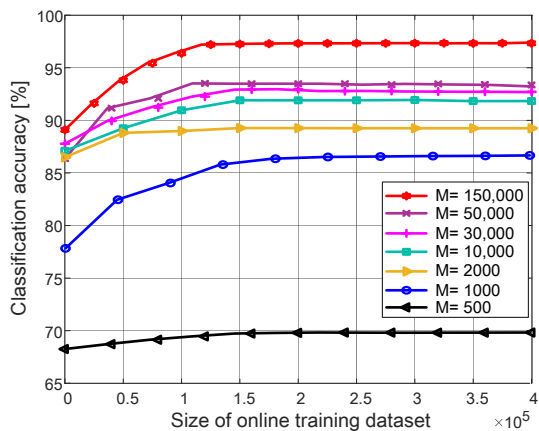
Fig. 8. Classification accuracy as a function of the size of the online training set, for different values of the offline training set size (M). The best 3 labelers out of 7 are selected.
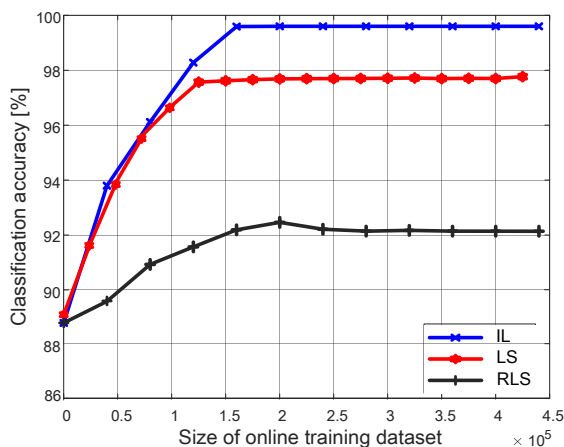

Fig. 9. Comparison among the different labelers' selection strategies in terms of classification accuracy, as the online training set size varies.

Labelers Selection (RLS) and Ideal Labelers (IL) selection. The former randomly selects the labelers from the available labelers pool, while the latter always selects the optimal labelers in an ideal environment; in such an environment, all labelers generate samples with 100% labeling accuracy. In this figure, we consider that each labelers selection strategy picks up three labelers out of the available seven. As shown, the proposed LS algorithm consistently outperforms the RLS approach by selecting the optimal subset of labelers. Also, it can be seen that our LS algorithm substantially improves the classification accuracy to approach the benchmark solution with ideal labelers.

Another interesting advantage for our LS algorithm is its robustness against Bad-quality Labelers (BLs). Such labelers may generate fake/low-quality samples or labels, which may confuse the neighboring vehicles. However, the proposed LS algorithm can tackle this issue, thanks to the proposed labelers' reputation scheme. After interacting with bad-quality labelers, the LS algorithm can detect the inconsistency in the received samples compared to the aggregated samples from Good-quality Labelers (GLs). Hence, the LS algorithm assigns a low reputation to bad-quality labelers and, by selecting only the high-reputation labelers, it avoids receiving information from

bad-quality labelers. In Figure 10, we consider that the ego vehicle has four bad-quality labelers with a labeling accuracy that follows the uniform distribution $\mathcal{U}(70\%, 78\%)$, while it has three good-quality labelers with a labeling accuracy that follows the uniform distribution $\mathcal{U}(90\%, 98\%)$. Figure 10-(a) shows the classification accuracy variations as functions of the size of the online training, while increasing the number of selected labelers using our LS algorithm. Conversely, Figure 10-(b) shows the classification accuracy variations, while increasing the number of selected labelers using RLS.

We can observe that our LS algorithm always outperforms the RLS approach for different numbers of selected labelers. However, increasing the number of labelers results in a significant drop in the classification accuracy, due to receiving information from low-quality labelers. Also, we can see that for large number of selected labelers (i.e., 6 or 7), the obtained accuracy increases at the beginning then decreases as the size of the online training set grows. In both cases, the quality drop is a consequence of the inclusion of low quality information coming from bad-quality labelers, which harm the learning process and confuse the classifier. As shown by the dotted curve in Figure 10-(a), the proposed LS and QDS schemes allow the selection of the *optimal* – not necessarily the largest – set of information to use, resulting in better classification performance.

In Figure 11, we depict the effect of increasing the number of selected labelers on the obtained classification accuracy and time of convergence for the IL selection, where all labelers have 100% labeling accuracy. We define the time of convergence as the time needed to reach the desired level of accuracy. It is clear that any increase in the number of selected labelers will directly result in a decrease of the time of convergence, since the ego vehicle will be able to aggregate a large number of samples from different labelers in a much shorter time.

