Clustering of Geographical Segments for Predictive Quality of Service of Connected Vehicles

Oscar Stenhammar, Saeed Razavikia, Gábor Fodor, Carlo Fischione

***Abstract*—To meet the growing demands for connectivity and reliability in cellular networks, it is essential to ensure reliable quality of service (QoS) guarantees for end users. The integration of predictive QoS (pQoS) in cellular networks enables proactive fulfillment of QoS requirements for a diverse range of applications, including intelligent transportation systems. This study presents a pQoS framework in cellular networks, particularly for connected vehicles, that divides the road into segments, clusters them, and assigns a pQoS model to each cluster. By implementing this framework, we mitigate the concept drift of the pQoS model induced by variations in the propagation environment and interference. Each predictive cluster model is locally trained on vehicles traveling within the cluster boundaries using federated learning. A significant challenge is balancing the trade-off between the number of clusters, prediction accuracy, and communication overhead for updating local models. This trade-off suggests the novel problem of performing a joint optimization of the training and number of clusters. To address such difficult optimization, we propose an iterative approximate solution using proximal alternative minimization for which we provide convergence guarantees. Ultimately, by evaluations with real-world data, our numerical findings reveal that our proposed clustered predictive model reduces the mean absolute percentage error by 8%, and the mean absolute error by 7%, compared to conventional predictive approaches proposed by prior studies.**

I. Introduction

THERE are high demands on the development of the sixth generation (6G) in wireless communication technologies [3], [4]. Advancements in 6G wireless technology must support innovations within intelligent transportation systems to improve comfort and safety for passengers and drivers [5], [6]. These safety features necessitate exceptionally reliable communication due to the severe consequences of communication delays. Consequently, there is a critical need for network operators to ensure predefined quality of service (QoS) levels, which encompass key performance indicators (KPIs) such as data rate, latency, and probability of successful packet reception (reliability) [7]. Current strategies to meet the desired QoS in 5G networks primarily adopt a reactive approach, addressing performance degradations after occurrence [8].

There are obvious limitations of managing the network with a reactive approach. Addressing problems after they occur implies a delayed response to network issues. This can lead to the inability to guarantee consistent service quality, especially in rapidly changing environments. These drawbacks have pushed the development of predictive QoS (pQoS) to enable proactive service management in forthcoming 6G frameworks. By anticipating network conditions and adjusting resources accordingly, this proactive approach satisfies the stringent requirements of autonomous or remotely operated vehicles [9]. The network operators can use the output of pQoS algorithms to optimize network performance for maintaining trustworthy QoS for end-users. Should the required QoS levels be unachievable, service management mechanisms must promptly alert the user, enabling the implementation of precautionary safety measures [10]. These properties make pQoS particularly relevant for vehicular communication.

The main difficulty with vehicular wireless communication is the inherent time-varying stochastic characteristics of QoS processes, influenced by the dynamic wireless propagation environment. Data-driven methodologies such as machine learning (ML) have been proposed as effective tools for developing accurate pQoS models based on comprehensive datasets [10]–[12]. However, the dynamic nature of environmental conditions within an extensive geographical area can lead to concept drift of the QoS data. Concept drift refers to the phenomenon where there is a shift in the statistical properties of the underlying data for the target variable in machine learning [13]. The wireless channel propagation properties, interference, and load of the network are examples of properties that generate concept drift. This phenomenon worsens for moving vehicles, which experience rapidly varying communication conditions [11]. Concept drift degrades the model performance because the trained model becomes outdated due to changing underlying statistical properties of the data collected by vehicles. The wireless propagation environment contributes significantly to non-independent and identically distributed (IID) characteristics of data in pQoS scenarios, necessitating robust modeling techniques that can adapt to these concept drifts effectively.

©2025 IEEE. Originally published in *IEEE Transactions on Vehicular Technology* [1]. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work.

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP), funded by the Knut and Alice Wallenberg Foundation, and the Digital Future research center in KTH. G. Fodor was supported by the Swedish Strategic Research (SSF) grant for the FUS21-0004 SAICOM project. The work of C. Fischione was also funded by SSF, Digital Futures and SweWIN KTH research centers. The computations were enabled by the Berzelius resource provided by the Knut and Alice Wallenberg Foundation at the National Supercomputer Centre. *(Corresponding author: Oscar Stenhammar.)*

The authors are with the School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm, Sweden (e-mail: ostenh@kth.se, sraz@kth.se, gaborf@kth.se, carlofi@kth.se). Oscar Stenhammar and Gábor Fodor are also with Ericsson Research, Sweden (e-mail: oscar.stenhammar@ericsson.com, gabor.fodor@ericsson.com).

A highly preliminary version of the problem formulation presented in this paper has appeared in [2].

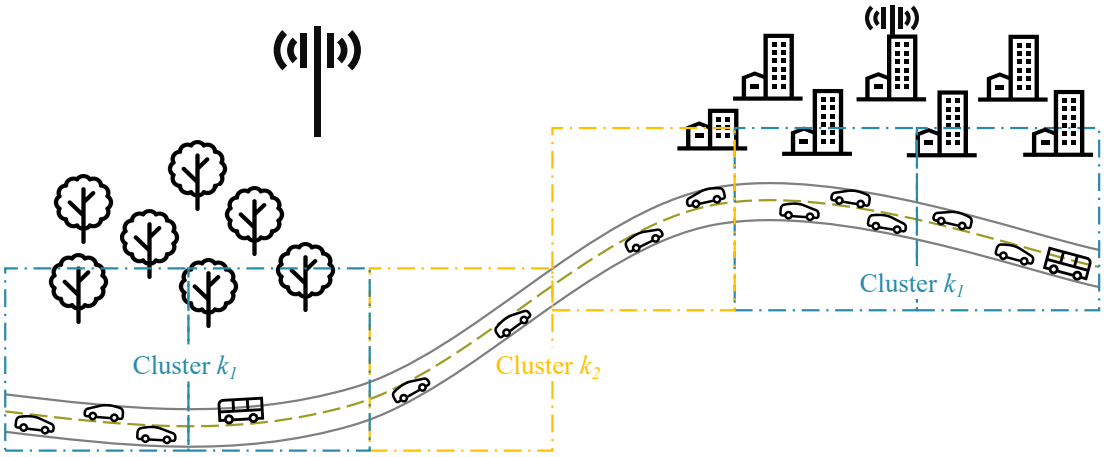


Fig. 1. An illustration of the foundational concept of our predictive framework. The geographical segments of the road are represented by the squares. Our approach proposes that segments are clustered based on statistical similarities in the QoS parameters from vehicles within the cluster. As an example, clusters k_1 and k_2 are illustrated in blue and orange respectively to visualize how segments can be clustered. To empower collaborative learning between segments within the same cluster, each cluster is assigned its own predictive model to predict the QoS.

A. Contributions

To enhance the performance of pQoS in cellular networks, this study proposes an innovative predictive framework for QoS serving connected vehicles exposed to dynamically changing environments. The central hypothesis is that a single global predictive model cannot effectively capture the non-IID characteristics inherent in radio data for pQoS. Building on this hypothesis, our proposed framework divides the road into discrete geographical segments. Segments containing data with similar statistical characteristics are clustered, which allows these segments to adopt the same predictive model. This improves the performance of both the training of such a model and its adoption for inference. Our approach results in predictive models that are personalized for geographical regions with similar underlying QoS data distributions, even though such regions might not be geographically adjacent. In order to mitigate concept drift caused by time-varying propagation environments, it is more effective to cluster geographical areas rather than clustering vehicles as done in previous works [14], [12]. A conceptual illustration in Fig. 1 demonstrates how the proposed system divides the road into square geographical segments. The system clusters segments with similar statistical characteristics of the QoS data, and each cluster is assigned a predictive model. The objective of this is to enable vehicles within the same cluster to collectively perform online training and inference using federated learning (FL).

Given a set of geographical segments, it is not trivial how to find the optimal number of clusters that minimize the accuracy of the predictive models. The main concept in this work is that with more clusters, each cluster covers a small area. A predictive cluster model covering a smaller area will achieve a better personalization of the underlying data distributions in the area that the cluster represents. The hypothesis is that this will increase prediction accuracy. However, too many clusters will result in clusters that represent too small regions to carry enough data to train an accurate ML model. Furthermore, with more clusters, there are more predictive models that need to be updated at the central server with frequent uplink and downlink communications. This implies a trade-off between the number of clusters, the prediction accuracy of the cluster models and communication efficiency.

To find optimal parameters for the predictive cluster models and the optimal number of clusters, we formulate a novel joint optimization problem. We propose a solution for the formulated problem that implements proximal alternating minimization [15]. This solution alternates between learning the optimal model parameters using FL and finding the optimal number of clusters with singular value thresholding (SVT) [16].

The overarching goal is to adapt pQoS models to variations in communication conditions by clustering different geographical regions while optimizing the number of clusters. By aggregating models from vehicles within the same cluster, the collective strength can be leveraged to improve the efficiency and accuracy of the training process through FL. The contribution of this paper can be summarized as follows.

- We propose a pQoS framework that counteracts concept drift for moving vehicles. The framework splits the global geographical region into smaller geographic segments, and clusters segments with similar underlying data distributions. A personalized predictive model is trained for each cluster. Finally, the number of clusters is optimized.
- We formulate a joint optimization problem that aims to jointly optimize the prediction error of the cluster models and the number of clusters.
- To provide a sub-optimal solution for this joint optimization problem, we adopt proximal alternating minimization. We incorporate FL to learn the cluster pQoS models and SVT to optimize the number of clusters.
- We conduct theoretical analyses to provide convergence guarantees of the proposed algorithm.
- We evaluate our pQoS framework on a measured dataset from [17] in a realistic scenario and investigate how the geographical segment size affects the end performance.

The remainder of this paper is organized as follows: Section III illustrates the system model, defines the notation for

the main variables used in the paper, and formulates the problem. Section IV describes the method of the proposed solution. Section V develops a theoretical analysis to prove convergence for the proposed method. Section VI provides numerical results to assess the proposed method performance. Finally, Section VII summarizes and concludes the paper.

II. Literature review

This section presents a literature review of the latest advancements within pQoS and FL for vehicular networks. The review is conducted to provide background on the topics and highlight research gaps that motivate our work.

A. Predictive Quality of Service

When developing pQoS models, it is essential to identify QoS requirements to ensure satisfactory communication performance [18]. In industrial Internet of Things (IoT) and vehicular applications, pQoS plays a critical role in maintaining efficient processes [10]. The study in [10] outlines required QoS levels for crucial applications such as teleoperated driving and high-density platooning, covering throughput, latency, reliability, and prediction horizons. These insights underscore the importance of precise QoS specifications in designing robust pQoS models. One of the aspects discussed in [10] is the prediction horizon, which is also considered a critical aspect in [14]. Operators and users benefit from forecasting QoS metrics over timeframes ranging from milliseconds to hours or even days. Predictive capabilities enable proactive measures, such as rerouting vehicle trajectories to avoid poor QoS conditions. These adjustments are essential for maintaining optimal performance in network-dependent applications.

Recent research interest in pQoS has grown, with a predominant focus on employing ML models. Accurate QoS forecasting is a challenging task that requires extensive datasets to train and refine ML models effectively. An extensive measurement campaign [19] collected data tailored specifically for pQoS analysis. This campaign collected data from user equipment devices (UEs) located on moving vehicles in various environmental conditions. A private network was used to capture a range of metrics from both the physical and medium access control layers. Numerous studies have used this dataset to evaluate throughput prediction models. Experimental studies in [11], [14] have shown that incorporating an extended set of input features significantly improves predictive accuracy. Among the various ML strategies tested, the gradient boosting ensemble method emerged as the most accurate approach in [11]. These findings highlight the importance of comprehensive datasets in developing effective ML models for pQoS, which is critical for advancing network quality assessment.

Many general pQoS models for vehicular applications use data from an entire dataset region to train the predictive weights. However, to develop predictive models tailored to specific environmental conditions, [12] manually divided the dataset from [19] into three environmental classes: suburban, rural, and highway. Separate random forest models were trained for each category, improving predictive performance. This indicates the presence of concept drift in the underlying QoS data between these environments. To further validate this, [11] demonstrated that concept drift exists in vehicular pQoS. To mitigate the difficulties implied by concept drift, [14] segmented the road network and constructed a probabilistic predictive model that utilizes statistical data specific to each segment. This method enhances granularity and accuracy by aligning predictions with the unique characteristics of different road environments. These studies highlight the importance of environmental specificity in developing precise pQoS models for vehicular settings.

