On the Limit Performance of Floating Gossip

Gianluca Rizzo,*‡ Noelia Perez Palma,† Marco Ajmone Marsan,† and Vincenzo Mancuso†
gianluca.rizzo@hevs.ch, noelia.perez@imdea.org, ajmone@polito.it, vincenzo.mancuso@imdea.org
* HES SO Valais † Institute IMDEA Networks ‡ Università di Foggia

***Abstract*—In this paper we investigate the limit performance of Floating Gossip, a new, fully distributed Gossip Learning scheme which relies on Floating Content to implement location-based probabilistic storage of machine learning models in an infrastructure-less manner.**

We consider dynamic scenarios where continuous learning is necessary, and we adopt a mean field approach to investigate the limit performance of Floating Gossip in terms of amount of data that users can incorporate into their models, as a function of the main system parameters. Differently from existing approaches in which either communication or computing aspects of Gossip Learning are analyzed and optimized, our approach accounts for the compound impact of both aspects. We validate our results through detailed simulations, proving good accuracy. Our model shows that Floating Gossip can be very effective in implementing continuous training and update of machine learning models in a cooperative manner, based on opportunistic exchanges among moving users.

I. Introduction

With the standardization of the 5G NR (New Radio) sidelink [1], in addition to the LTE sidelink [2], WiFi direct [3], and Bluetooth 5.2 [4], a number of options now exist for implementing wireless D2D (Device to Device) communications [5]. Such rich set of technologies is likely to foster the development of new services based on the direct exchange of information among users. This will facilitate the implementation of context-aware location-based services such as mobile crowdsensing applications, or services like Gigwalk [6], Waze [7], and Millionagents [8], to name a few, since users can share location and context through D2D automatically.

In particular, services that require the continuous acquisition of fresh information (we will refer to them with the generic term *observations*) about some specific context, and the subsequent elaboration in each user equipment (UE) of a machine learning (ML) model are poised to become common. Such ML models will allow UEs to devise up-to-date strategies for the optimization of a utility function while performing a task.

In this paper we study the effectiveness of D2D-based opportunistic services to support privacy-preserving ML-based decision making in UEs that operate autonomously, without the need for a RAN infrastructure, without centralized support, and without exchanging any observation. Our main goal is to determine the maximum amount of information that a group of UEs can autonomously learn and incorporate in their ML models, without centralized coordination, only exploiting their own computation capacity combined with D2D communications and gossip learning (GL) [9]. The GL approach is beneficial both in terms of maintaining data out of reach of indirect observers—hence avoiding tampering with and exposing private information—and in terms of scalability, because it only moves ML models based on observations, not the observations themselves. Note that a reduction of the volume of data results in simpler processing at mobile sensing platforms [10] thus making simple devices—e.g., mobile UEs—able to participate in model training processes.

Several works have previously looked at components of the system we consider. For example, a number of works on Floating Content (FC) have characterized the opportunistic D2D aspect [11]–[13]. Other works on GL have tackled the ML aspects [14]–[16]. We name the system that we propose and analyze Floating Gossip (FG), because of its roots in GL and in FC augmented with computing within UEs. Our investigation is based on a mean field approach validated through detailed simulation experiments. Specifically, the main contributions of this work are the following.

- We present Floating Gossip, a fully distributed Gossip Learning scheme based on localized probabilistic storage of models in an infrastructure-less manner.
- We derive a mean field-based analytical approach to modeling the limit performance of FG, in terms of *model freshness* (i.e., of average time elapsed since the collection of the most recent observation included in an ML model) and of *learning capacity*, i.e., of the maximum amount of observations which can be incorporated in an ML model.
- We evaluate numerically our results, validating our assumptions against simulation under different configurations, showing the accuracy of our mean field approach, and characterizing the properties of FG limit performance as a function of the main system parameters.

II. Related Work

Our work is inspired by Gossip Learning (GL) [9], a collaborative ML approach that is motivated mainly by the need to guarantee data privacy and scalability. In GL, learning is achieved by combining model training on the local dataset of each node (updated over time with new observations) with model sharing and merging with other devices. We use GL with the restriction that model sharing can only occur over D2D within the FC boundaries, which imposes specific performance constraints in addition to computation constraints. A similar system has been proposed and studied in terms of accuracy of generated models in [14]. In that work, federated learning relies on opportunistic model exchanges like in our work, and the authors show that encounter-based learning can be effective. However, they assume that training data can be at least partially exchanged over D2D, thus exposing data to privacy risks.

Giaretta and Girdzijauskas [15] consider a static network, and analyze the impact on the behavior of GL of data distribution, of network connectivity, and communication speed. Our work considers the same aspects, adding the peculiarities of a dynamic, opportunistic communication environment.

First, while a fully distributed data model is often assumed, in which each device only owns a data point and trains the model once (see [16] and references therein) — only the authors of [15] propose to train over multiple points, and present a simulation study for its performance impact — we are more general, because we allow not only devices to train multiple times on different data points, but also account for the possibility that multiple devices might observe the same event and therefore own non-unique data points. The latter has not been considered in other works so far.

Second, network connectivity is typically taken as stable, with little or no limitations imposed by physical distance and interconnection infrastructure, which is often fully comprised within a same data center or within a multi-thread computing architecture [16]–[18]. Instead, network connectivity is a key factor, which can slow down the convergence of GL, as shown for the particular case of stochastic gradient descent algorithms in [19]. We consider only opportunistic communications of physically distinct and not co-deployed devices, with no infrastructure support at all, so that the number of neighbors that can exchange models is limited. Most important, we consider dynamic topologies with churn in UEs, hence timely and opportunistically exchange of models becomes fundamental.

Third, while many works that focus on networked devices consider unrealistically high communication speed or neglect transmission time and connection establishment overheads [20], or neglect communications delay almost completely [21], we account for the limitations of D2D connectivity and protocols, which consist not only in relatively low data rates, but also in the fact that a connection takes time to be established and a device might be busy exchanging data with a UE when a new communication opportunity arises, which is therefore missed. Hence, differently from existing approaches in which either communication or computing aspects of GL are analyzed and optimized, here we focus on the compound impact of both aspects.

Eventually, we remark that there are no comprehensive analytical models to evaluate the performance of GL, and all reviewed papers show results based on simulations or real trace analysis. In contrast, we only resort to simulation to validate the complex model developed in this paper. The closest work, related to the analytical methodology used in our work is the seminal work of Chaintreau *et al.* on gossiping as a tool to spread information [22], which shows how to use mean field analysis on the epidemic diffusion of messages. We also use mean field analysis, but we account for the specifics of learning and computing in the generation of the information to be shared, and the specificity of FC as a means to keep models available in a given area of interest.