Finally, in Figure 13 we look at the effect of data freshness on the obtained classification accuracy using the *MNIST* dataset and while considering the data mode. First, we model data freshness within the considered dataset as follows:

- The road is divided into segments, where it is assumed that an *event* (that needs to be detected by all vehicles) has occurred in a segment i_0 at time t_0 (see Figure 12);
- The neighboring vehicles at segment i_0 and time t_0 can acquire High Freshness (HF) data, while the vehicles located at segment i_1 and time t_0 can acquire Medium Freshness (MF) data;
- At time $t_0 + \Delta$, the vehicles at segment i_0 will have the MF data, while the vehicles located at segment i_1 will have Low Freshness (LF) data.

We then establish a link between freshness and sample quality, by associating low-quality samples with low-freshness, medium-quality samples with medium-freshness, and high-quality samples with high-freshness.

In Figure 13, we assume that the online training set is acquired at v_0 from the neighboring vehicles with different levels of freshness. First, the results highlight that classification accuracy can be significantly improved with increasing level of data freshness. Also, increasing the online training set size
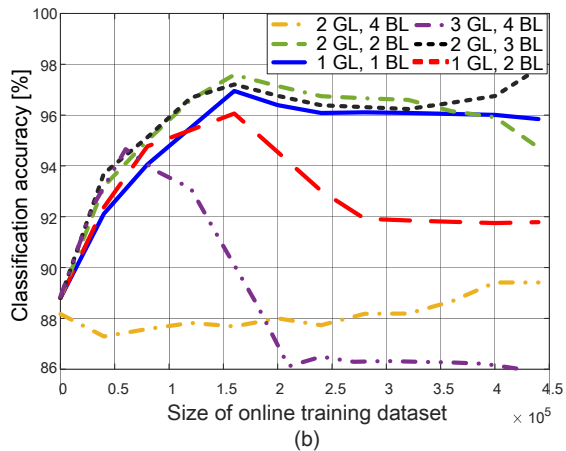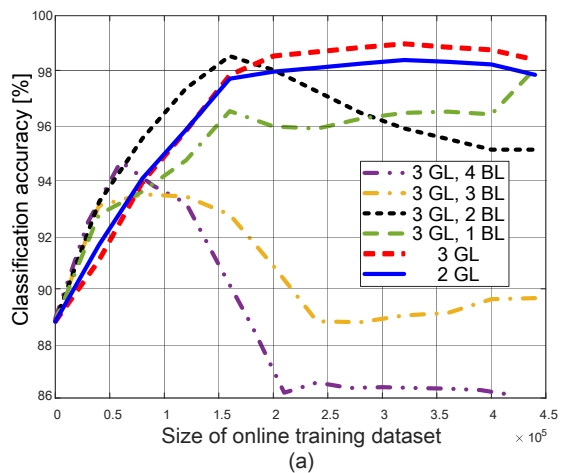
Fig. 12. A snapshot from the considered CAV network at: (a) time t_0, and (b) time $t_0 + \Delta$.

Fig. 13. Classification accuracy as a function of the size of the online training set, for different levels of freshness and $M = 500$, while considering the data mode.

Fig. 10. Effect of the number and quality of labelers on the obtained classification accuracy, for a varying size of the online training set: (a) LS algorithm and (b) RLS. For the ego vehicle, the availability of 3 good-quality labelers and 4 bad-quality labelers is considered.
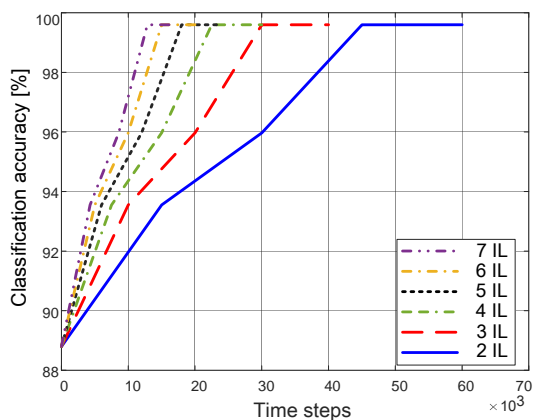

Fig. 11. Time of convergence for the IL selection, for different numbers of labelers.