B. Federated Learning for Vehicular Networks

FL enables decentralized learning, allowing models to be trained on data from different units, locations, or classes without requiring raw data exchange. However, scaling FL for large models in vehicular networks presents significant challenges. These include limited communication resources, mobility, and statistical heterogeneity in data distributions. To address these challenges while mitigating concept drift, [20] proposed a clustered FL architecture that facilitates learning within clusters of vehicles. Vehicles are grouped into clusters based on test dataset performance on specific cluster models. Each cluster trains models locally, transmitting only aggregated updates to a central server, optimizing communication efficiency.

Another approach to clustering users into a predetermined number of groups for joint learning was introduced in [21]. This framework, called the iterative federated clustering algorithm (IFCA), implements weight-sharing techniques for the initial layers of neural networks across clusters. IFCA is particularly beneficial when different user groups have distinct objectives, allowing for more customized FL applications. The effectiveness of collaborative learning using IFCA has been demonstrated in [22], [23], where models trained within the same cluster leverage collective learning to enhance efficiency. These studies highlight the potential of clustered FL to meet the demands of vehicular networks and improve the feasibility of large-scale FL deployments in such environments.

III. System Model and Problem Formulation

We consider a cellular network on a specified geographical region with a set of base stations. The density of connected users varies throughout the considered area, for example between a rural and an urban area, causing the interference and network load to vary. Furthermore, the area accommodates a grid of roads in varying environments for wireless channel propagation. A set of vehicles is considered, where each vehicle is at some point traveling on the roads within the considered area. The vehicles have associated applications that require communication using the cellular network to operate. To ensure that the necessary QoS level for the vehicles is feasible, each vehicle needs a reliable temporal QoS prediction over a certain time horizon in the future. We further assume that the vehicles have enough computational resources to run and train the pQoS models.

A. Process Overview

Vehicular communication is associated with high mobility. In varying environments, high mobility has a major effect on the evolution of the communication's KPI. For example, wireless channel propagation characteristics, interference, and load of the network can change drastically between an urban and rural environment. Hence, the presence of concept drift is highly probable in vehicular communication.

To combat the negative effects of concept drift on pQoS data for vehicular scenarios, we propose a pQoS framework that bounds the predictive model to a smaller geographical area. The foundational idea is that if environmental properties cause concept drift, it is better to let predictive models belong to a geographical area rather than to the vehicles. Initially, the global geographical area under consideration is segmented into smaller geographically square segments. The aim is to create a set of segments, where each segment has a uniform environment, such that concept drift cannot occur for a user within the segment. To initialize the learning framework, all segments are assigned a pre-trained multilayer perceptron (MLP) for pQoS, trained on the entire global region. Each model is then trained on data from vehicles within the geographical boundaries of the segment, using FL to learn collaboratively. Next, segments with similar properties are clustered to allow for collaborative learning. Fig. 1 graphically illustrates the concept of segments and clusters in our predictive framework. We choose the MLP as the predictive model for its adaptability and smooth integration in FL.

Segments are clustered to optimize the number of clusters iteratively until convergence is reached. By optimizing the number of clusters, the number of predictive models is decreased, which saves computational and storage resources. Fewer clusters imply clusters covering larger areas, which allows for more training data for each predictive model. With FL, a random set of vehicles performs local updates of the cluster model while inside the cluster. When a vehicle participating in the training procedure exits the boundaries of a cluster, it transmits its local update of the model in the uplink to the central server. Through the downlink, it receives the predictive model for the cluster it is entering. The local updates are aggregated periodically at the central server.

In the remainder of the paper, the variables and parameters will be referenced to according to the table of notation in Table I.

TABLE I
TABLE OF NOTATION.

Parameter	Meaning
$\mathcal{C}_k$	Cluster k
F	Loss function
h	Time horizon of prediction
K	Number of clusters
N	Number of segments
$\mathbf{O}$	Clustering matrix that determine the clusters
$\overline{\mathbf{O}}$	Clustering matrix calculated from (8)
$\boldsymbol{\theta}_k$	Parameters for predictive model k
$\boldsymbol{\theta}_k^*$	Optimal parameters for predictive model k
$\boldsymbol{\Theta}$	Matrix containing all $\boldsymbol{\theta}_k$
$\mathcal{S}_n$	Segment n
V_k	Vehicles participating training in cluster k
ω	Predictive model
$\mathbf{X}_k$	Input matrix to predictive model
$\mathbf{x}_k$	Column vector of $\mathbf{X}_k$ containing KPIs
$\mathcal{X}_k$	Dataset from cluster k
$\mathcal{X}_{\mathcal{S}_n}$	Dataset from segment $\mathcal{S}_n$

B. Problem Formulation

The main objective is to learn a parametric model for each cluster by minimizing some loss function. In each cluster, we want to learn the parametric model $\omega(\boldsymbol{\theta}_k; \mathbf{X}_k(t))$ at discrete time t with input matrix $\mathbf{X}_k(t)$, from the dataset $\mathcal{X}_k$, and parameters $\boldsymbol{\theta}_k \in \mathbb{R}^W$ for cluster k. At time t, the parametric model predicts the QoS for time $t+h$. More precisely, let $y(t)$ be the QoS KPI of interest at time t. Then, the predicted QoS $\hat{y}$ at time horizon h from current time t is given by

$$\hat{y}(t+h) = \omega(\boldsymbol{\theta}_k; \mathbf{X}_k(t)), \tag{1}$$

where

$$\mathbf{X}_k(t) = [\mathbf{x}_k(t), \ldots, \mathbf{x}_k(t-(t_w-1))], \tag{2}$$

with t_w historical measurements. Here, the column vector $\mathbf{x}_k(t)$ $n \in \mathbb{R}^P$ contains the relevant input data such as physical layer KPIs. By a slight notation abuse, the time dependency of $\mathbf{X}_k(t)$ will be omitted throughout this work for improved readability. Our aim is to find the optimal model parameters

$$\boldsymbol{\theta}_k^* = \arg\min_{\boldsymbol{\theta}_k} F(\boldsymbol{\theta}_k; \mathcal{X}_k), \quad \forall k \in [K], \tag{3}$$

that minimizes the overall loss function

$$F(\boldsymbol{\theta}_k; \mathcal{X}_k) = \frac{1}{D} \sum_{(\mathbf{X}_i, y_i) \in \mathcal{X}_k} f\Big(y_i(t+h), \omega_k(\mathbf{X}_i; \boldsymbol{\theta}_k)\Big). \tag{4}$$

Here, f is some loss function and D is the number of datapoints with input and output $(\mathbf{X}_i, y_i)$ picked from dataset $\mathcal{X}_k$ to evaluate F. By minimizing Eq. (4), the parameter $\boldsymbol{\theta}_k$ implicitly becomes a function of the prediction horizon h in $y(t+h)$.

Given that we are dealing with a distributed setting, data distribution among the clusters and the number of clusters directly impacts the learning optimization in (3). This motivates the importance of accounting for the number of clusters. More clusters for the same geographic region imply that each cluster covers a smaller geographical area. This enhances the personalization of the predictive models by fostering higher correlation among the data within each cluster. Nevertheless, more clusters lead to a reduced amount of training data per cluster, excessive memory allocation at the central server, and increased model handovers as vehicles transition between segments.

Hence, we propose jointly optimizing the learning loss function and the number of clusters K. The objective is to obtain optimal model parameters $\boldsymbol{\theta}_k^*$ for each cluster with a minimum number of clusters. Formally, we pose the following optimization problem:

$$\begin{aligned} \min_{K, \{\boldsymbol{\theta}_k\}} \quad & \lambda \cdot K + \frac{1}{K} \sum_{k=1}^{K} F(\boldsymbol{\theta}_k; \mathcal{X}_k), \\ \text{s.t.} \quad & F(\boldsymbol{\theta}_k; \mathcal{X}_k) < \epsilon, \quad \forall k, \end{aligned} \tag{5}$$

where $\lambda \in \mathbb{R}$ denotes the regularization parameter and ϵ is a positive constants $\epsilon \in \mathbb{R}^+$. The constraint in (5) is introduced to set the QoS requirements by ensuring that the loss remains below ϵ. The parameter ϵ must be determined such that there always exist model parameters $\boldsymbol{\theta}_k$ that can achieve a loss smaller than ϵ for at least one cluster k. Note that the model parameters $\boldsymbol{\theta}_k$ and the test dataset $\mathcal{X}_k$ depend on the number of clusters K. A change in the number of clusters results in the rearrangement of clusters, consequently affecting the model parameters and the test dataset for the newly formed clusters.

In conclusion, the problem in (5) presents a significant challenge as it does not appear to have been addressed in the existing literature. In this problem, the number of clusters K affects the datasets from $\mathcal{X}_k$ that are used to train and evaluate the parameters $\boldsymbol{\theta}_k$. However, it is not obvious how the relation between the number of clusters and the parameters $\boldsymbol{\theta}_k$ should be modeled. This inherent nature and complexity make it a difficult problem to solve. The problem in (5) can be re-written to

$$\min_{K,\{\boldsymbol{\theta}_k\}} \quad \lambda \cdot K + \frac{1}{K}\sum_{k=1}^{K}\frac{1}{D}\sum_{i=1}^{D} f\Big(y_i(t+h), \omega_k(\mathbf{X}_i;\boldsymbol{\theta}_k)\Big),$$

where f is a loss function sum over D datapoints with index i such that $(\mathbf{X}_i, y_i) \in \mathcal{X}_k$. The case where we use a neural network as ω_k, with non-linear activation functions, is NP-hard [24]. To show that problem (5) has a combinatorial nature and is NP hard, we consider the case where the predictive function is polynomial.

Proposition 1. *Consider the optimization problem* (5)*. Suppose the predictive function ω is polynomial with fixed degree in its parameters θ_k. Also, assume that the loss function is the squared error,*

$$f\Big(y_i(t+h), \omega_k(\mathbf{X}_i;\boldsymbol{\theta}_k)\Big) = \Big(y_i(t+h) - \omega_k(\mathbf{X}_i;\boldsymbol{\theta}_k)\Big)^2.$$

Then, the optimization problem (5) *is NP-hard.*

Proof. For any fixed partition of the dataset $\mathcal{X}_k$, the optimal θ_k for each cluster can be efficiently computed via least squares. Now, we assume that the optimal predictive parameters $\boldsymbol{\theta}_k^*$ for each cluster k are known. This simplification reduces the optimization problem to choosing the number of clusters K and partitioning the data into $\{\mathcal{X}_k\}$ data such that

$$\min_{K,\{\mathcal{X}_k\}} \quad \lambda \cdot K + \frac{1}{K}\sum_{k=1}^{K}\frac{1}{D}\sum_{i=1}^{D} \Big(y_i(t+h) - \omega_k(\mathbf{X}_i;\boldsymbol{\theta}_k^*)\Big)^2.$$

If we ignore the constant penalty term $\lambda \cdot K$ by setting $\lambda = 0$, we recover the K-means objective, which minimizes the sum of squared errors within each cluster. Since K-means clustering is NP-hard [25], this special case is NP-hard. Thus, the original problem in (5), which also involves selecting K and fitting predictive parameters for each cluster, is NP hard. □

Given these characteristics, it is clear that any solution approach unavoidably will be suboptimal. In the next section, we propose an equivalent problem re-formulation and an approximate solution approach to tackle this challenging problem, providing a promising direction for addressing this difficult and previously unexplored issue.

IV. Joint Learning and Clustering

In this section, we first give an alternative problem formulation to (5), and then we propose a suboptimal solution method.