III. The Floating Gossip Scheme—System model

A. What's Floating Gossip?

We consider a dynamic environment where many events happen, such as a crowded theme park or a post-disaster area. During their stay in the environment, a number of actors (humans or machines, that we term *nodes*) collect information that helps them make decisions and/or perform some task. The individually collected information consists in local viewpoints (termed *observations*) of what is happening in the environment at a given time. By using observations, each node builds its own partial, incomplete representation of the environment (termed *model*), and may therefore take potentially sub-optimal decisions. To go beyond such limitations, we consider that nodes periodically combine their models with the ones received through D2D opportunistic exchanges. The combined model is a potentially better representation of the environment because it incorporates a larger number of observations, some older, some more recent. In general, the utility of a model is related to the number of incorporated observations, as well as to the time at which they have been collected.

Furthermore, we consider that models are relevant only in specific geographical areas, and that observations become obsolete after a fixed amount of time. Therefore stale observations are discarded, and nodes moving outside of the region of interest drop the instances of the models they own.

Our approach is thus similar to the one adopted in GL, but it is based on models and observations whose availability persists in the region of interest thanks to an opportunistic floating content scheme [23]. Our approach can be seen as a machine learning process, where each observation is incorporated into the model (through a *training* algorithm), and models are exchanged and fused (through a *merging* algorithm), as in GL. In the rest of this paper we will refer to this case, but we will not address any specific ML algorithm, so that our model is general enough to be applicable to several different ML techniques.

B. Some definitions

We consider a population of nodes that cooperate in constructing a multidimensional representation of the environment in which they operate. Each dimension of the representation is associated with a model.

For the sake of simplicity we describe the system operations in the case of one model, concerned with one dimension of the environment (e.g., the road interruptions in the disaster area), but later we will also look at the case of several models evolving in parallel.

Nodes move within an area, a portion of which defines the model Replication Zone (RZ). When entering the RZ, nodes have a *model structure* (also called *default model*), not yet populated with fresh data (for example, a rescue team entering a disaster area has a map that does not yet contain recent information on the critical issues in the area). While in the RZ, nodes collect data (i.e., observations) that are used to integrate the model structure with a training operation. By so doing, each node generates and updates its own *model instance*.

Each node is equipped with a wireless transceiver, and a positioning system that gives its position in space. We say two nodes are *in contact* when they are able to directly exchange information via wireless communications.

When two nodes come in contact of one another within the RZ, they may exchange their model instances by means of D2D, provided that the nodes are not already engaged in other D2D exchanges, or that their models do not incorporate identical sets of observations. Each node then produces a better informed version of its own model instance by merging it with the received one.

A model is thus a description of the environment according to one dimension, and can be used to support decisions related to such dimension. The local model instance of a node is generally different from that of other nodes. A model instance is trained with the node's own observations (which form the model *local training set*), and can be merged with other nodes' model instances to incorporate their observations (the union of the local training sets of all merged models forms the model total training set).

The model training operation is a transformation which takes as input a model instance, and a new set of one or more observations, and gives as output a new model instance, with a training set given by the union of the training set of the old model instance and the new set of observations. In what follows we assume that model training is performed one observation at a time, though our approach easily extends to the case in which all observations in the new set are incorporated as a batch in the model instance.

The *merging* operation is a transformation which takes as input two instances of a model, and gives as output a new model instance, whose training set is the union of the training sets of both of the input instances. It occurs after an exchange of models among nodes, if the newly received model instance(s) incorporate observations that were not present in the local model instance.

The process by which the new model is derived is not specified in our work. Typically, in the case of Artificial Neural Networks (ANNs), the coefficients of the model obtained through merging are derived as a weighted average of the coefficients of the merged model instances, though more complex approaches are possible. Such a definition of the merging operation seems to suggest that it produces a model instance with the same performance (e.g., in terms of accuracy) as a model instance obtained by training over the union of the two training sets. However, this is well known to be an approximation.

Every time a node collects a new observation, it uses the new observation to train its local instance of the model. The *age* of an observation is the time since its generation. Finally, users exiting the RZ discard their local model instance, as well as all collected observations.

C. Modeling assumptions

We consider nodes moving over the plane according to a stationary mobility model such that, at any time instant, nodes are uniformly distributed in space. We consider the RZ to be a finite region of the plane, of arbitrary shape and area A. With α we denote the mean rate at which nodes enter the RZ, (which coincides with the mean rate at which nodes exit the same area because of equilibrium).

We assume that the environment is constantly changing, thus requiring a continuous process of observation collection and incorporation in the models. We assume that there exist M models in total, and each node can hold a single instance of up to W models, assuming that a realistic implementation of FG will impose such practical limitation. When the number of models M is larger than W, we assume each node chooses a subset of models of interest, which is equivalent to subscribe to up to W observation channels. For the remaining $M - W$ models, the node will not accept any instance, nor it will record any observation associated to them. To simplify the notation, in this paper we derive analytical expressions that hold for $M \geq W$. However, substituting $\min(M, W)$ for M would make the same expressions valid for the general case.

Differently from GL, in FG learning occurs in a fully distributed way, using opportunistic data exchange through D2D protocols. We partially neglect the overhead of such protocols and retain two main aspects: $i)$ connection establishment takes time t_0, which we consider as a constant, and $ii)$ data transfer is not instantaneous, but depends on the quantity of data to exchange bidirectionally, since we consider the transfer rate to be fixed. We also assume that data exchanges occur pairwise, and it is not possible to create and manage groups with more than two connected nodes, which would be highly unstable due to mobility. We say that two nodes establishing a connection or exchanging models are *busy*, so they cannot accept any request to connect with other nodes until they finish and immediately disconnect.

These parameters are important to determine the success of a model transfer upon a contact between two nodes. Other important factors are the frequency of contacts observed by each node, g, and the duration of contacts, which is denoted by t_c. The values of g and t_c are the same for all nodes. In particular, t_c is a random variable with probability density function (pdf) denoted by $f(t_c)$. The mean time necessary for the transfer of a model instance between two nodes is instead denoted as T_L, which is a function of the capacity of the channel between two nodes in contact, as well as of the size of the model instances, which we assume to be equal to L bits. The order in which model instances are transferred upon a contact is important, but since scheduling optimization is out of the scope of our investigation, we assume that models to be transferred are chosen at random among the available ones.