improves the performance when a vehicle v_0 receives high freshness data from its neighboring vehicles. On the contrary, receiving LF data will have a minor effect on enhancing classification accuracy, even with an increasing size of the online training set. Second, the figure depicts the effect of AL cooperation on vehicles with MF data and LF data. By enabling the cooperation between vehicles with HF data and
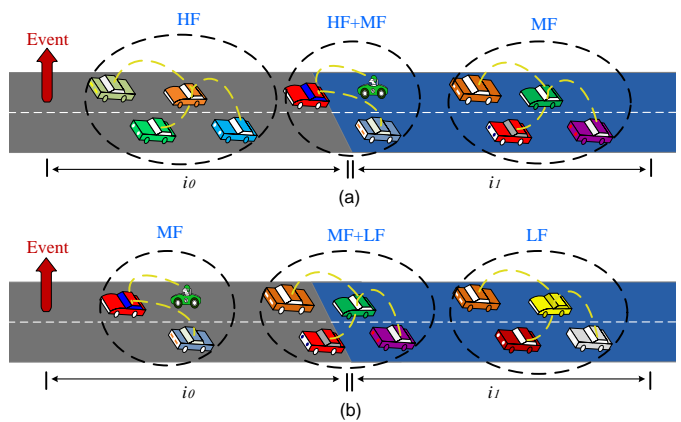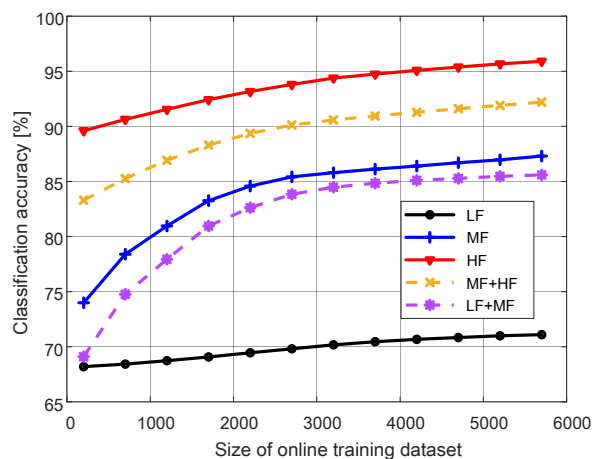
vehicles with MF data, the latter can significantly improve their performance. This also applies to vehicles with LF data when they cooperate with vehicles with MF data.

VII. CONCLUSION

We proposed an active learning framework for connected automated vehicles, which leverages vehicle-to-vehicle communication to increase the amount of collected data in the training set. Given that a vehicle can receive from its neighbors multiple data, labels, or a combination of the two, we proposed label integration methods, as well as labelers and data selection algorithms, which account for the labelers' accuracy, data freshness, and data diversity. In this context, we formulated labelers' reputation and samples quality indicators and provided a theoretical analysis for the proposed labelers and data selection algorithms. We evaluated our approaches using different real-world datasets, and we showed that they outperform state-of-the-art solutions. In particular, numerical results highlight their effectiveness and ability to provide 5-10% increase in classification accuracy, with respect to conventional active learning schemes that consider majority voting for labels integration and/or random data selection.

Furthermore, the proposed algorithms can efficiently differentiate between high and low quality labelers over time and select the optimal subset of labelers and data, thus substantially improving the classification accuracy by 6% compared to random labelers selection scheme. As future work, it would be interesting to combine the proposed AL framework with emerging distributed learning techniques (such as federated learning). Since most of the existing schemes for distributed learning suffer from slow convergence, a new AL solution would be needed, which leverages the distributed nature of vehicular networks, while coping with such highly dynamic scenarios and ensuring sufficiently fast convergence.

REFERENCES

[1] X. Cheng, R. Zhang, and L. Yang, "Wireless toward the era of intelligent vehicles," *IEEE Internet of Things Journal*, vol. 6, no. 1, 2019.

[2] S. Zhang, J. Chen, F. Lyu, N. Cheng, W. Shi, and X. Shen, "Vehicular communication networks in the automated driving era," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 26–32, 2018.

[3] M. Zhou, Y. Yu, and X. Qu, "Development of an efficient driving strategy for connected and automated vehicles at signalized intersections: A reinforcement learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 1, pp. 433–443, 2020.

[4] E. Uhlemann, "Time for autonomous vehicles to connect [connected vehicles]," *IEEE Veh. Tech. Mag.*, vol. 13, no. 3, pp. 10–13, 2018.

[5] G. Giordano, M. Segata, F. Blanchini, and R. L. Cigno, "The joint network/control design of platooning algorithms can enforce guaranteed safety constraints," *Ad Hoc Networks*, 2019.

[6] S. Huang, R. Jin, and Z. Zhou, "Active learning by querying informative and representative examples," *IEEE Trans. on Pattern Analysis and Machine Intell.*, vol. 36, no. 10, pp. 1936–1949, 2014.

[7] K. Wang, D. Zhang, Y. Li, R. Zhang, and L. Lin, "Cost-effective active learning for deep image classification," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 27, no. 12, pp. 2591–2600, 2017.

[8] L. Lin, K. Wang, D. Meng, W. Zuo, and L. Zhang, "Active self-paced learning for cost-effective and progressive face identification," *IEEE Trans. on Pattern Analysis and Machine Intell.*, vol. 40, no. 1, 2018.

[9] J. Konečný, B. McMahan, and D. Ramage, "Federated optimization: Distributed optimization beyond the datacenter," *arXiv preprint arXiv:1511.03575*, 2015.