A. Problem Reformulation

To arrive at a convenient reformulation of problem (5) and an equivalent form that will allow us to derive an adequate approximate solution, we start by giving the following definitions. Let $K_{\max}$ be the maximum possible number of clusters. First, we define the set of all cluster models as $\boldsymbol{\Theta} = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{K_{\max}}] \in \mathbb{R}^{W \times K_{\max}}$. Then, we define the clustering matrix $\overline{\mathbf{O}} \in \mathbb{R}^{N \times K_{\max}}$ as

$$\overline{\mathbf{O}} = \begin{bmatrix} \phi(\mathcal{S}_1 \in \mathcal{C}_1) & \cdots & \phi(\mathcal{S}_1 \in \mathcal{C}_{K_{\max}}) \\ \vdots & \ddots & \vdots \\ \phi(\mathcal{S}_N \in \mathcal{C}_1) & \cdots & \phi(\mathcal{S}_N \in \mathcal{C}_{K_{\max}}) \end{bmatrix}, \tag{6}$$

whose elements are defined as

$$\phi(\mathcal{S}_n \in \mathcal{C}_k) = \frac{1}{K_{\max}-1}\left(1 - \frac{F(\boldsymbol{\theta}_k; \mathcal{X}_{\mathcal{S}_n})}{\sum_{k=1}^{K_{\max}} F(\boldsymbol{\theta}_k; \mathcal{X}_{\mathcal{S}_n})}\right), \quad \forall k, n. \tag{7}$$

The function $\phi(\mathcal{S}_n \in \mathcal{C}k)$ subtracts from 1 the ratio of local and total prediction errors. The numerator is the prediction error of dataset $\mathcal{X}_{\mathcal{S}_n}$ from segment $\mathcal{S}_n$ using model $\boldsymbol{\theta}_k$ from cluster $\mathcal{C}k$. The denominator is the sum of the prediction errors of $\mathcal{X}\mathcal{S}n$ on all models $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{K_{\max}}$. By imposing that the sum of row n in $\overline{\mathbf{O}}$ has to be 1,

$$\sum\nolimits_k \phi(\mathcal{S}_n \in \mathcal{C}_k) = 1, \quad \forall n \in [N].$$

the factor $1/(K_{\max}-1)$ is included in the function ϕ to normalize the sum of each row of $\overline{\mathbf{O}}$ to 1. The dataset $\mathcal{X}_{\mathcal{S}_n}$ is a dataset from segment $\mathcal{S}_n$. The function $\phi(\mathcal{S}_n \in \mathcal{C}_k)$ returns a value in the range $[0, 1]$. Note that a choice of n and k that returns a high value of the function $\phi(\mathcal{S}_n \in \mathcal{C}_k)$, indicates that the prediction error is low. This implies that it would be beneficial if $\mathcal{S}_n \in \mathcal{C}_k$.

For the sake of notation, we use an operator $\Phi : \mathbb{R}^W \times \mathbb{R}^K \mapsto \mathbb{R}^{N \times K_{\max}}$ to present all the elements-wise operations of function ϕ in (6) and (7) as

$$\overline{\mathbf{O}} = \Phi(\boldsymbol{\Theta}, \mathcal{X}_{\mathcal{S}}), \tag{8}$$

where $\mathcal{X}_{\mathcal{S}} = \{\mathcal{X}_{\mathcal{S}_n}\}_{\forall n \in [N]}$. Given model parameters $\boldsymbol{\Theta}$ and dataset $\mathcal{X}$, we form the clustering matrix $\overline{\mathbf{O}}$ as described above. Furthermore, let $F^* > 0$ be a positive constant such that $\sum_{\hat{k}=1}^{K_{\max}} F(\boldsymbol{\theta}_{\hat{k}}; \mathcal{X}_{\mathcal{S}_n}) \geq F^*$, for $n \in [N]$. Then, to incorporate the constraints on the loss function from (5) into the clustering matrix $\overline{\mathbf{O}}$, we use the lower bound given in (5), which results in the following inequality:

$$\max_k \phi(\mathcal{S}_n \in \mathcal{C}_k) \geq \frac{1}{K_{\max}-1}\left(1 - \frac{\epsilon}{F^*}\right) =: \eta, \quad n \in [N]. \tag{9}$$

Now, by invoking the clustering matrix $\overline{\mathbf{O}}$, we can incorporate the clustering problem into the proposed regularized loss in (5). Indeed, the maximum number of clusters is equal to the number of columns in matrix $\overline{\mathbf{O}}$. Also, since the number of clusters is smaller than the number of segments, i.e., $K_{\max} < N$, the number of independent columns of matrix $\overline{\mathbf{O}}$ represents the number of clusters, and therefore $K = \text{rank}(\mathbf{O})$. By setting the number of clusters K to the rank of $\mathbf{O}$, we allow the use of tools to minimize the rank in order to optimize K.

Equipped with the definitions above, we are now ready to give an equivalent reformulation of problem (5):

$$\begin{aligned}\min_{\mathbf{O},\{\boldsymbol{\theta}_k\}} \lambda \cdot \text{rank}(\mathbf{O}) + \|\mathbf{O}-\overline{\mathbf{O}}\|_{\text{F}}^2 + \frac{1}{K}\sum_{k=1}^{K} F(\boldsymbol{\theta}_k;\mathcal{X}_k)\\ \text{s.t.} \quad \|\mathrm{e}_n^\top \mathbf{O}\|_\infty \geq \eta, \forall\, n \in [N], \quad \mathbf{O}\mathbb{1}_{K_{\max}\times 1} = \mathbb{1}_{N\times 1},\end{aligned} \tag{10}$$

where $\mathbb{1}_{N\times 1}$ stands for all one vector of size $N \times 1$, and e_n denotes the canonical vector of size N which has the value one at element n and zero elsewhere. Due to the non-convexity of the rank function, the optimization problem in (10) is still NP-hard in general [26]. However, the introduction of the rank constraint will allow for further relaxation of the optimization problem to find an algorithm to approximately solve (10). In the following subsection, we will use the nuclear norm [27], a common convex relaxation of the rank [28]. With this relaxation, we will then implement SVT to obtain an approximate solution for the number of clusters.

B. Relaxed Solution Approach

To jointly optimize the loss function and the number of clusters, we alternately minimize the loss function with respect to the model parameters $\{\boldsymbol{\theta}_k\}_{k=1}^K$ and clustering matrix $\mathbf{O}$. We first replace the rank function with its convex relaxation, i.e., the nuclear norm [27] to circumvent its NP-hardness. Specifically, the nuclear norm is the sum of all singular values. This promotes a low rank of clustering matrix $\mathbf{O}$, consequently, fewer clusters. Specifically, we relax optimization problem in (10) by the following regularized optimization problem:

$$\begin{aligned}\min_{\mathbf{O},\{\boldsymbol{\theta}_k\}} \lambda||\mathbf{O}||_* + \|\mathbf{O}-\overline{\mathbf{O}}\|_{\text{F}}^2 + \frac{1}{K}\sum_{k=1}^{K} F(\boldsymbol{\theta}_k;\mathcal{X}_k)\\ \text{s.t.} \quad \|\mathrm{e}_n^\top \mathbf{O}\|_\infty \geq \eta, \quad \mathbf{O}\mathbb{1}_{K_{\max}\times 1} = \mathbb{1}_{N\times 1}.\end{aligned} \tag{11}$$

Next, proximal alternating minimization [15] is used to obtain the optimal parameters. We use proximal alternating minimization because this algorithm has the ability to solve dual non-convex minimization problems, which applies to the general loss function F in (4). We first define the Lagrangian function below.

$$\begin{aligned}L(\boldsymbol{\Theta},\mathbf{O}) =& \lambda\|\mathbf{O}\|_* + \mathbf{1}_{\Omega_\eta}(\mathbf{O}) + \|\mathbf{O}-\overline{\mathbf{O}}\|_{\text{F}}^2\\ &+ \frac{1}{K}\sum_{k=1}^{K} F(\boldsymbol{\theta}_k;\mathcal{X}_k),\end{aligned} \tag{12}$$

where $\mathbf{1}_\Omega(\cdot)$ is the indicator function for a closed subset Ω defined by

$$\mathbf{1}_\Omega(\boldsymbol{x}) = \begin{cases} 0 & \text{if } \boldsymbol{x} \in \Omega, \\ +\infty & \text{otherwise.}\end{cases} \tag{13}$$

Moreover, Ω_η is the feasible set of constraints in (11), defined by

$$\begin{aligned}\Omega_\eta = \{\mathbf{O} \in \mathbb{R}^{N\times K_{\max}} \mid \mathbf{O}\mathbb{1}_{K_{\max}\times 1} = \mathbb{1}_{N\times 1},\\ \|\mathrm{e}_n^\top \mathbf{O}\|_\infty \geq \eta, \forall n \in [N]\}.\end{aligned} \tag{14}$$

We use the proximal alternating minimization to minimize the Lagrangian function in (12) by iterating the following:

$$\boldsymbol{\Theta}^{*(i+1)} = \arg\min_{\boldsymbol{\Theta}} \Big\{ L(\boldsymbol{\Theta}, \mathbf{O}^{*(i)}) + \frac{\alpha_i}{2}\|\boldsymbol{\Theta} - \boldsymbol{\Theta}^{*(i)}\|_{\text{F}}^2 \Big\}, \tag{15a}$$

$$\mathbf{O}^{*(i+1)} = \arg\min_{\mathbf{O}} \Big\{ L(\boldsymbol{\Theta}^{*(i+1)}, \mathbf{O}) + \frac{\beta_i}{2}\|\mathbf{O} - \mathbf{O}^{*(i)}\|_{\text{F}}^2 \Big\}, \tag{15b}$$

where i refers to the i^{th} iteration, $\boldsymbol{\Theta}^{*(i+1)}$ is the optimal solution for (15a), $\mathbf{O}^{*(i+1)}$ is the optimal solution for (15b) and β_i, α_i are the stepsizes at iteration i. We show in Section VIII that this iterative procedure in (15) converges to a critical point of the Lagrangian function $L(\boldsymbol{\Theta}, \mathbf{O})$ under some mild condition for the loss function $F(\boldsymbol{\theta}_k, \mathcal{X}_k)$. In the following, we provide the updated rules for each step.

1) Updating the Learning Model: Given clustering matrix $\mathbf{O}$, we first update the learning parameters based on (15a) in a distributed fashion. Indeed, at iteration i, by substituting the definition of the Lagrangian function $L(\boldsymbol{\Theta}, \mathbf{O}^{(i)})$ from (12) into (15a), we obtain

$$\boldsymbol{\Theta}^{(i+1)} = \arg\min_{\boldsymbol{\Theta}} \frac{1}{K}\sum_{k=1}^{K} F(\boldsymbol{\theta}_k;\mathcal{X}_k) + \frac{\alpha_i}{2}\sum_{k=1}^{K}\|\boldsymbol{\theta}_k - \boldsymbol{\theta}_k^{(i)}\|^2. \tag{16}$$

The minimization of (16) is obtained through iterative gradient descent. However, since the data is distributed across multiple clusters, we need to apply the distributed gradient descent. In a bid to ensure privacy, and reduce the communication costs between servers and clusters, we use the FedAVG scheme [29] to obtain the minimizer of (16).

In this regard, let us describe one round of the standard FedAVG scheme here. Specifically, the central server broadcasts the latest model $\boldsymbol{\theta}_{k,v}^{(i,j)} = \boldsymbol{\theta}_k^{(i,j)}$ to all participating vehicles $v \in [V_k]$ in $k \in [K]$ in cluster k, at iteration i and communication round $j = 0, ..., T_1$. Every participating vehicle performs E local epochs as

$$\boldsymbol{\theta}_{k,v}^{(i,j)} = \boldsymbol{\theta}_{k,v}^{(i,j)} - \gamma_i^j \nabla F(\boldsymbol{\theta}_{k,v}^{(i,j)}; \mathcal{X}_{k,v}^{(i,j)}),\ k \in [K]. \tag{17}$$

Here, $\mathcal{X}_{k,v}^{(i,j)}$ is a dataset from vehicle $v \in [V_k]$ at iteration (i,j), and γ_i^j is the local step size $\gamma_i^0 = \alpha_i$. After every round j, the vehicles transmit the parameter updates $\boldsymbol{\theta}_{k,1}^{(i,j)}, \ldots, \boldsymbol{\theta}_{k,V_k}^{(i,j)}$