Once exchanged, instances belonging to the same model are then merged through local computation, again without the help of an infrastructure. We assume that received model instances are not used and propagated until they have been fused with the local instance. This requires storage and computing resources at the nodes not only for incorporating observations into local model instances (training), but also to fuse different instances of the same model, when they are received from neighbors (merging). We assume that storage/compute resources are used as in a system with two FIFO queues, one for training and one for merging. We assume a single shared service with finite capacity, that gives strict priority to merging tasks, with no preemption of training tasks. We do this because merging tasks move more information at once than training tasks, so they are more important to enable the derivation of better models. For simplicity, we assume that queues have no buffering limit, although in practice we will show that they need limited storage space. Training and merging times are assumed to be constant, and are indicated as T_T and T_M, respectively.

Moreover, since observations are sequentially generated and model instances are asynchronously acquired, they are queued individually and then trained or merged one by one. However, an observation can be recorded simultaneously by $1 \leq \Lambda \leq W$ nodes (i.e., by the Λ nodes interested in the model the observation is relevant for, and being in proximity of the point at which the observation can be gathered). Model instances resulting from training and merging are uniquely identified by the set of observations used to build the merged instances. Therefore, we also assume that two instances of the same model with a same (aggregate) training set are indistinguishable. We assume however that relevant observations are all those generated within a maximum age, which we denote as τ_l. That is, observations older than τ_l do not need to be incorporated into a model, and they are not counted as part of the training set. This makes sense in those scenarios in which the process which underlies the generation of the observations is not stationary over time. Finally, we remark that since the model size is finite, an instance of the model can only accommodate a limited number of observations. We therefore assume that if a new observation is added when the model instance is "full", the oldest observation is dropped from the training set.

IV. Performance Analysis of Floating Gossip

The main goal of the FG scheme is to enable the elaboration of "well-enough trained" models, and to share them among users within the RZ.

In what follows, we consider the system at times t in which the process of model sharing within the RZ is in steady state.

A first performance metric for our FG scheme is the *observation incorporation rate*, i.e., the rate at which observations with a given age are incorporated in a model.

Definition 1. *For a given model, a given age of observation τ, and for a time t, the observation incorporation rate $R(t, \tau)$ at time t is the average of the ratio between the amount of observations with age $\in (\tau, \tau + \Delta\tau)$ incorporated in an instance of the model at time t, and the time interval $\Delta\tau$, for $\Delta\tau \to 0$.*

Since we study the system at steady state, we assume that the observation incorporation rate does not depend on t, or on the location of observation generation. The observation incorporation rate is an indicator of the effectiveness and of the speed with which our FG scheme is able to store new observations in a model.

Another measure of the ability of the FG scheme to incorporate the most recent measured data is the *model staleness.*

Definition 2. *The staleness F_i of a model instance is the average time elapsed since the generation of the most recent observation included in the model instance. The staleness F of a model is the average of F_i over all instances of the model present in the RZ.*

Model staleness is a measure of how much "up to date" on average an instance of a model is, in a given scenario. Note however that a staleness value of τ does not say *how many observations* are included in a given interval $(\tau, \tau+\Delta\tau)$. Thus, for instance, it does not necessarily imply that enough observations are included on average in an instance of the model to allow the model to have a comprehensive picture of the status of the observed system at τ.

A. Derivation of Observation Incorporation Rate and of Model staleness

Definition 3. *With reference to a given observation and the node on which it resides, the seeding probability P is the probability that the observation is included in a model instance at the node by the time the node exits the RZ.*

With Y we denote the probability that at a node the local instance is not merged with a received instance because the training set of the former is a superset of the training set of the latter.

Definition 4. *We say that the system is in the substable regime when $P \approx 1$ and $Y \approx 0$.*

The substable regime implies that the computing load is low enough for an observation to have very high chances to be incorporated in an instance by training. The condition $Y \approx 0$ instead is typically achieved when on average each model instance has a large training set, so that the likelihood that a merging between the local instance and an incoming model does not increase the training set of the local instance is low. In practice, the substable regime assumption is an approximation, which brings to accurate results when the above conditions are met.

We now introduce three of the key parameters describing the dynamics of the FG system: the *model availability*, the *node busy probability* and the *observation availability.*

Definition 5. *The availability of model m at time t, denoted as $a_m(t, A)$, is the mean fraction of nodes possessing a non-default instance (i.e., an instance with a non-empty training set) of the m-th model, where A is the RZ area.*

In general, model availability takes different values for different models in a same RZ.

Definition 6. *The probability that a node within a RZ with area A is busy at time t, denoted with $b(t, A)$, is the probability that the node is not available for information exchange with other nodes in contact, at the given time instant.*

The following result states that, $\forall m$, the steady state values of $a_m(t, A)$ and $b(t, A)$ depend neither on the specific trajectories (i.e., on the amount of initial seeders for each model) nor on the time at which each model has been seeded, but only on properties of the system once all transients are exhausted.

Lemma 1. *For $t \to \infty$, in the substable regime the mean values of the variables $a_m(t, A)$, $\forall m$, and $b(t, A)$ at the mean field limit (denoted as a and b, respectively) are given by the unique solution of the following fixed point problem in a:*

$$a = 0.5\left(H + \sqrt{H^2 + \frac{4T_S(a)\lambda\Lambda}{bNS(a)w}}\right) \tag{1}$$

with

$$\begin{aligned}
H &= 1 - \frac{T_S(a)(\alpha + \lambda\Lambda)}{bNS(a)w}\\
b &= K - \sqrt{K^2 - 1}\\
K &= 1 + \frac{1}{4gT_S(a)} + \frac{\alpha}{2gN}\\
\gamma &= 2Mw^2a \qquad\qquad (2)\\
S(a) &= \int_{t_0}^{+\infty} \min\left(1, \left\lfloor\frac{(t_c - t_0)}{T_L}\right\rfloor \frac{1}{\gamma}\right) f(t_c)dt_c\\
T_S(a) &= \int_0^{+\infty} \min\left(t_c, \gamma T_L + t_0\right) f(t_c)dt_c
\end{aligned}$$

where t_0 is the transfer setup time, N is the mean number of nodes in the RZ, and $w = \min\left(\frac{W}{M}, 1\right)$. Independently from the initial condition, any trajectory of the system converges to the solution of Equation (1).

For the proof, please refer to Section A in the appendix.

Let us now consider an observation which has been incorporated into the m-th model, and let τ denote its age. Let $o(\tau)$ denote the mean *observation availability*, i.e., the mean fraction of nodes with the m-th model whose local instance includes the given observation at age τ, at the mean field limit.

Definition 7. *Given a node in the RZ, a given model, and a given observation of age τ, at the mean field limit, the mean observation availability $o(\tau)$ is the probability to find that observation of age τ in the training set of an instance of that model at that node, assuming that 1) that node possesses an instance for that model, and 2) the given observation has been incorporated by training in at least one instance of that model in the whole RZ.*

This definition implies that, given a node, a model and an observation of age τ, the probability that it is included in an instance of that model on that node is $wao(\tau)$. Let d_I denote the mean time taken by an observation to be incorporated through training, and d_M the mean time required by an instance of the m-th model received by a node to be merged.