[10] Y. Fu, C. Li, R. Yu, T. H. Luan, and Y. Zhang, "A decision-making strategy for vehicle autonomous braking in emergency via deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, pp. 1–1, 2020.

[11] K. Lin, Y. Li, J. Deng, P. Pace, and G. Fortino, "Clustering-learning-based long-term predictive localization in 5g-envisioned internet of connected vehicles," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–15, 2020.

[12] T. Liu and D. Tao, "Classification with noisy labels by importance reweighting," *IEEE Trans. on Pattern Analysis and Machine Intell.*, vol. 38, no. 3, pp. 447–461, 2016.

[13] B. van Rooyen, A. Menon, and R. C. Williamson, "Learning with symmetric label noise: The importance of being unhinged," *NIPS*, pp. 10–18, 2015.

[14] J. Chen, Y. Zhou, A. Zipf, and H. Fan, "Deep learning from multiple crowds: A case study of humanitarian mapping," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 57, no. 3, pp. 1713–1722, 2019.

[15] J. Zhang, X. Wu, and V. S. Shengs, "Active learning with imbalanced multiple noisy labeling," *IEEE Trans. on Cybernetics*, vol. 45, no. 5, pp. 1095–1107, 2015.

[16] M. Fang, T. Zhou, J. Yin, Y. Wang, and D. Tao, "Data subset selection with imperfect multiple labels," *IEEE Trans. on Neural Networks and Learning Syst.*, vol. 30, no. 7, pp. 2212–2221, 2019.

[17] G. Hua, C. Long, M. Yang, and Y. Gao, "Collaborative active visual recognition from crowds: A distributed ensemble approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 582–594, 2018.

[18] Y. Liu and M. Liu, "An online learning approach to improving the quality of crowd-sourcing," *IEEE/ACM Trans. on Networking*, vol. 25, no. 4, pp. 2166–2179, 2017.

[19] C. Aggarwal, *Data Mining: The Textbook*. Springer International Publishing, 2015. [Online]. Available: https://books.google.com.qa/books?id=cfNICAAAQBAJ

[20] R. Wang, T. Liu, and D. Tao, "Multiclass learning with partially corrupted labels," *IEEE Trans. on Neural Networks and Learning Systems*, vol. 29, no. 6, pp. 2568–2580, 2018.

[21] Y. Wang, V. Menkovski, I. W. Ho, and M. Pechenizkiy, "Vanet meets deep learning: The effect of packet loss on the object detection performance," in *VTC-Spring*, 2019, pp. 1–5.

[22] M. K. Abdel-Aziz, S. Samarakoon, M. Bennis, and W. Saad, "Ultra-reliable and low-latency vehicular communication: An active learning approach," *IEEE Comm. Letters*, pp. 1–1, 2019.

[23] Y. Huang, Z. Liu, M. Jiang, X. Yu, and X. Ding, "Cost-effective vehicle type recognition in surveillance images with deep active learning and web data," *IEEE Trans. on Intell. Transp. Syst.*, pp. 1–8, 2019.

[24] A. A. Abdellatif, C. F. Chiasserini, and F. Malandrino, "Active learning-based classification in automated connected vehicles," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2020, pp. 598–603.

[25] L. Pipino, Y. Lee, and R. Wang, "Data quality assessment," *Comm. ACM*, vol. 45, no. 4, pp. 211–218, 2002.

[26] A. Jøsang, "A logic for uncertain probabilities," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 9, no. 3, p. 279–311, 2001.

[27] J. Kang, Z. Xiong, D. Niyato, D. Ye, D. I. Kim, and J. Zhao, "Toward secure blockchain-enabled internet of vehicles: Optimizing consensus management using reputation and contract theory," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2906–2920, March 2019.

[28] X. Huang, R. Yu, J. Kang, Z. Xia, and Y. Zhang, "Software defined networking for energy harvesting internet of things," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1389–1399, June 2018.

[29] T. M.Cover and J. A.Thomas, *Elements of Information Theory*, 2nd ed. John Wiley & Sons, 2006.

[30] J. Siebert, "Vehicle recognition using rule based methods." [Online]. Available: https://www.openml.org/d/54

[31] M. Makridis, K. Mattas, and B. Ciuffo, "Traffic simulation data for Antwerp," 2017, accessed: 2020-03-30. [Online]. Available: https://doi.org/10.7910/DVN/BCUYIR

[32] "MNIST dataset." [Online]. Available: https://www.kaggle.com/oddrationale/mnist-in-csv

[33] J. Brownlee, *Master Machine Learning Algorithms: Discover How They Work and Implement Them From Scratch*, 2016. [Online]. Available: https://machinelearningmastery.com/master-machine-learning-algorithms/