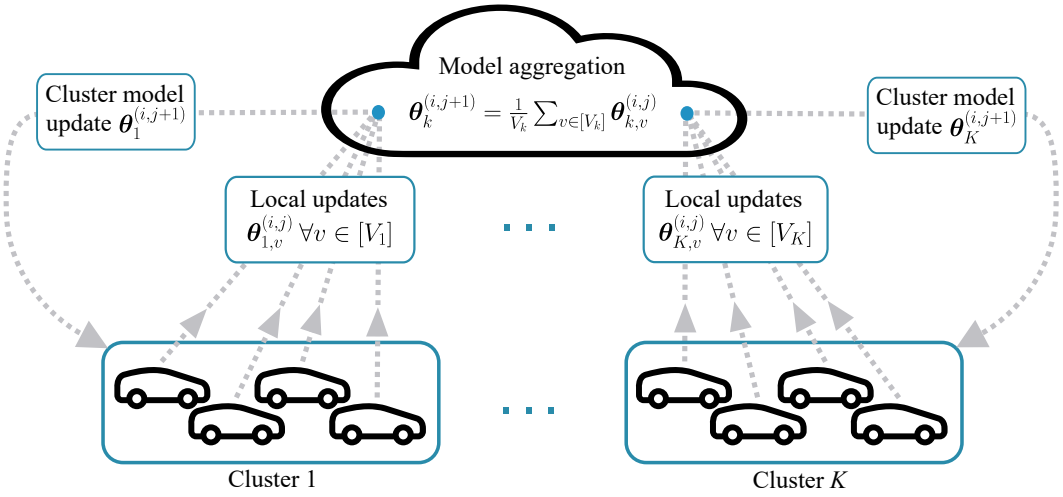


Fig. 2. This scheme represents how cluster models are trained. As vehicle v enters a cluster, it receives the model parameters $\boldsymbol{\theta}_{k,v}^{(i,j)}$. While the vehicle is located within the boundaries of cluster k, it collects data $\mathcal{X}_{k,v}^{(i,j)}$ and accordingly performs local updates on the model parameters. When the vehicle exits the cluster, the model parameters from the local updates are transmitted to the central server. After such a communication round, the central server aggregates the collected model updates. New vehicles entering the cluster now receive the model parameters $\boldsymbol{\theta}_{k,v}^{(i,j+1)}$. This iteration continues for $j = T_1$ iterations. At that point, the first iteration of the predictive models is complete, and we set $\boldsymbol{\theta}_k^{(i+1)} = \boldsymbol{\theta}_k^{(i,T_1)}$ for the models in each cluster k. To proceed, the proximal alternating minimization initiates the clustering process.

via the uplink to the central server, where the new set of cluster models $\boldsymbol{\theta}_k^{(i,j+1)}$ are calculated as

$$\boldsymbol{\theta}_k^{(i,j+1)} = \frac{1}{V_k}\sum_{v\in[V_k]}\boldsymbol{\theta}_{k,v}^{(i,j)}. \tag{18}$$

This procedure is illustrated in Fig. 2. The updating of the parameters $\boldsymbol{\theta}_k^{(i+1,j)}$ continues for $j = T_1$ communication rounds, at which point we set $\boldsymbol{\theta}_k^{(i+1)} = \boldsymbol{\theta}_k^{(i,T_1)}$. Finally, the approximate optimal parameters are given by $\boldsymbol{\Theta}^{(i+1)} = \left[\boldsymbol{\theta}_1^{(i+1)}, \ldots, \boldsymbol{\theta}_K^{(i+1)}\right]$.

2) Updating the Clustering Matrix: Given the model parameters $\boldsymbol{\Theta}^{(i+1)}$ and using the operator Φ, the central server obtain the clustering matrix at iteration i as

$$\overline{\mathbf{O}}^{(i)} := \Phi(\boldsymbol{\Theta}^{(i+1)}, \mathcal{X}_{\mathcal{S}}). \tag{19}$$

At his point, we incorporate $\boldsymbol{\Theta}^{(i+1)}$ from the previous step i and input (12), (14) into (15b), which leads to the optimization below:

$$\mathbf{O}^{(i+1)} = \arg\min_{\mathbf{O}}\ \lambda\|\mathbf{O}\|_* + \frac{\beta_i}{2}\|\mathbf{O}-\mathbf{O}^{(i)}\|_{\mathrm{F}}^2 + \|\mathbf{O}-\overline{\mathbf{O}}^{(i)}\|_{\mathrm{F}}^2 + \mathbf{1}_{\Omega_\eta}(\mathbf{O}). \tag{20}$$

We use the SVT algorithm introduced in [16] to obtain the optimal clustering matrix. More precisely, let $\mathbf{O}^{(i,l)}$ be the clustering matrix at iteration $l = 0, \ldots, T_2$ of the SVT, in which the initial value is obtained by $\mathbf{O}^{(i,0)} = \mathbf{O}^{(i)}$. Then, for matrix $\mathbf{Y}$ of size $N \times K_{\max}$, we employ the following iterative sequence at the central server to obtain a low-rank solution to (20).

$$\mathbf{Y}^{(t)} = \mathcal{D}_{\beta_i,\rho}(\mathbf{O}^{(i,l)} + \frac{2}{\beta_i}\overline{\mathbf{O}}^{(i)}), \tag{21a}$$

$$\mathbf{O}^{(i,l+1)} = \mathcal{P}_{\Omega_\eta}(\mathbf{Y}^{(t)}), \tag{21b}$$

where $\mathcal{D}_{\beta_i,\rho}(\cdot)$ is the soft-thresholding operator at level ρ, which is defined by

$$\mathbf{Y}^{(t)} = \mathcal{D}_{\beta_i,\rho}(\mathbf{O}^{(i,l)} + \frac{2}{\beta_i}\overline{\mathbf{O}}^{(i)}) := \mathbf{U}(\boldsymbol{\Sigma} - \frac{\rho}{\beta_i}I)_+\mathbf{V}^{\mathsf{H}}, \tag{22}$$

the operation $a_+ = \max\{0, a\}$ is applied element-wise to the matrix, and $\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathsf{H}}$ represents the singular value decomposition of matrix $\mathbf{O}^{(i,l)} + {}^2\!/_{\beta_i}\overline{\mathbf{O}}^{(i)}$. Moreover, $\mathcal{P}_{\Omega_\eta}(\mathbf{Y}^{(t)})$ is the projection operator that maps the matrix $\mathbf{Y}^{(t)}$ onto the feasible set Ω_η. We perform this entire procedure for T_2 iterations to obtain the optimal low rank clustering matrix by $\mathbf{O}^{(i+1)} = \mathbf{O}^{(i,T_2)}$. At this point, $K = \|\mathbf{O}^{(i+1)}\|_*$ and $\mathcal{S}_n \in \mathcal{C}_{k_n^*}$ where $k_n^* = \arg\max_k\ [\mathbf{O}^{(i+1)}]_n$ maps the cluster affiliation of segment n. Here, n is the row index and k is the column index of matrix $\mathbf{O}^{(i+1)}$.

The overall procedure is summarized in Algorithm 1. In the next section, we provide the theoretical analysis regarding the convergence of Algorithm 1.

V. Theoretical Analysis

In this section, we provide theoretical analyses of the proposed joint learning and clustering in Algorithm 1. The convergence of the framework is analyzed and the computational and communication complexity is presented.

A. Convergence Overview

Algorithm convergence refers to the process by which an iterative algorithm gradually leads the algorithm to a stable solution. After a certain number of iterations, further iterations yield minimal changes and the improvement in the objective function becomes negligibly small. This indicates that the algorithm has effectively settled into an optimal or near-optimal state.

Algorithm 1 Pseudo-code of joint learning and clustering.

Input: segments $n \in [N]$, test dataset $\mathcal{X}_n$, clusters $k \in [K]$ initial cluster model parameters $\hat{\theta}_k$ number of parallel update rounds T
$k_n^* = \text{argmin}_{k\in[K]} F(\hat{\theta}_k, \mathcal{X}_n), \quad \forall n \in [N]$
for $i = 0, 1, ..., I-1$ **do**
  for $k \in [K]$ in parallel at $j = 0, ..., T_1 - 1$ **do**
    $\mathcal{V}_k \leftarrow$ random set of vehicles $\{v_n\}$ in segments n such that $k = k_n^*$
    $V_k \leftarrow |\mathcal{V}_k|$
    transmit $\boldsymbol{\theta}_{k,v}^{(i,j)} \leftarrow \boldsymbol{\theta}_k^{(i,j)}$ to vehicles $\mathcal{V}_k$
    for vehicle $v = 1, ..., V_K$ in parallel **do**
      $\boldsymbol{\theta}_{k,v}^{(i,j)} = \boldsymbol{\theta}_{k,v}^{(i,j)} - \gamma_i^t \nabla F(\boldsymbol{\theta}_{k,v}^{(i,j)}; \mathcal{X}_{k,v}^{(i,j)})$
    end for
    transmit $\boldsymbol{\theta}_{k,v}^{(i,j)}$ to central server
    $\boldsymbol{\theta}_k^{(i,j+1)} = \frac{1}{V_k} \sum_{v=1}^{V_k} \boldsymbol{\theta}_{k,v}^{(i,j)}$
  end for
  $\boldsymbol{\theta}_k^{(i+1)} \leftarrow \boldsymbol{\theta}_k^{(i,T_1)}$
  $\overline{\mathbf{O}}^{(i)} = \Phi(\boldsymbol{\Theta}^{(i+1)}, \mathcal{X}_{\mathcal{S}})$
  $\boldsymbol{\Theta}^{(i+1)} = [\boldsymbol{\theta}_1^{(i+1)}, \ldots, \boldsymbol{\theta}_K^{(i+1)}]$
  for $l = 0, ..., T_2 - 1$ **do**
    $\mathbf{Y}^{(t)} = \mathcal{D}_{\beta_i,\rho}(\mathbf{O}^{(i,l)} + \frac{2}{\beta_i}\overline{\mathbf{O}}^{(i)})$
    $\mathbf{O}^{(i,l+1)} = \mathcal{P}_{\Omega_\eta}(\mathbf{Y}^{(t)})$
  end for
  $\mathbf{O}^{(i+1)} \leftarrow \mathbf{O}^{(i,T_2)}$
  $K \leftarrow \|\mathbf{O}^{(i+1)}\|_*$
  $k_n^* \leftarrow \arg\max_k [\mathbf{O}^{(i+1)}]_n, \quad \forall n \in [N]$
end for
Output: K, $\{k_n^*\}$, $\boldsymbol{\Theta}^{(I)}$

From a practical standpoint, convergence has a direct impact on system performance. Algorithm convergence establishes the stability of the algorithm. A converging algorithm can reduce the frequency of model updates, which is especially beneficial in settings like FL, where communication overhead is a concern. When convergence is achieved, the models can be updated less frequently without compromising performance, minimizing communication overhead and computational costs.

B. Convergence Analysis

In this section, we prove the convergence of the proposed joint learning and clustering in Algorithm 1. To start with, we need to make the necessary assumptions for our proposed algorithm to converge. The assumptions relate to the loss function and the gradients and are standard in the literature [30], [31].

Assumption 1. ***(Bounded Loss Function)*** *There exists a constant F^* such that $F(\boldsymbol{\theta}; \mathcal{X}) \geq F^*$ for all $\boldsymbol{\theta}$ and $\mathcal{X}$.*

Assumption 2. ***(L-Smoothness)*** *We say that loss function $F(\boldsymbol{\theta}; \mathcal{X})$ is L-smooth for all $\boldsymbol{\theta}$ and $\boldsymbol{w}$ if*

$$F_k(\boldsymbol{\theta}) \leq F_k(\boldsymbol{w}) + (\boldsymbol{\theta} - \boldsymbol{w})^\top \nabla F_k(\boldsymbol{\theta}) + \frac{L}{2}\|\boldsymbol{\theta} - \boldsymbol{w}\|^2, \quad (23)$$

where $F_k(\boldsymbol{\theta}) := F(\boldsymbol{\theta}; \mathcal{X}_k)$ for $k \in [K]$.

Assumption 3. ***(Strongly Convex)*** *Loss function $F(\boldsymbol{\theta}; \mathcal{X})$ is μ-strongly convex for all $\boldsymbol{\theta}$ and $\boldsymbol{w}$ if*

$$F_k(\boldsymbol{\theta}) \geq F_k(\boldsymbol{w}) + (\boldsymbol{\theta} - \boldsymbol{w})^\top \nabla F_k(\boldsymbol{\theta}) + \frac{\mu}{2}\|\boldsymbol{\theta} - \boldsymbol{w}\|^2, \quad (24)$$

where $F_k(\boldsymbol{\theta}) := F(\boldsymbol{\theta}; \mathcal{X}_k)$ for $k \in [K]$.

Assumption 4. ***(Gradient Divergence)*** *Let $\overline{\nabla F}_k(\boldsymbol{\theta}) := \mathbb{E}\|\nabla F(\boldsymbol{\theta}^{(i,j)}; \chi_k^{(i,j)})\|$, where expectation is over the distribution of the dataset $\mathcal{X}_k$. Then, the variance of stochastic gradients in each cluster is bounded $\mathbb{E}\|\nabla F(\boldsymbol{\theta}; \chi_k^{(i,j)}) - \overline{\nabla F}_k(\boldsymbol{\theta})\|^2 \leq \sigma_k^2$ for $k \in [K]$.*

Assumption 5. ***(Bounded Variance)*** *The expected squared norm of stochastic gradients is uniformly bounded over all the communication rounds and updates, i.e., $\mathbb{E}\|\nabla F(\boldsymbol{\theta}^{(i,j)}; \chi_k^{(i,j)})\|^2 \leq G^2$ for $k \in [K]$, and $j \in [T_1]$.*

Recall that these are standard assumptions in the ML literature [30], [31]. Now we are ready to present the main result. In the following proposition, we prove that Algorithm 1 converges to a critical point of the Lagrangian function (12) under Assumptions 1-5.