Lemma 2. *At the mean field limit, in the steady state for the model diffusion process, in the substable regime the mean arrival rate of merging tasks at a node r is $r = MaSw^2g(1-b)^2$, where S, a and b are given by Lemma 1.*

For the proof, please refer to Section B in the appendix.

Lemma 3. *Let t^* be the mean sojourn time in the RZ. When the stability condition*

$$\left(\frac{Mw\lambda\Lambda T_T}{N} + rT_M\right) \vee \left[\frac{1}{t^* 2(1 - rT_M)}\cdot\right.$$

$$\left.\cdot\left(\frac{rT_M^2}{1 - rT_M} + \frac{T_T(2 - \frac{M\lambda\Lambda}{N}T_T)}{1 - \frac{M\lambda\Lambda}{N}T_T}\right)\right] \leq 1 \tag{3}$$

holds, d_M and d_I are given by

$$d_M = T_M + \frac{rT_M^2}{2(1 - rT_M)} + \frac{Mw\lambda\Lambda}{N}{T_T}^2$$

$$d_I = \frac{1}{1 - rT_M}\left(\frac{rT_M^2}{2(1 - rT_M)} + T_T + \frac{\frac{Mw\lambda\Lambda}{N}T_T^2}{2(1 - \frac{Mw\lambda\Lambda}{N}T_T)}\right) \tag{4}$$

Proof. The system can be modeled as $M/D/1$ queue with two classes of customers, in which merging has non-preemptive priority over training. For the high priority class, the arrival rate is rT_M, and it is $\frac{Mw\lambda\Lambda T_T}{N}$ for low priority. From the condition of stability of the $M/D/1$ queue we derive the condition $rT_M + \frac{Mw\lambda\Lambda T_T}{N} \leq 1$. The derivation of delay formulas follows standard results from queuing theory for such systems. The second stability condition derives from imposing that the mean delays for both classes be less than the mean sojourn time in the RZ. □

The following result gives an analytical expression for the incorporation rate, as a function of the mean value of the observation availability at the mean field limit and over finite time intervals, for large RZ areas, hence for a large amount of nodes involved in the scheme.

Theorem 1. *When the stability condition* (3) *holds, at the mean field limit the incorporation rate at age τ is given by $R(\tau) = \lambda o(\tau)$. The mean observation availability at age τ at the mean field limit $o(\tau)$ is given by the solution of the following delay differential equation:*

$$\frac{do(\tau)}{dt} = \frac{bS(a)w^2}{T_S(a)}\left[(1 - a)o(\tau) + \right.$$

$$\left. + ao(\tau - d_M)(1 - o(\tau - d_M))\right] - \frac{\alpha w}{N}o(\tau) \tag{5}$$

with initial condition

$$o(\tau) = \begin{cases} 0 & \tau < d_I \\ \frac{1 + (\Lambda - 1)}{\lceil aN \rceil} & d_I \leq \tau \leq d_I + d_M \end{cases} \tag{6}$$

For the proof, please refer to Section C in the appendix. This result states that the probability of observing a difference between any point of the trajectory of the availability of a given observation and the solution of (5) goes to zero as the RZ area (and hence the mean number of nodes in a RZ) grows. That is, in the mean field limit, the error made by considering a deterministic system characterized by $o(\tau)$, instead of the actual system goes to zero. Moreover, at any time τ from observation generation, for any A, $o(\tau)$ is the expected value of the observation availability at τ in a finite system.

Theorem 2. *$\forall i \in \mathbb{Z}^+$, let $\gamma_i = \sum_{k=1}^{i} \xi_k$, with ξ_k i.i.d. $\sim Exp(\lambda)$. When the stability condition holds, the mean staleness F of a model is lower bounded by*

$$F \geq \frac{\delta \sum_{i=1}^{\infty} iE\left[o(\gamma_i)|\gamma_i \leq \tau_l\right] \prod_{j=1}^{i-1}(1-E\left[o(\gamma_j)|\gamma_i \leq \tau_l\right])}{\sum_{i=1}^{\infty} E\left[o(\gamma_i)\right] \prod_{j=1}^{i-1}(1-E\left[o(\gamma_i)|\gamma_i \leq \tau_l\right])} \quad (7)$$

For the proof, please refer to Appendix Section D.

V. Learning Capacity of a FG system

In this section, we derive a set of results which allow evaluating the key performance indicators of our decentralized FG framework. In what follows, we mostly refer to the case in which models are neural networks, but considerations may be extended to a large class of deep and shallow learning.

We know from the literature (see [24] and references therein) that when the total number of coefficients of an ML model is larger than the cardinality of the training dataset, then labels can be perfectly recovered. One of the interesting consequences of this result is that it establishes a linear relationship between model size L, i.e., the total amount of bits required to represent the model coefficients, and the maximum amount of information that the model can store. Thus, if a model has length L, and k coefficients are needed to encode one bit of information *independently from other coefficients*, with a given ML algorithm, then the ratio L/k is the largest amount of information (expressed in bits) that a model is able to store and recreate with vanishing little error [24]. Note that typically data in a training set is not i.i.d., and the correlation among data points translates into looser requirements on the maximum amount of information that the model can store than the one indicated by these results.

One of the measures of effectiveness of FG training in a dynamic setting is the *node stored information*.

Definition 8. *Consider a FG system at steady state for the process of model diffusion. The node stored information is the mean of the total amount of observations incorporated in all of the model instances possessed by a node, with an age not greater than τ_l.*

Lemma 4. *When the stability condition* (3) *holds, the node stored information is $Mwa \min\left(\frac{L}{k}, \lambda \int_0^{\tau_l} o(\tau)d\tau\right)$.*

Proof. The probability for a user to possess an instance of a given model is wa, and thus Mwa is the mean number of model instances it possesses, at the mean field limit. For each model, the mean amount of observations incorporated is the minimum between the maximum amount that the model can store, and the mean of the total number of observations incorporated in an instance of that model with an age between 0 and τ_l, given by the integral of the observation incorporation rate within $[0, \tau_l]$. □

Note that at the end of Section III-C we have assumed that, in a model which is already storing an amount of information L/k, the incorporation of the new data point implies discarding the oldest data point in the model. This assumption corresponds to an ideal case, as in practical cases many models simply degrade their accuracy when incorporating more points than they can store. Thus, Lemma 4 gives an upper bound on the total amount of information stored at a node.

Another key performance indicator of our FG scheme is the *learning capacity*.