Proposition 2. *Let $(\boldsymbol{\Theta}^{(i)}, \mathbf{O}^{(i)})_{i=1}^{\infty}$ be the sequence generated by Algorithm 1 and step size α_i and β_i be selected such that $\sum_{i=1}^{\infty} \alpha_i \leq \infty$ and $\sum_{i=1}^{\infty} \beta_i \leq \infty$. Under Assumptions 1-5, the sequence asymptotically converges to a critical point of the Lagrangian function $L(\mathbf{O}, \boldsymbol{\Theta})$, i.e., for $\tilde{\boldsymbol{\Theta}} = \lim_{i\to\infty} \boldsymbol{\Theta}^{(i)}$ and $\tilde{\mathbf{O}} = \lim_{i\to\infty} \mathbf{O}^{(i)}$, we have*

$$\partial L(\tilde{\boldsymbol{\Theta}}, \tilde{\mathbf{O}}) = 0, \quad (25)$$

with probability of at least $1 - \sum_{i=1}^{\infty} \frac{e^{-\tilde{c}_3 \alpha_i^2}}{i^2}$.

Proof. The proof of Proposition 2 is provided in Section VIII. □

In Section VI, we empirically assess the performance of the proposed Algorithm 1 by providing comprehensive numerical experiments.

C. Computational Complexity Analysis

In this subsection, we analyze the computational complexity of Algorithm 1, which integrates proximal alternating minimization, SVT, and FedAVG. Algorithm 1 implements FedAVG across K different clusters, each containing multiple vehicles that perform local updates. The vehicles compute local updates based on their collected data, with the computational complexity for local updates on each vehicle being $O(d|\boldsymbol{\theta'}|)$. Here, d is the number of data points in each client, and $|\boldsymbol{\theta'}|$ is the length of the the model parameters vector $\boldsymbol{\theta'}$, transmitted to the central server. Here, we abuse the notation and exclude the indices for iteration (i, j), cluster k, and vehicle v for clarity, because they are not important in this context. The computational complexity for local updates in one vehicle is

$$O(IT_1 Ed|\boldsymbol{\theta'}|).$$

Here, I is the number of iterations in the proximal alternating minimization, T_1 is the number of iterations for the FedAVG and E is the number of local epochs.

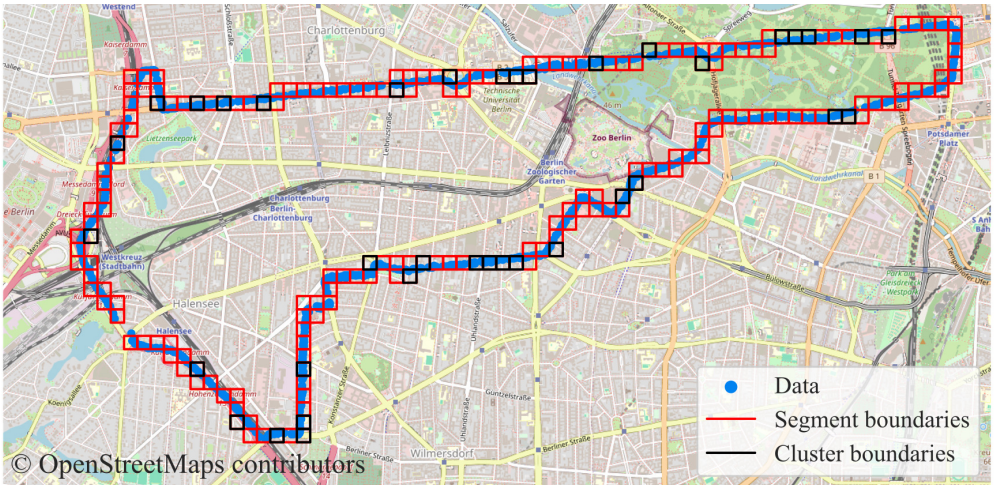


Fig. 3. A map of the area where the data in [17] were collected. The exact locations of where the data was collected are marked with blue dots, creating the trajectory of the loop that the vehicles were driving in. The selected segments are illustrated by the squares, 100 by 100 meters large. Additionally, one example of a cluster is attached for illustrative purposes, marked by the black squares.

Each vehicle then sends its local updates to the central server, which aggregates the updates from all vehicles in each cluster. The number of vehicles, or clients, participating in the training process in cluster k at iteration i of the proximal alternating minimiation is expressed by $V_{(k,i)}$. The aggregation step at the central server involves summing the updates from all vehicles, which yields a complexity of

$$O(T_1|\boldsymbol{\theta'}|\sum_{i=0}^{I}\sum_{k=1}^{K_i}V_{(k,i)}).$$

The matrix in (22) on which SVT is performed has dimensions $N\times K$, where $K<N$. The SVT algorithm addresses the proximal operator in the alternating minimization framework, where the primary computational step includes singular value decomposition (SVD), thresholding, and reconstruction. The computational complexity of performing SVD on an $N\times K$ matrix is $O(NK^2)$, given that $K<N$. The thresholding step is $O(K)$, and the reconstruction step is $O(NK^2)$. Therefore, the overall complexity of SVT is dominated by the SVD step, resulting in $O(NK^2)$. This clustering process is repeated for IT_2 times, where T_2 is the number of clustering iterations.

Combining the complexities of SVT and FedAVG, the overall computational complexity of the algorithm at the central server can be summarized as

$$O(IT_2NK_{\max}^2+T_1|\boldsymbol{\theta'}|\sum_{i=0}^{I}\sum_{k=1}^{K_i}V_{(k,i)}), \tag{26}$$

with the aggregation step being less computationally intensive.

D. Communication Complexity Analysis

In this subsection, we analyze the communication cost of Algorithm 1. We multiply the number of parameters $|\boldsymbol{\theta'}|$ by the bits per element b to calculate bits transmitted per link

$$b|\boldsymbol{\theta'}|.$$

We communicate the model parameters $\boldsymbol{\theta'}$ in the uplink to aggregate the K_i cluster models at the central server, on the proximal alternating minimization iteration i. The aggregated K_i cluster models are then transmitted to the clients $V_{(k,i)}$ for T_1 training iterations and I iterations of proximal alternating minimization. This leads to one transmission in the uplink for each client $V_{(k,i)}$, and one transmission in the downlink for each cluster K_i. Here, $V_{(k,i)}$ represents the number of participating clients in the training procedure of the predictive model in cluster k at iteration i. This leads to the total communication cost

$$\left(K_i+\sum_{k=1}^{K_i}V_{(k,i)}\right)T_1b|\boldsymbol{\theta'}|, \tag{27}$$

at iteration i. The sum $\sum_{k=1}^{K_i}V_{(k,i)}$ is relatively constant as vehicles enter and exit the cluster. However, notice that as the number of clusters K_i decreases with iteration i, the communication cost decreases.

The Shannon-Hartley theorem can be used to evaluate the computational complexity as

$$C=\frac{1}{L}\left(K_i+\sum_{k=1}^{K_i}V_{(k,i)}\right)T_1b|\boldsymbol{\theta'}|=B\log_2(1+\text{SNR}), \tag{28}$$

where C is the channel capacity in bits per second, L is the latency, B is the bandwidth and SNR is the signal-to-noise ratio. With the expression in (28), one can investigate how the number of clusters affects the relationship between the bandwidth and the latency for a known SNR. As K_i decreases with the iterations i, the communication complexity will decrease for each until convergence is reached.

VI. Numerical Evaluation

To evaluate the efficiency of our proposed predictive framework, we conduct experiments with real-world measurements. This section first describes the environment of our numerical evaluation and then presents the results.

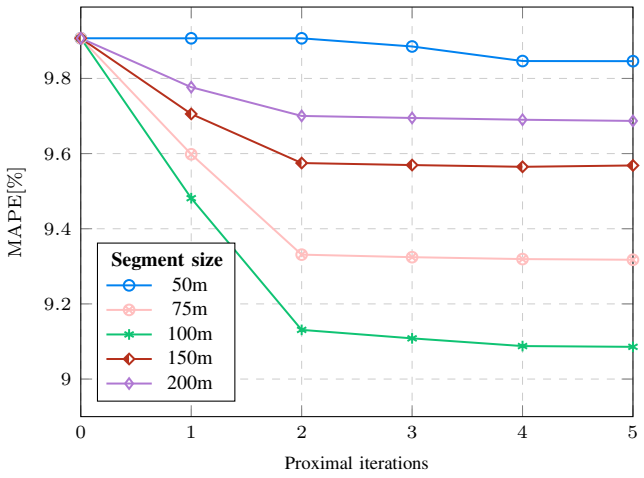


Fig. 4. The mean absolute percentage error (MAPE) as a function of the proximal iterations. The zeroth cluster iteration indicates the MAPE of the global predictive model, obtained prior to segmentation. Five parallel experiments with prediction horizon the graph of $h = 1$s are conducted with different segment sizes, labeled in the legend. Each proximal iteration indicates a complete iteration of proximal alternating minimization. Among the five different sizes, the segment size of 100 meters learns to predict QoS most effectively while achieving convergence. The approaches using other segment sizes also converge but do not achieve as low MAPE as the approach with 100 meters segment size.

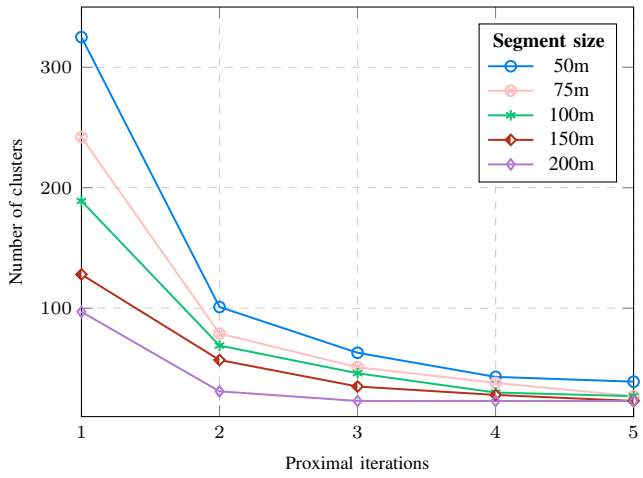


Fig. 5. The number of clusters plotted as a function of the proximal iterations, from the results of the same experiment as plotted in Fig. 4. Five parallel experiments with prediction horizon $h = 1$s are conducted with different segment sizes, labeled in the legend. Each proximal iteration indicates a complete iteration of proximal alternating minimization. The plot indicates that the number of clusters converges with the iterations of our solution.

A. Experimental Environment

To assess the efficacy of our proposed pQoS framework, we conduct an empirical study. We focus on the comparative analysis of the performance between the proposed clustered predictive models and a single global predictive model by leveraging the collected data from [17]. The collection of this dataset has been carried out using user equipments (UEs), connected to a public network, positioned in four vehicles. The vehicles were navigating on a 17km circuit within central Berlin depicted in Fig. 3, encompassing diverse environments such as a highway, tunnel, park, and residential area. Concurrently, global positioning system (GPS) data detailing position and speed have been recorded. All data has been sampled at a frequency of one second. The data acquisition spans multiple parameters, including physical layer attributes and radio resource control metrics during uplink and downlink transmissions in numerous cells across two different public network operators. To avoid different underlying data distributions affected by unknown information, such as antenna placement, etc., we focus our experiments on the operator referred to as operator 1 in the dataset. An alternative approach to handling two different operators would include the operator ID as an input feature to the prediction algorithm. However, in this experiment, we focus on providing a proof of concept of our predictive algorithm and simply chose to study operator 1.

The experiment is conducted on an NVIDIA A100 GPU cluster. The computational environment includes 8 GPUs with 40GB VRAM, and 2 CPUs of the model AMD Epyc 7742 with 1TB RAM. The models are implemented using TensorFlow with CUDA 12.1.1 for GPU acceleration. Training of the neural networks is performed with a learning rate of 10^{-3}, batch size of 8, for 30 epochs. Regarding the dataset, 80% of the data within each segment is used for training and 20% is used for testing.

In our experimental evaluation, we focus on throughput as the measure of QoS, with the objective of predicting downlink throughput over a time horizon of h seconds. The input data features, denoted as $\mathcal{X}(t)$, include physical layer metrics, throughput, position, hour of the day, and speed of the vehicle from five consecutive historical samples. Using these input features allows the models to capture differences in vehicular traffic behavior which might affect mobile traffic patterns across different periods. For example, the differences between hours with low traffic and rush hours. Our primary task is to determine the optimal parameters $\boldsymbol{\theta}_k$, $\forall k \in [K]$ for the predictive function, as described in (1).

An MLP with three hidden layers of 256, 256, and 128 neurons is incorporated for the pQoS model ω and the loss function $f(y, \hat{y})$ in (4) is selected as the mean squared error (MSE). To initiate the predictive framework, the global predictive model $\boldsymbol{\theta}$ is trained on data from the entire area in Fig. 3. Next, the global geographical region of the dataset is split into square segments, marked by squares in Fig. 3. At this point, all segments are supplied with the global predictive model, creating one cluster per segment such that $K_{\max} = N$ with cluster models $\boldsymbol{\theta}_k = \boldsymbol{\theta}$. Here, our predictive framework is implemented to train the predictive models $\boldsymbol{\theta}_k$ and optimize the number of clusters K.

To lower the communication cost presented in Eq. (27) and the computational cost in Eq. (26), we implement transfer learning [32], meaning that we only update the last layer in the predictive model. This means that we only need to transmit and train the last layer of the MLP, which lowers the complexities by lowering $|\boldsymbol{\theta}'|$. In this way, we keep the general QoS dynamics learned by the global model but personalize the predictive to each cluster by training the last layer of the MLP.

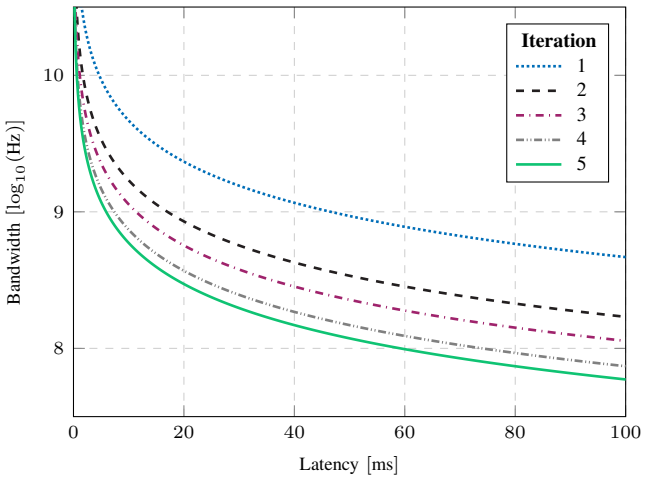


Fig. 6. The bandwidth in logarithmic scale as a function of the latency in milliseconds. The plot is based on (28). It illustrates the communication cost based on the number of clusters obtained for each proximal iteration plotted in Fig. 5 for the segment size of 100 meters. Moreover, we set the SNR $= 10$ dB to obtain this result. The plot illustrates how the communication cost decreases with the number of proximal iterations, and motivates to optimize the number of clusters.

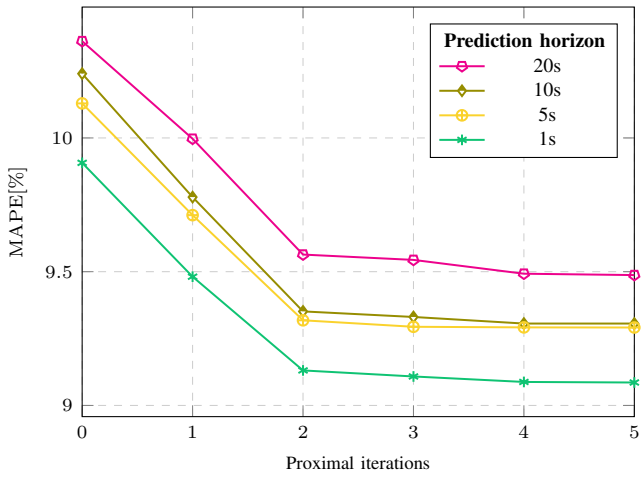


Fig. 7. A plot of how the MAPE converges with the proximal iterations for different prediction horizons, using the segment size of 100 meters. Specifically, the graph of $h = 1$s is the same graph as in Fig. 4 with segment size of 100 meters. The plot demonstrates how the MAPE increases with the prediction horizon, indicating that the autocorrelation of the QoS decreases with increasing prediction horizon h.

B. Results

To investigate how the end performance of our framework is affected by the size of the geographical segments, we run five parallel experiments with 50, 75, 100, 150, and 200 meters as sizes of the square segments. The performance of the framework depending on the segment size is compared numerically in Fig. 4 using a prediction horizon of $h = 1$ second. To evaluate the performance of the pQoS model in this figure, we use the mean absolute percentage error (MAPE) accuracy metric as a function of the number of proximal alternating minimization iterations i.

The convergence of our proposed algorithm is verified through Fig. 4. The zeroth iteration represents the performance of the global model, which uses data from the global area to train and perform inference. The MAPE of the algorithm using a segment size of 100 meters decreases rapidly during the first two proximal iterations. After the second iteration, the algorithm starts to converge and the MAPE stabilizes for the upcoming proximal iterations. The pQoS framework finally scores a MAPE of 9.09%. Fig. 3 includes one example of an identified cluster of segments, marked by black squares, at the last iteration with a segment size of 100 meters.

TABLE II
A TABLE OF THE AVERAGE NUMBER OF DATA POINTS IN EACH CLUSTER AT THE LAST PROXIMAL ITERATION FOR THE EXPERIMENTS WITH DIFFERENT SEGMENT SIZES DEPICTED IN FIG. 4 AND FIG. 5.

Segment size [m]	50	75	100	150	200
Data points	1685	3009	3170	3969	5461

The implementation using the other segment sizes of 50, 75, 150 and 200 meters also converge. However, they converge to a higher MAPE. This illustrates how the performance of the proposed algorithm is affected by the size of the geographical segments. Our interpretation of the results is that with too small segments, there is not enough training data to represent the underlying distribution of the data within the segment. As the segment size increases, the number of data points in the cluster increases. This realization can be made by observing Table II. The number of data points affects both how a segment is clustered and how well the training procedure can learn from the acquired data. However, if the size of the geographical segment is too large, we believe that concept drift is present within the segment. As the presence of concept drift in QoS data in vehicular cases has been demonstrated in [11], it is a reasonable explanation. This emphasizes the importance of finding an adequate size of the segment.

In a restricted global region, the number of clusters is also affected by the segment size. The number of clusters is illustrated as a function of the proximal iterations in Fig. 5. The initial number of clusters is indeed affected by the segment size. However, we can see that the number of clusters converges with the proximal iterations. After five iterations, the numbers of clusters are 23, 27, and 39. We can verify with Fig. 4 and Fig. 5 that our proposed framework for pQoS establishes joint convergence on prediction error and number of clusters.

To visualize how the optimization of the number of clusters affects the downlink communication cost, we use the expression derived in (28). The sum $\sum_{k=1}^{K_i} V_{(k,i)}$, corresponding to the uplink communication cost, is neglected since it can be seen as a constant over the iterations. We plot the bandwidth as a function of the latency for SNR $= 10$dB in Fig. 6. The results are plotted for the number of clusters K_i at iteration i. We use the number of clusters K_i obtained in Fig. 5 for a segment size of 100 meters. The figure demonstrates how the bandwidth decreases with the proximal iterations, for a specific latency and SNR. Since $K_1 = 189$ and $K_5 = 27$, the communication cost after iteration $i = 5$ is $K_1/K_5 = 7$ times cheaper than after iteration $i = 1$. This motivates the optimization of the number of clusters. If we observe Fig. 4, it is obvious that

the prediction error is kept relatively constant after proximal iteration $i = 2$. By only observing the prediction error, it is not clear why optimizing the number of clusters is useful. However, by realizing that the communication cost is decreased by a factor of 7, it is clear that optimizing the number of clusters is beneficial.

The behavior of our proposed algorithm is also examined over several different prediction horizons $h = [1, 5, 10, 20]$ seconds for the segment size of 100 meters. The results are plotted in Fig. 7. This plot indicates that the prediction accuracy decreases as the prediction horizon increases. However, our proposed method still converges.

To further investigate the performance of the proposed pQoS framework, we continue the numerical evaluation by comparing our framework with some benchmarks for pQoS. First, we implement the most simple predictor, which assumes that the future throughput to predict is equal to the throughput at the current time. We will refer to this approach as the naïve predictor. The other benchmark we consider is the global predictive model, using data from the entire geographical area to train and perform inference. We use the global predictive model trained before constructing the segments, which we already have from before. We will refer to this approach as the global predictor. Global predictors using MLPs have been used extensively in previous works, as discussed in Section II. Our proposed model is referred to as the clustered model.

To provide a meaningful benchmark, we also include a comparison using k-means clustering. In this predictive approach, segments are clustered using k-means in the same number of clusters as determined by our best performing predictor using a segment size of 100 meters. That is 27 clusters. Then, we create 27 predictive MLPs, identical to the ones used for our clustered predictive model. These MLPs are also trained locally on vehicles in each cluster and aggregates using FL, just as our clustered predictive approach.

We compare the performance of our proposed pQoS algorithm, the clustered predictor, with the naïve predictor, global predictor, and k-means predictor by Fig. 8. The plot presents the cumulative distribution function (CDF) of the absolute percentage error for the different predictors. The naïve predictor has the lowest performance, which is expected and reassuring because it implies that predictive efforts are justified. The CDF of the global predictor is slightly lower than the clustered predictor, meaning that it is outperformed by the clustered predictor. The k-means prediction algorithm performs quite similar to the global model. However, the CDF reveals that the clustered predictive model is more likely to provide predictions with lower absolute percentage error. Additional numerical results are provided in Table III including the MAPE and mean absolute error (MAE) as performance metrics. The results indicate an advantage for the proposed clustered predictive framework. Our approach outperforms the k-means predictive method, which is quite close to the global predictor in performance. More specifically, our clustered predictive model reduces the MAPE by 8%, and the MAE by 7%, compared to the global predictive approach from previous studies. Ultimately, all numerical results demonstrate the advantage of our proposed clustered predictive framework under concept drift.

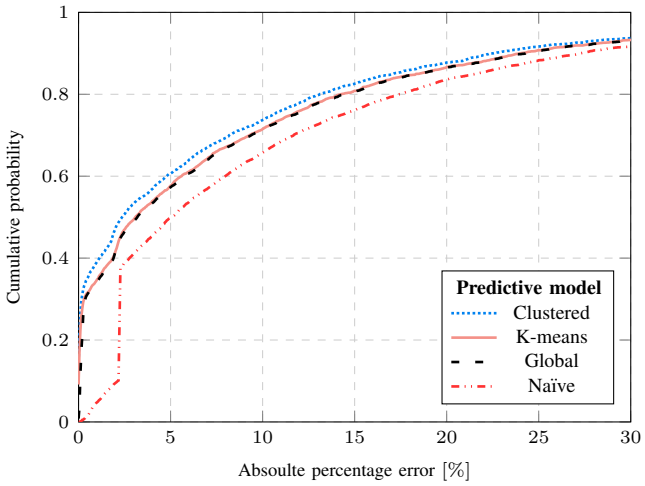


Fig. 8. A plot of the (cumulative distribution function) CDF of the absolute percentage error to compare our proposed clustered predictive algorithm, the global predictor, and the naïve predictor. In this comparison, the prediction horizon is set to $h = 1$s, and the segment size for the clustered approach is 100 meters. The naïve predictor, assuming the current throughput is equal to the future throughput, obtains the worst performance. The performance of the k-means algorithm is slightly better than the global predictive model. However, the clustered predictive algorithm has superior performance compared to the global and k-means predictors since it has a higher probability of achieving a lower absolute percentage error.

TABLE III
THE MAPE AND MAE OF THE GLOBAL AND CLUSTERED MODELS, COMPARED TO THE NAÏVE PREDICTOR.