Definition 9. *In a FG system at steady state, the learning capacity is the maximum achievable ratio between the node stored information and the total observation arrival rate.*

The learning capacity gives an idea of up to which point our FG scheme is able to effectively incorporate in the models possessed by nodes in the region those observations which are continuously generated within it, and thus to keep the models up to date. Specifically, the learning capacity of an FG scheme is the solution of the following problem:

Problem 1 (FG learning capacity).

$$\begin{aligned} &\underset{M,L}{\text{maximize}} \;\; wa \min\left(\frac{L}{\lambda k}, \int_0^{\tau_l} o(\tau)d\tau\right) \\ &\text{subject to :} \\ &\qquad \text{Equations } (1),\ (3),\ (5) \\ &\qquad M \geq 1, \quad L \geq L_m \end{aligned}$$

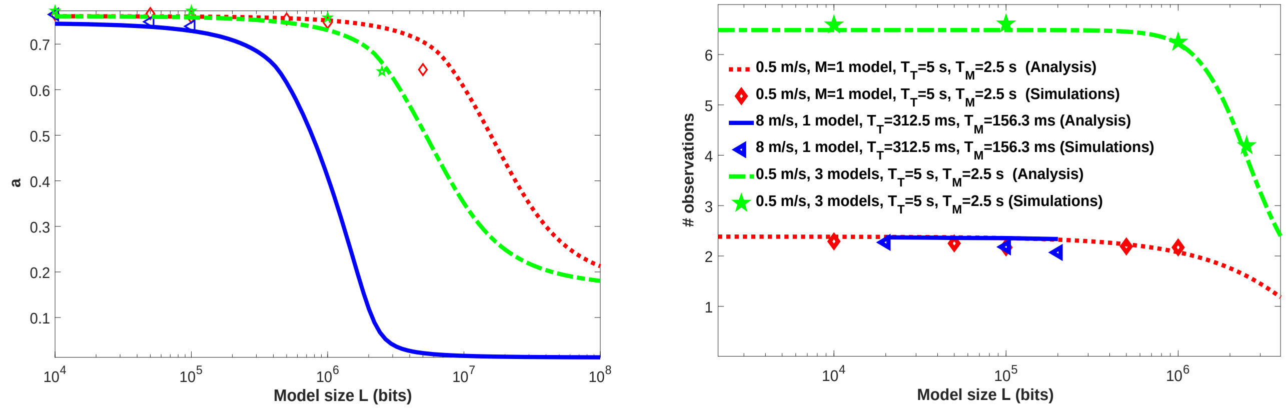
Proposition 1. If (M^*, L^*) is a solution of Problem 1, then $L^* = L_m$.

Proof. $\frac{wLa}{\lambda k}$ is directly proportional to the *area capacity* of the floating system represented by the RZ area and the M models floating within it (see [25] and references therein). Thus, for the first element of the minimum in the objective function, Proposition 1 follows from its properties. As for the second element, we note that w does not depend on L. As for $o(\tau)$, as shown in Theorem 1, it is a nondecreasing function of τ. Due to this, it is easy to see from its expression that the integral $\int_0^{\tau_l} o(\tau)d\tau$ monotonically decreases when a decreases. At the same time, when keeping M constant, a decreases monotonically as L increases, due to the well known FC saturation effects. Indeed, holding constant everything else, the lower the content size, the lower the amount of contact time spent in content transfers which do not complete due to the end of contact time. Finally, for the same reason, the left member of constraint (3) decreases when decreasing L, increasing the maximum value of M which satisfies all constraints in Problem 1. □

Therefore, Problem 1 can be cast into a maximization problem over M only, with model size equal to its minimum value. Thus, it can be solved efficiently with greedy approaches.

VI. Numerical evaluation

In this section, we use the mean field model presented in this paper to characterize the limit performance (in the mean



(a) Mean model availability.

(b) Node stored information.

Fig. 1: Mean model availability a (left) and node stored information (right) in a FG system versus model size L, for different values of training time (T_T) and merging time (T_M), of node speed, and of number of models M. Simulation results (represented by markers) are averages with a 95% confidence intervals of at most 5% (not shown in the figure).

field sense) of a FG system as a function of the main system parameters, and we assess the accuracy of the model by means of detailed simulations.

In the simulator, we consider a square area with side 200 m, at the center of which is a circular RZ, whose radius is 100 m. A fixed number of nodes (200 is the default value) move according to the Random Direction Mobility Model, with reflections at the boundary of the simulated area. Nodes have a transmission radius of 5 m, and enough storage to accommodate an arbitrary number of models. When two nodes are in contact, the channel data rate is taken to be constant, equal to 10 Mb/s. The default duration of training and merging tasks is 5 s and 2.5 s, respectively. The maximum age for an observation is 5 min, and the default model size is 10 kb (hence, model exchanges require 2 ms). At the beginning of each simulation run, nodes are randomly positioned within the simulation area, and they only possess a default model instance. Simulation time is divided in equally sized slots, whose duration has been chosen in such a way as to minimize quantization effects on the accuracy of results. Simulations are run until steady state is reached, and transients are discarded. In a first set of experiments, we have assessed model accuracy by comparing analytical and simulation results for the mean model availability and the mean number of observations stored by each node, as a function of model size. As Fig. 1 shows, the estimates derived with our mean field approach well match the simulation results across different values of node speed, of model size, of number of models, and of training and merging times. We can also see that when model size gets close to the maximum amount which can be sustained by the system (due to limitations in contact duration among nodes and in channel throughput), our mean field model yields slightly optimistic results, due to the difference between finite systems and their mean field limit.

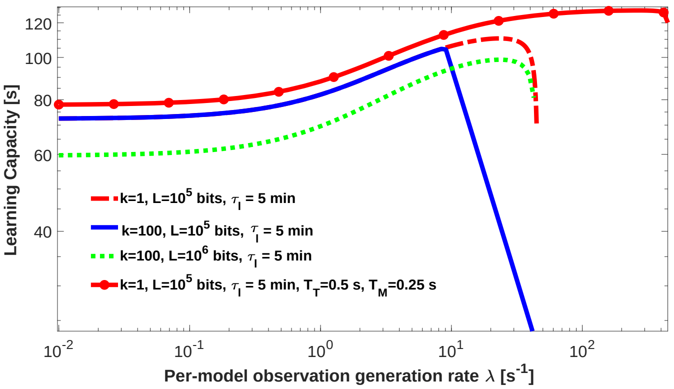


Fig. 2: Learning capacity in a FG system versus per-model observation generation rate λ, as a function of observation lifetime τ_l, model size and capacity (L/k), and training/merging times T_T and T_M.