Model	MAPE [%]	MAE [MBit/s]
Clustered predictive models	9.09	4.22
K-means predictive model	9.79	4.52
Global predictive model	9.91	4.54
Naïve predictor	11.98	5.77

VII. CONCLUSION

QoS data for vehicular scenarios generally contain non-IID characteristics and concept drifts. Such properties cause difficulties in applications demanding predictions of the QoS. To address this issue, this work investigates an approach to decrease the effects of concept drift and data heterogeneity in pQoS. The foundational idea relies on that concept drift is caused by varying environments. Our solution is to provide a pQoS model that lets the predictive model belong to a smaller geographical region, rather than using the same model for all locations. Our framework creates small geographical segments and clusters segments with similar data. We construct a predictive model for each cluster. Each cluster model is trained collaboratively by the vehicles in the cluster using FL. The predictive loss function and the number of clusters are jointly optimized using proximal alternating minimization, alternating between FL and SVT. We perform a theoretical analysis that provides convergence guarantees under general assumptions. Furthermore, we provide analyses of the computational and communication complexities. Ultimately, we demonstrate that our approach improves prediction accuracy compared to a

global predictive model commonly used in previous work for pQoS. With numerical evaluations, we show that the algorithm converges in terms of prediction error and number of clusters. By plotting the communication cost, we show that optimizing the number of clusters decreases the communication cost of our algorithm.

From our results, it is obvious that the selection of the segment size is crucial. For future work, it would be interesting to further investigate how the segment size affects the end performance. It would be beneficial to obtain theoretical guidelines for selecting the segment size and to allow adaptive updates of the segment size. Furthermore, the numerical evaluation shows that a longer prediction horizon increases the prediction error. This motivates future work to handle QoS predictions on longer time horizons.

VIII. Appendix

A. Proof of Proposition 2

For the proof, we first obtain an upper bound on the distance between two consecutive sequences for the iterative procedure in (15). Then, we show that the accumulated values of distances over an infinite number of iterations are bounded. More precisely, let $\mathbf{\Theta}^* := \mathbf{\Theta}^{*(i+1)}$ and $\mathbf{O}^* := \mathbf{O}^{*(i+1)}$ be the minimizers of (15) at iteration i. Also, $\mathbf{\Theta}^{(i+1)}$ and $\mathbf{O}^{(i+1)}$ are obtained from (16) and (20), respectively. We first obtain an upper bound on the distance between $\mathbf{O}^{(i+1)}$ and $\mathbf{O}^{(i)}$. In this regard, we have that

$$\|\mathbf{O}^{(i+1)} - \mathbf{O}^{(i)}\|^2 \leq \|\mathbf{O}^{(i+1)} - \mathbf{O}^*\|^2 + \|\mathbf{O}^* - \mathbf{O}^{(i)}\|^2. \quad (29)$$

Here, we used triangle inequality. Since $\mathbf{O}^{(i+1)}$ is the outcome of the SVT algorithm, we can use the following theorem to obtain an upper bound on the first term at the right-hand side.

Theorem 1. *([33, Theorem 1]) Let $\{\mathbf{O}^{(i,l+1)}, \mathbf{Y}^{(t)}\}_{t=1}^{T_2}$ be produced by* (21)*. If $\sup_t \delta_t < 2$, then*

$$\|\mathbf{O}^* - \mathbf{O}^{(i+1)}\|^2 \leq \frac{\tilde{c}}{\sum_{t=1}^{T_2} \delta_t}, \quad (30)$$

where $\tilde{c}$ is a constant independent of T_2 and δ_t is the step size for the SVT algorithm.

Hence, by substituting (30) with $\delta_t = 1$ into (29), we obtain the following.

$$\|\mathbf{O}^{(i+1)} - \mathbf{O}^{(i)}\|^2 \leq \frac{\tilde{c}}{T_2} + \|\mathbf{O}^* - \mathbf{O}^{(i)}\|^2. \quad (31)$$

Now, we need to obtain an upper bound on the distance between $\mathbf{\Theta}^{(i+1)}$ and $\mathbf{\Theta}^{(i)}$. To this end, similar to (29), we have

$$\|\mathbf{\Theta}^{(i+1)} - \mathbf{\Theta}^{(i)}\|^2 \leq \|\mathbf{\Theta}^{(i+1)} - \mathbf{\Theta}^*\|^2 + \|\mathbf{\Theta}^* - \mathbf{\Theta}^{(i)}\|^2. \quad (32)$$

From the FedAVG step, we have the following results on the model parameters.

Lemma 1. *Let F^* and F_k^* be the minimum values of F and F_k, and $\Gamma = F^* - \sum_{k=1}^K F_k^*$. Under Assumptions 2-5, for all $t \in [T_1]$, with probability at least $1 - \mathrm{e}^{-\tilde{c}_3 \alpha_i^2}/i^2$, we have*

$$\|\mathbf{\Theta}^{(i+1)} - \mathbf{\Theta}^{(*)}\|^2 \leq \frac{4K_{\max} B(\tilde{c}_1 + 1)}{\mu^2 (T_1 + \varsigma)} + \alpha_i \Upsilon_i, \quad (33)$$

where $\varsigma = \max\{8L/\mu, E\} - 1$, and $B = \sum_{k=1}^K \frac{\sigma_k^2}{K^2} + 6L\Gamma + 8(E-1)^2 G^2$ and $\Upsilon_i = J^2 \sqrt{2 \ln i}$.

Proof. See Appendix VIII-B. □

Next, Lemma 1 yields the following inequality.

$$\|\mathbf{\Theta}^{(i+1)} - \mathbf{\Theta}^{(i)}\|^2 \leq \frac{\tilde{B}}{T_1 + \varsigma} + \alpha_i \Upsilon_i + \|\mathbf{\Theta}^* - \mathbf{\Theta}^{(i)}\|^2, \quad (34)$$

where $\tilde{B} = 4K_{\max} B(\tilde{c} + 1)/\mu^2$. Afterwards, by summing up both sides of inequalities in (34) and (31), we have that

$$\begin{aligned} &\frac{\alpha_i}{2}\|\mathbf{\Theta}^{(i+1)} - \mathbf{\Theta}^{(i)}\|^2 + \frac{\beta_i}{2}\|\mathbf{O}^{(i+1)} - \mathbf{O}^{(i)}\|^2 \leq \\ &\quad \frac{\tilde{B}\alpha_i}{2(T_1 + \varsigma)} + \frac{\beta_i \tilde{c}}{2T_2} + \frac{\alpha_i^2}{2}\Upsilon_i + \frac{\alpha_i}{2}\|\mathbf{\Theta}^* - \mathbf{\Theta}^{(i)}\|^2, \\ &\quad + \frac{\beta_i}{2}\|\mathbf{O}^* - \mathbf{O}^{(i)}\|^2. \end{aligned} \quad (35)$$

The next step is to obtain an upper bound on the last two terms on the right-hand side on (35). Therefore, from (15), we can write that

$$L(\mathbf{\Theta}^*, \mathbf{O}^{(i)}) + \frac{\alpha_i}{2}\|\mathbf{\Theta}^* - \mathbf{\Theta}^{(i)}\|^2 \leq L(\mathbf{\Theta}^{(i)}, \mathbf{O}^{(i)}), \quad (36)$$

$$L(\mathbf{\Theta}^*, \mathbf{O}^*) + \frac{\beta_i}{2}\|\mathbf{O}^* - \mathbf{O}^{(i)}\|^2 \leq L(\mathbf{\Theta}^*, \mathbf{O}^{(i)}). \quad (37)$$

By summing up both sides of inequalities, we obtain the following:

$$\begin{aligned} &\frac{\beta_i}{2}\|\mathbf{O}^* - \mathbf{O}^{(i)}\|^2 + \frac{\alpha_i}{2}\|\mathbf{\Theta}^* - \mathbf{\Theta}^{(i)}\|^2 \leq \\ &\qquad L(\mathbf{\Theta}^{(i)}, \mathbf{O}^{(i)}) - L(\mathbf{\Theta}^*, \mathbf{O}^*). \end{aligned} \quad (38)$$

Hence, $L(\mathbf{\Theta}^{(i)}, \mathbf{O}^{(i)})$ does not increase. Next, we substitute (38) into (35), which gives us:

$$\begin{aligned} &\frac{\alpha_i}{2}\|\mathbf{\Theta}^{(i+1)} - \mathbf{\Theta}^{(i)}\|^2 + \frac{\beta_i}{2}\|\mathbf{O}^{(i+1)} - \mathbf{O}^{(i)}\|^2 \leq \\ &L(\mathbf{\Theta}^{(i)}, \mathbf{O}^{(i)}) - L(\mathbf{\Theta}^*, \mathbf{O}^*) + \frac{\tilde{B}\alpha_i}{2(T_1 + \varsigma)}, \\ &+ \frac{\beta_i \tilde{c}}{2T_2} + \frac{\alpha_i^2}{2}\Upsilon_i. \end{aligned} \quad (39)$$

Then, by performing a telescoping sum over the iterations, we get the following:

$$\begin{aligned} &\sum_{i=1}^{\infty} \frac{\alpha_i}{2}\|\mathbf{\Theta}^{(i+1)} - \mathbf{\Theta}^{(i)}\|^2 + \frac{\beta_i}{2}\|\mathbf{O}^{(i+1)} - \mathbf{O}^{(i)}\|^2 \leq \\ &L(\mathbf{\Theta}^0, \mathbf{O}^0) - L^* + \frac{\tilde{B}}{2(T_1 + \varsigma)} \sum_{i=1}^{\infty} \alpha_i, \\ &+ \frac{\tilde{c}}{2T_2} \sum_{i=1}^{\infty} \beta_i + \sum_{i=1}^{\infty} \alpha_i^2 \Upsilon_i. \end{aligned} \quad (40)$$

The inequality in (40) holds with probability as least

$$1 - \sum_{i=1}^{\infty} \frac{\mathrm{e}^{-\tilde{c}_3 \alpha_i^2}}{i^2}. \quad (41)$$

Here, we used union bound for Lemma 1 which holds for all i iterations. Note that setting a large enough initial value for α_i can ensure that the probability in (41) approaches one.

Next, under Assumption 5 and considering the fact that $\|\cdot\|_*$ is a continuous function, we conclude that $\inf L(\boldsymbol{\Theta}, \mathbf{O}) > -\infty$. Moreover, because $\sum_{i=1}^{\infty} \alpha_i < \infty$ and $\sum_{i=1}^{\infty} \beta_i < \infty$, the right hand side of (40) is a finite number, i.e,

$$\sum_{i=1}^{\infty} \frac{\alpha_i}{2} \|\boldsymbol{\Theta}^{(i+1)} - \boldsymbol{\Theta}^{(i)}\|^2 + \frac{\beta_i}{2} \|\mathbf{O}^{(i+1)} - \mathbf{O}^{(i)}\|^2 < \infty.$$

Let $\tilde{\gamma} = \min_i\{\beta_i, \alpha_i\} > 0$. Then, we have

$$\frac{\tilde{\gamma}}{2} \sum_{i=1}^{\infty} \|\boldsymbol{\Theta}^{(i+1)} - \boldsymbol{\Theta}^{(i)}\|^2 + \|\mathbf{O}^{(i+1)} - \mathbf{O}^{(i)}\|^2 < \infty.$$

Accordingly, we can deduce that

$$\lim_{i \to \infty} \left(\|\boldsymbol{\Theta}^{(i+1)} - \boldsymbol{\Theta}^{(i)}\| + \|\mathbf{O}^{(i+1)} - \mathbf{O}^{(i)}\| \right) = 0. \tag{42}$$

This proves the convergence. Based on the [15, Propostion 6] together with (42), we can deduce that $\boldsymbol{\Theta}^{(\infty)}$ and $\mathbf{O}^{(\infty)}$ approach a critical point of $L(\boldsymbol{\Theta}, \mathbf{O})$, i.e., $\partial L(\boldsymbol{\Theta}^{(\infty)}, \mathbf{O}^{(\infty)}) = 0$, which concludes the proof.

B. Proof of Lemma 1

Since the variables $\boldsymbol{\Theta}^{(i+1)}$ and $\boldsymbol{\Theta}^*$ are random, we first obtain an upper bound on the distance of their expected values. We use the key lemmas from [34], which is stated below.

Lemma 2. *Let Assumptions 2-5 hold. If the step size α_i is non-increasing and $\alpha_i \leq 1/4L$, then it follows that*

$$\mathbb{E}\|\boldsymbol{\theta}^{(i,j+1)} - \boldsymbol{\theta}^*\|^2 \leq (1 - \gamma_i^t \mu)\mathbb{E}\|\boldsymbol{\theta}^{(i,j)} - \boldsymbol{\theta}^*\|^2 + (\gamma_i^t)^2 B, \tag{43}$$

where $B = \sum_{k=1}^{K} \frac{\sigma_k^2}{K^2} + 6L\Gamma + 8(E-1)^2 G^2$ and $\Gamma = F^* - \sum_{k=1}^{K} F_k^*$ with F^* and F_k^* are the minimum values of F and F_k, respectively.

Proof. This Lemma is the direct result of [34, Lemma 1-3]. □

For a diminishing stepsize, $\gamma_i^t = 2/\mu(t+\varsigma)$ for some $\varsigma > 0$ such that $\gamma_i^t \leq 1/4L$. Then we can show that $\mathbb{E}\|\boldsymbol{\theta}^{(i,j)} - \boldsymbol{\theta}^*\|^2 \leq \nu/(t+\varsigma)$ where $\nu = \max\{4B/\mu^2, (\varsigma+1)\Delta_1\}$ and $\Delta_1 = \mathbb{E}\|\boldsymbol{\theta}^{(i,1)} - \boldsymbol{\theta}^*\|^2$. One can show this by induction. Indeed, for $t = 1$ it holds because of the definition of ν. Next, we assume that $\mathbb{E}\|\boldsymbol{\theta}^{(i,j)} - \boldsymbol{\theta}^*\|^2 \leq \frac{\nu}{t+\varsigma}$ for some t. Then, Lemma 2 gives us the following inequality.

$$\begin{aligned}
\mathbb{E}\|\boldsymbol{\theta}^{(i,j+1)} - \boldsymbol{\theta}^*\|^2 &\leq (1 - \gamma_i^t \mu)\mathbb{E}\|\boldsymbol{\theta}^{(i,j)} - \boldsymbol{\theta}^*\|^2 + (\gamma_i^t)^2 B, \\
&= \frac{t+\varsigma-2}{(t+\varsigma)^2}\nu + \frac{4B}{\mu^2(t+\varsigma)^2}, \\
&\leq \frac{t+\varsigma-1}{(t+\varsigma)^2}\nu, \\
&\leq \frac{\nu}{t+\varsigma+1}.
\end{aligned} \tag{44}$$

Next, the parameter ν can be upper bounded as $\nu \leq 4B/\mu^2 + (\varsigma+1)\Delta_1$, which yields

$$\mathbb{E}\|\boldsymbol{\theta}^{(i,j)} - \boldsymbol{\theta}^*\|^2 \leq \frac{4B/\mu^2 + (\varsigma+1)\Delta_1}{t+\varsigma}. \tag{45}$$

Then, assume $\Delta_1 \leq \tilde{c}_2 4B/\mu^2(\varsigma+1)$ for some $\tilde{c}_2$. Consequently, we get the following

$$\mathbb{E}\|\boldsymbol{\theta}^{(i,j)} - \boldsymbol{\theta}^*\|^2 \leq \frac{4B(\tilde{c}_1+1)}{\mu^2(t+\varsigma)}. \tag{46}$$

Now, to find the upper bound for the difference of $\boldsymbol{\theta}^{(i,j)}$ and $\boldsymbol{\theta}^*$, we use the Hanson-Wright inequality stated below.

Theorem 2. *With [35, Theorem 1.1] in mind, let $\boldsymbol{\psi} = [\psi_1, \ldots, \psi_W] \in \mathbb{R}^W$ be a random vector with independent components ψ which satisfy $\mathbb{E}[\psi] = 0$ and $\|\psi\| \leq J$. Then, for every $\varepsilon > 0$, we have*

$$\mathbb{P}\{|\|\boldsymbol{\psi}\|^2 - \mathbb{E}\|\boldsymbol{\psi}\|^2| > \varepsilon\} \leq 2\mathrm{e}^{-\tilde{c}_3 \min\left(\frac{\varepsilon^2}{WJ^4}, \frac{\varepsilon}{J^2}\right)}, \tag{47}$$

where $\tilde{c}_3$ is a positive constant.

By letting $\psi = \boldsymbol{\theta}^{(i,j)} - \boldsymbol{\theta}^*$ and setting $J := \max_{i,j} \|\boldsymbol{\theta}^{(i,j)} - \boldsymbol{\theta}^*\|$ over the entire procedure of learning, we use one-sided version of the inequality in Theorem 2 to obtain the following upper bound.

$$\begin{aligned}
\|\boldsymbol{\theta}^{(i,j)} - \boldsymbol{\theta}^*\|^2 &\leq \mathbb{E}\|\boldsymbol{\theta}^{(i,j)} - \boldsymbol{\theta}^*\|^2 + \alpha_i J^2 \sqrt{2\ln i}, \\
&\leq \frac{4B(\tilde{c}_1+1)}{\mu^2(t+\varsigma)} + \alpha_i J^2 \sqrt{2\ln i},
\end{aligned} \tag{48}$$

with the probability at least

$$1 - \mathrm{e}^{-\tilde{c}_3 \alpha_i^2 2\ln i} \geq 1 - \frac{\mathrm{e}^{-\tilde{c}_3 \alpha_i^2}}{i^2}. \tag{49}$$

Since we have T_1 communication rounds and $K_{\max}$ clusters, we can conclude that

$$\|\boldsymbol{\Theta}^{(i+1)} - \boldsymbol{\Theta}^{(*)}\|^2 \leq \frac{4K_{\max}B(\tilde{c}_1+1)}{\mu^2(T_1+\varsigma)} + \alpha_i \Upsilon_i,$$

where $\Upsilon_i = J^2\sqrt{2\ln i}$ for $i > 1$. Therefore, we can conclude the proof.

REFERENCES

[1] O. Stenhammar, S. Razavikia, G. Fodor, and C. Fischione, "Clustering of geographical segments for predictive quality of service of connected vehicles," *IEEE TVT*, vol. 74, no. 11, pp. 18 049–18 064, 2025. DOI: 10.1109/TVT.2025.3580989.

[2] O. Stenhammar, G. Fodor, and C. Fischione, "Clustered predictive quality of service for connected vehicles using federated learning," in *IEEE ICC Workshops*, 2024, pp. 626–631.

[3] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, *et al.*, "6G internet of things: A comprehensive survey," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 359–383, 2022.

[4] S. Dang, O. Amin, B. Shihada, and M.-S. Alouini, "What should 6G be?" *Nature Electronics*, vol. 3, no. 1, pp. 20–29, 2020.

[5] G. Kakkavas, M. Diamanti, V. Karyotis, K. N. Nyarko, M. Gabriel, *et al.*, "5G perspective of connected autonomous vehicles: Current landscape and challenges toward 6G," *IEEE Wirel. Commun.*, pp. 1–8, 2024.

[6] M. N. Ahangar, Q. Z. Ahmed, F. A. Khan, and M. Hafeez, "A survey of autonomous vehicles: Enabling communication technologies and challenges," *Sensors*, vol. 21, no. 3, 2021.

[7] G. Fodor, J. Vinogradova, P. Hammarberg, K. K. Nagalapur, Z. T. Qi, *et al.*, "5G new radio for automotive, rail, and air transport," *IEEE Commun. Mag.*, vol. 59, no. 7, pp. 22–28, 2021.

[8] W. Yang, B. Jeon, C. Mun, and H.-S. Jo, "Cellular-V2X QoS adaptive distributed congestion control: A deep Q network approach," in *IEEE CCNC*, 2023, pp. 967–968.

[9] 5GAA Automotive Association, *Making 5G proactive and predictive for the automotive industry*, available at https://5gaa.org/5gaa-releases-white-paper-on-making-5g - proactive - and - predictive - for - the - automotive - industry/, published: Dec. 2019, accessed: 2024-06-20.

[10] M. Boban, M. Giordani, and M. Zorzi, "Predictive quality of service: The next frontier for fully autonomous systems," *IEEE Netw.*, vol. 35, no. 6, pp. 104–110, 2021.

[11] A. Palaios, C. L. Vielhaus, D. F. Külzer, C. Watermann, R. Hernangómez, *et al.*, "Machine learning for QoS prediction in vehicular communication: Challenges and solution approaches," *IEEE Access*, vol. 11, pp. 92 459–92 477, 2023.

[12] A. Krause, A. Palaios, A. Kumar, P. Schulz, and G. Fettweis, "An improved data collection framework for enabling ML-based QoS prediction for vehicular communication," in *WSA & SCC*, 2023, pp. 1–6.

[13] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, *et al.*, "Learning under concept drift: A review," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 12, pp. 2346–2363, Dec. 2019.

[14] R. Blasco, G. C. Ferrante, C. Watermann, and A. Palaios, "Predictive quality of service in cellular networks: Challenges, framework, and application in vehicular communications," *IEEE Commun. Mag.*, vol. 61, no. 3, pp. 44–49, 2023.

[15] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran, "Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka-Łojasiewicz inequality," *Math. Oper. Res*, vol. 35, no. 2, pp. 438–457, 2010.

[16] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIOPT*, vol. 20, no. 4, pp. 1956–1982, 2010.

[17] R. Hernangómez, P. Geuer, A. Palaios, D. Schäufele, C. Watermann, *et al.*, "Berlin V2X: A machine learning dataset from multiple vehicles and radio access technologies," in *IEEE VTC2023-Spring*, 2023.

[18] D. F. Külzer, M. Kasparick, A. Palaios, R. Sattiraju, O. D. Ramos-Cantor, *et al.*, "AI4Mobile: Use cases and challenges of AI-based QoS prediction for high-mobility scenarios," in *IEEE VTC2021-Spring*, 2021, pp. 1–7.

[19] A. Palaios, P. Geuer, J. Fink, D. F. Külzer, F. Göttsch, *et al.*, "Network under control: Multi-vehicle E2E measurements for AI-based QoS prediction," in *IEEE PIMRC*, 2021, pp. 1432–1438.

[20] A. Taik, Z. Mlika, and S. Cherkaoui, "Clustered vehicular federated learning: Process and optimization," *IEEE T-ITS*, vol. 23, no. 12, pp. 25 371–25 383, Dec. 2022.

[21] A. Ghosh, J. Chung, D. Yin, and K. Ramchandran, "An efficient framework for clustered federated learning," *IEEE Trans. Inf. Theory*, vol. 68, no. 12, pp. 8076–8091, Dec. 2022.

[22] M. Isaksson, F. Vannella, D. Sandberg, and R. Cöster, "Mmwave beam selection in analog beamforming using personalized federated learning," in *IEEE FNWF*, 2023, pp. 1–6.

[23] M. Tang, W. Tang, and F. Xie, "Accurately predicting quality of services in IoT via using self-attention representation and deep factorization machines," *IEEE T-ITS*, vol. 24, no. 11, pp. 13 276–13 285, Jun. 2023.

[24] A. L. Blum and R. L. Rivest, "Training a 3-node neural network is NP-complete," *Neural Networks*, vol. 5, no. 1, pp. 117–127, 1992.

[25] D. Aloise, A. Deshpande, P. Hansen, and P. Popat, "NP-hardness of Euclidean sum-of-squares clustering," *Machine Learning*, vol. 75, pp. 245–248, May 2009.

[26] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM review*, vol. 52, no. 3, pp. 471–501, 2010.

[27] E. J. Candes and Y. Plan, "Matrix completion with noise," *Proc. IEEE*, vol. 98, no. 6, pp. 925–936, 2010.

[28] E. J. Candes and B. Recht, "Exact matrix completion via convex optimization," *Found. Comput. Math.*, pp. 717–772, 9 2009.

[29] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *AISTATS*, vol. 54, PMLR, Apr. 2017, pp. 1273–1282.

[30] Y. Zhang, J. C. Duchi, and M. J. Wainwright, "Communication-efficient algorithms for statistical optimization," *J. Mach. Learn. Res.*, vol. 14, no. 104, pp. 3321–3363, 2013.

[31] H. Yu, S. Yang, and S. Zhu, "Parallel restarted SGD with faster convergence and less communication: Demystifying why model averaging works for deep learning," in *AAAI Press*, vol. 33, 2019, pp. 5693–5700.

[32] Z. Zhu, K. Lin, A. K. Jain, and J. Zhou, "Transfer learning in deep reinforcement learning: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 11, pp. 13 344–13 362, Jul. 2023.

[33] Y. Lei and D.-X. Zhou, "A singular value thresholding algorithm for matrix completion," *J. Fourier Anal. Appl.*, vol. 25, no. 6, pp. 2957–2972, 2019.

[34] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FedAvg on non-IID data," *ICLR*, 2020.

[35] M. Rudelson and R. Vershynin, "Hanson-Wright inequality and sub-Gaussian concentration," *Electron. Commun. Probab.*, no. 82, pp. 1–9, 18 2013.