To investigate the properties of the FG learning capacity, in Fig. 2 we plot the optimal values deriving from the solution of Problem 1 as a function of the per-model observation generation rate. As the plot suggests, when the model capacity L/k is large enough, the learning capacity grows with the observation arrival rate (and thus with the amount of information which needs to be captured). This is due to the fact that a higher rate of arrivals of observations leads to a larger model availability, and thus to a higher chance for observations to be incorporated in a model and to diffuse in the RZ. The experiments also show that learning capacity peaks at a point at which further increases in load are not sustainable due to node computing capacity or data transfer capacity limits. For higher loads, it decreases sharply, until the system is no longer stable, i.e., it is no longer able to sustain the demand for computing (due to training and merging) that the arrival rate of observations induces. The curves also show that decreasing training and merging times by a factor ten with respect to their default values increases the maximum value of observation arrival rate beyond which the system is unstable by a factor ten. Not surprisingly, decreasing those computing times increases also learning capacity, since it accelerates the diffusion of models and of observations within the considered region. When model capacity is not "large enough", even if the system is in its linear regime, the amount of observations incorporated (and thus the learning capacity) grows until it hits the model capacity limits. After that, it remains constant,

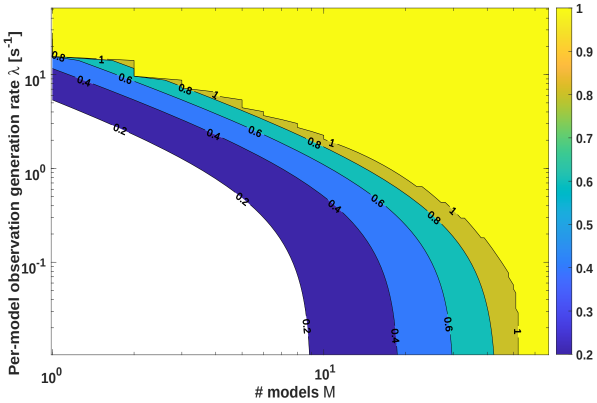


Fig. 3: Value of the left hand side of stability condition (3) versus number of models M and per-model observation generation rate λ, for observation lifetime $\tau_l = 300s$, model size $L = 10$ kb, and $k = 1$.

implying a learning capacity that decreases with the inverse of the load.

Next, with Fig. 3 we characterize the stability region of the system, i.e., the range of values of load and of number of models for which (3) holds, so that the FG system is actually able to incorporate new observations in a timely manner. In the figure, the color scale corresponds to the value of the left-hand-side of (3), which must be below one. Hence, the yellow area above the isotonic line at value 1 represents unstable points. The same line also illustrates the tradeoff between number of models and observation generation rate. In this specific case, the system is stable with at most about 40 models with observations generated every 100 seconds, but also with just one model whose observations can be generated as frequently as about twenty times per second, which is a rate suitable to capture a video. From Fig. 3, it emerges that for small values of per-model observation generation rate (with respect to observation lifetime), the system is not limited by node computing load due to training and merging, rather by the maximum amount of models which it can sustain for a given model size. Indeed, from the figure, stability depends mainly on number of models in the system. Note that, for a given maximum age of observations, there is a lower bound on the minimum per-model observation generation rate, below which models are likely to incorporate very few (if any) observations, thus making the whole FG system useless. When observation generation rate increases, the system performance is progressively more dependent on the maximum computing load it can tolerate. Clearly, in this "computing-limited" regime, an increase in the number of models entails not only an increase in the training load, but also an increase in overall observation arrival rate, and thus merging load. Indeed, as the plot shows in this regime, keeping the stability condition unaltered after such an increase would entail decreasing the per-model observation arrival rate, with a relationship of (roughly, logarithmically) inverse proportionality. Note that discontinuities in the plot are due to quantization in the number of models.

Coming to model staleness, we report in Fig. 4 the model prediction normalized to the per-model mean observation interarrival time, versus the per-model observation generation rate λ, in the case of observation lifetime $\tau_l = 300$ s, $k = 1$. As we can expect, staleness decreases for increasing observation generation rate, as it is the age of the most recent observation incorporated in a model. However, we are interested in how quickly an observation is incorporated into a model. As we can see, for low values of λ, an increase of λ brings the FG system to incorporate more quickly an observation, as a higher observation arrival rate implies a larger model availability and thus a more effective diffusion of observations in the region. However, for large values of λ, the average computing load of nodes becomes high enough to actually slow down the process of diffusion of observations. Thus, in this operating regime the normalized model staleness first peaks and then decreases. For high enough values of λ (where curves stop), the system becomes unstable. As expected, this takes place for lower values of λ in the case of a higher number of models. The figure also shows that by increasing the number of models floating in the RZ, the normalized staleness increases, as expected. However, the increase is less than linear with the number of models. At the peak, moving from $M = 1$ to $M = 25$ models only incurs a staleness increase of about 10%. From this observation we conclude that FG scales relatively well with M.

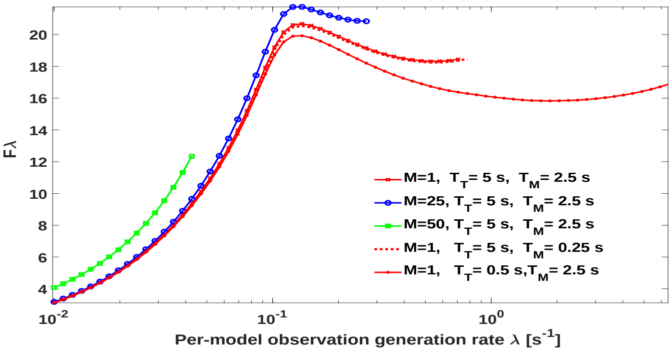


Fig. 4: Ratio of model staleness F, computed via its bound (7), over per-model mean observation interarrival time $1/\lambda$, versus per-model observation generation rate λ. Observation lifetime is $\tau_l = 300$ s, $k = 1$, with model size $L = 10$ kb.

VII. Conclusions

In this paper we have developed a mean field model to investigate the limit performance of Floating Gossip, a distributed learning scheme based on Floating Content for the opportunistic exchange of ML models stored by mobile users. Our study shows that Floating Gossip can be very effective in implementing opportunistic, fully distributed, privacy preserving, training and update of ML models in a cooperative manner. Differently from centralized training schemes, whose limit performance is heavily degraded by the effects of node churn and mobility, gossip-based schemes have proven to be very robust against those factors, and to depend heavily on nodes' ability to harvest data about the environment in which they operate.

Acknowledgments

This work was partially supported by COST INTERACT, by SNF Dymonet project, and by the AEON-CPS project (TSI-063000-2021-38) funded by the Spanish Ministry of Economic Affairs and Digital Transformation.

Appendix

A. Proof of Lemma 1 (outline)

First, it is easy to see that, with the given assumptions, the number of non-default model instances possessed by a node is distributed as a binomial $B(Mw, a_m(t,A))$. The mean number of non-default model instances possessed by a node in stationary state is thus $Ma_m(t,A)w$.

To derive the mean value of the variables $a_m(t,A)$ and $b(t,A)$ at the mean field limit, we first derive the differential equations for the drift [26], [27]. Let N_a denote the mean number of nodes in the region possessing a given model at time t. We have therefore $N_a = a_m(t,A)AD$, where D is node density. The rate of change of N_a is given by the sum of three components. The first is given by those nodes which complete a model instance transfer in a contact with another node, $\frac{ADb(t,A)}{T_s}a_m(t,A)(1-a_m(t,A))S(t)$. The second term is given by the rate at which a node receives an observation for that model (i.e., $\frac{\lambda\Lambda}{DA}$) multiplied by the probability that the given node does not have the model yet (i.e., $(1-a_m(t,A))w$). Finally, the term $\frac{\alpha}{N}wa(t)$ accounts for the rate at which nodes with a model instance leave the given region.

Let us now consider the rate at which the mean number of busy nodes in the region evolves over time. Its increase is due to contacts (taking place at a rate of gA), among two non-busy nodes (with a probability $(1-b(t,A))^2$). Its decrease is due to the end of the process of model exchange with another node. Such an event takes place with a mean rate $\frac{ADb(t,A)}{T_s}$. Putting all together, and normalizing by AD, we get

$$\begin{cases} \dfrac{da_m(t,A)}{dt} = \dfrac{b(t,A)}{T_s(t)}a_m(t,A)(1-a_m(t,A))S(t)w^2 + \\ +\dfrac{\lambda\Lambda(1-a_m(t,A))w}{DA} - \dfrac{\alpha}{DA}wa_m(t,A) \\ \dfrac{db(t,A)}{dt} = 2g(1-b(t,A))^2 - \dfrac{b(t,A)}{T_S(t)} - \dfrac{2\alpha}{DA}b(t,A) \end{cases}$$

From this drift expression, by applying standard results, it is easy to prove the convergence to the mean field limit, from which the steady state expressions for a and b are obtained.

B. Proof of Lemma 2 (outline)

If g is the mean contact rate, of all contacts, those which may bring to an exchange of models are those in which both nodes are not busy. The probability of such an event is $(1-b)^2$. The mean amount of models received during an exchange is derived in a similar way as γ in Lemma 1.

C. Derivation of Theorem 1

We derive the drift, which describes the average local variation of observation availability over time. We assume the system satisfies the *homogeneous conditions*. That is, that at $t = 0$ the mean number of nodes per unit surface possessing a given model is the same for all models; that at any time instant, nodes possessing a given model are uniformly distributed within the region; and that the probability of a node to have that model is independent from the probability of any other node to have the same model.

Lemma 5. *When the system satisfies the homogeneous conditions, the* drift *of the CTMC is given by* $\frac{do(\tau,A)}{d\tau} = \frac{bS(a)}{T_S(a)}ao(\tau-d_M,A)(1-o(\tau-d_M,A))$*, with initial condition* $o(\tau,A) = 0$ *for* $\tau < d_I$*, and* $o(d_I,A) = \frac{1}{\lceil Na \rceil}$.

Proof. The derivation goes along the same lines as that of Lemma 1. Let $N_o(\tau,A)$ denote the number of nodes in the RZ possessing a model instance including the given observation, at a given time τ from observation generation, for a RZ area A. The rate at which this quantity varies over time is given by the sum of three components. The first is due to contacts between, on one side, a node with a model which incorporates the given observation, and a node possessing only the default model. The mean rate at which nodes exit the busy state is given by the ratio between the mean number of busy nodes at time τ in the region, given by $\bar{N}(A)b$, and the mean time taken by an exchange between two nodes, $T_S(a)$. Thus the rate of relevant events is $\frac{\bar{N}(A)b}{2T_S(a)}$. Moreover, let us consider one of these terminating exchanges. The probability that the model was transferred to the node possessing only the default model during such exchange is equal to the probability that the sender node had the model and that the receiving node did not have it, given by $2a(1-a)$, multiplied by the probability that the transferred model incorporated the given observation, $\frac{N_o(\tau,A)}{a\bar{N}(A)} = o(\tau,A)$, and by the probability that the model was transferred successfully, given by $S(a)$. Summing up, the first contribution is $\frac{\bar{N}(A)bS(a)}{T_S(a)}a(1-a)o(\tau,A)$. Similarly, for the second contribution, which accounts for the case in which the exchange is among two nodes each with a non-default model, we have $\frac{\bar{N}(A)bS(a)}{T_S(a)}a^2o(\tau-d_M,A)(1-o(\tau-d_M,A))$. This correspond to the rate at which nodes conclude the merging operation, after receiving the instance with the given observation. Thus the observation availability that matters is the one at the moment in which the exchange took place, i.e., at time $\tau - d_M$. The third contribution is given by nodes exiting the RZ, and it is given by the rate at which nodes exit the RZ (i.e., α) multiplied by the fraction of those exiting nodes which has the model and incorporates the observation (i.e., $ao(\tau,A)$). Putting all together, and normalizing, we get the final expression of the drift. □

The rest of Theorem 1 proof follows the standard procedure for mean field convergence, i.e., convergence of initial conditions, and condition on the drift of the Population Continuous

Time Markov Chain (PCTMC) associated to the system. As its drift is continuous for $\tau \geq d_I$, let $\bar{o}(\tau) = \lim_{R\to\infty} o(\tau, A)$. As our sequence of PCTMC models satisfies these properties, by Theorem 1 in [26] for any finite time horizon $T \leq \infty$, the sequence of population models associated to A converges *almost surely* to the dynamics of the ordinary differential equation in (5), which proves Theorem 1.

D. Derivation of Theorem 2

Proof. (sketch). For a given time t, let $i \in \mathbb{Z}^+$ be the label of the i-th last observation generated in the system, counting backward in time from t. We aim at computing the expectation of the time τ since the generation of the last observation incorporated in a given model. We have $E[\tau] = \frac{\sum_{i=1}^{\infty} E[\tau|i]P(i)}{\sum_{i=1}^{\infty} P(i)}$ where $E[\tau|i]$ is the expectation of τ conditioned on having observation 1 to $i-1$ not included in the model, and observation i included, while $P(i)$ is the probability of such an event. As arrivals are Poisson with intensity λ, $E[\tau|i] = \frac{i}{\lambda}$. $P(i)$ is computed taking into account that arrivals are independent, and that the probability of an observation generated x seconds ago to be included in a given model is $o(x)$. We have thus $F = \frac{\delta \sum_{i=1}^{\infty} i E_{\xi_1,\dots,\xi_i}\left[o(\gamma_i)\prod_{j=1}^{i-1}(1-o(\gamma_j))\middle|\gamma_i \leq \tau_l\right]}{\sum_{i=1}^{\infty} E_{\xi_1,\dots,\xi_i}\left[o(\gamma_i)\prod_{j=1}^{i-1}(1-o(\gamma_j))\middle|\gamma_i \geq \tau_l\right]}$. Equation (7) derives from applying Jensen's integral inequality to the above expression for staleness F. □

References

[1] M. H. C. Garcia, A. Molina-Galan, M. Boban, J. Gozalvez, B. Coll-Perales, T. Şahin, and A. Kousaridas, "A tutorial on 5g nr v2x communications," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1972–2026, 2021.

[2] R. Molina-Masegosa and J. Gozalvez, "LTE-V for sidelink 5G V2X vehicular communications: A new 5G technology for short-range vehicle-to-everything communications," *IEEE Vehicular Technology Magazine*, vol. 12, no. 4, pp. 30–39, 2017.

[3] IEEE Computer Society LAN MAN Standards Committee and others, "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," *ANSI/IEEE Std. 802.11-1999*, 1999.

[4] M. Woolley, "Bluetooth Core Specification Version 5.2 Feature Overview," *Bluetooth SIG: Kirkland, WA, USA*, 2020.

[5] R. I. Ansari, C. Chrysostomou, S. A. Hassan, M. Guizani, S. Mumtaz, J. Rodriguez, and J. J. Rodrigues, "5G D2D networks: Techniques, challenges, and future prospects," *IEEE Systems Journal*, vol. 12, no. 4, pp. 3970–3984, 2017.

[6] "Gigwalk: We've got your brand's back - Gigwalk," https://www.gigwalk.com/, (Accessed on 07/31/2022).

[7] "Driving directions, live traffic & road conditions updates - Waze," https://www.waze.com/en/live-map/, (Accessed on 07/31/2022).

[8] "Millionagents - marketing research for business on the basis of modern technologies," https://www.millionagents.com/en/, (Accessed on 07/31/2022).

[9] R. Ormándi, I. Hegedűs, and M. Jelasity, "Gossip learning with linear models on fully distributed data," *Concurrency and Computation: Practice and Experience*, vol. 25, no. 4, pp. 556–571, 2013.

[10] L. Wang, Z. Yu, D. Yang, T. Ku, B. Guo, and H. Ma, "Collaborative mobile crowdsensing in opportunistic d2d networks: A graph-based approach," *ACM Trans. Sen. Netw.*, vol. 15, no. 3, May 2019. [Online]. Available: https://doi.org/10.1145/3317689

[11] E. Hyytia, J. Virtamo, P. Lassila, J. Kangasharju, and J. Ott, "When does content float? Characterizing availability of anchored information in opportunistic content sharing," *IEEE INFOCOM*, April 2011.

[12] S. Ali, G. Rizzo, B. Rengarajan, and M. A. Marsan, "A simple approximate analysis of floating content for context-aware applications," *technical report, TR-IMDEA-Networks-2013-1*, 2013.

[13] G. Manzo, M. Ajmone Marsan, and G. Rizzo, "Performance Modeling of Vehicular Floating Content in Urban Settings," in *Teletraffic Congress (ITC 29), 2017 29th International*, vol. 1. IEEE, 2017, pp. 99–107.

[14] S. Lee, X. Zheng, J. Hua, H. Vikalo, and C. Julien, "Opportunistic Federated Learning: An Exploration of Egocentric Collaboration for Pervasive Computing Applications," in *2021 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2021, pp. 1–8.

[15] L. Giaretta and S. Girdzijauskas, "Gossip Learning: Off the Beaten Path," in *2019 IEEE International Conference on Big Data (Big Data)*, 2019, pp. 1117–1124.

[16] I. Hegedűs, G. Danner, and M. Jelasity, "Decentralized learning works: An empirical comparison of gossip learning and federated learning," *Journal of Parallel and Distributed Computing*, vol. 148, pp. 109–124, 2021.

[17] J. Daily, A. Vishnu, C. Siegel, T. Warfel, and V. Amatya, "GossipGraD: Scalable Deep Learning using Gossip Communication based Asynchronous Gradient Descent," *CoRR*, vol. abs/1803.05880, 2018. [Online]. Available: http://arxiv.org/abs/1803.05880

[18] M. Blot, D. Picard, N. Thome, and M. Cord, "Distributed optimization for deep learning with gossip exchange," *Neurocomputing*, vol. 330, pp. 287–296, 2019.

[19] Y. Chen, K. Yuan, Y. Zhang, P. Pan, Y. Xu, and W. Yin, "Accelerating Gossip SGD with Periodic Global Averaging," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 1791–1802.

[20] J. Guo, Y. Zuo, C.-K. Wen, and S. Jin, "User-Centric Online Gossip Training for Autoencoder-Based CSI Feedback," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 3, pp. 559–572, 2022.

[21] M. A. Dinani, A. Holzer, H. Nguyen, M. A. Marsan, and G. Rizzo, "Gossip Learning of Personalized Models for Vehicle Trajectory Prediction," in *2021 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, 2021, pp. 1–7.

[22] A. Chaintreau, J.-Y. Le Boudec, and N. Ristanovic, "The age of gossip: Spatial mean field regime," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 37, no. 1. ACM, 2009, pp. 109–120.

[23] J. Ott, E. Hyytiä, P. Lassila, T. Vaegs, and J. Kangasharju, "Floating content: Information sharing in urban areas," in *2011 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, March 2011, pp. 136–146.

[24] G. Friedland and M. Krell, "A capacity scaling law for artificial neural networks," *arXiv preprint arXiv:1708.06019*, 2017.

[25] S. Kaisar, J. Kamruzzaman, G. Karmakar, and M. M. Rashid, "Decentralized content sharing in mobile ad-hoc networks: A survey," *Digital Communications and Networks*, 2022.

[26] A. Kolesnichenko, V. Senni, A. Pourranjabar, and A. Remke, "Applying mean-field approximation to continuous time markov chains," in *ROCKS*, 2012.

[27] L. Bortolussi, J. Hillston, D. Latella, and M. Massink, "Continuous approximation of collective system behaviour: A tutorial," *Performance Evaluation*, vol. 70, no. 5, pp. 317–349, 2013.