DeepNDN: Opportunistic Data Replication and Caching in Support of Vehicular Named Data

DeepNDN: Opportunistic Data Replication and Caching in Support of Vehicular Named Data

Gaetano Manzo*†, Eirini Kalogeiton†, Antonio Di Maio‡, Torsten Braun†, Maria Rita Palattella§, Ion Turcanu‡, Ridha Soua‡, and Gianluca Rizzo*

*HES-SO Valais {name.surname}@hevs.ch †University of Bern {name.surname}@inf.unibe.ch
‡University of Luxembourg {name.surname}@uni.lu §LIST {name.surname}@list.lu

***Abstract*—Although many target applications in VANETs are information-centric, the performance of Named Data Networking (NDN) in vehicular ad-hoc networks is severely hampered by persistent network partitioning, typical of many vehicular scenarios. Existing approaches try to address this issue by relying on opportunistic communications. However, they leave open the crucial issue of how to guarantee content persistence and tight QoS levels while optimizing the resource utilization in the vehicular environment. In this work we propose DeepNDN, a communication scheme based on the joint application of NDN and of probabilistic spatial content caching, which enables content retrieval in fragmented and dynamic network topologies with tight delay constraints. We present a data-based approach to DeepNDN management, based on locally modulating content replication and delivery in order to achieve a target hit ratio in a resource-efficient manner. Our management algorithm employs a Convolutional Neural Network (CNN) architecture for effectively capturing the complex relations between spatio-temporal patterns of mobility and content requests and DeepNDN performance. Its numerical assessment in realistic, measurement-based scenarios suggest that our management approach achieves its target set goals while outperforming a set of reference schemes.**

I. Introduction

Vehicle-to-vehicle (V2V) communications are a key enabler of Intelligent Transportation Systems (ITSs) as well as of autonomous coordinated driving, enabling secure and efficient mobility services as well as a wealth of safety and entertainment applications. However the performance of these services, when relying on host-based communication models such as IP and on point-to-point information exchanges, is significantly affected by high mobility and dynamic network topologies, leading to unstable inter-vehicular connectivity, to packet loss and low service availability [1].

Among the candidate approaches to address these issues, of particular interest is Named Data Networking (NDN) [2], a communication paradigm based on in-network caching, and on information retrieval according to content name rather than host location. NDN takes advantage of content redundancy in the network to minimize the number of transmissions required to deliver a given content, potentially improving the effectiveness of content retrieval on the occurrence of network topology changes [3].

However, significant challenges stand in the way of an efficient and effective applicability of NDN in the vehicular domain [1]. Specifically, the fragmented structure of vehicular network topologies, typical of a large number of urban and extra-urban settings, limits the scope of NDN to single connected components. In addition, NDN fails to maintain stable paths between requesters and content sources in highly dynamic topologies and variable network densities [4]–[6].

For highly sparse and dynamic settings, in which store-carry-and-forward is the main mode of communication, a large number of Delay-Tolerant Networking (DTN) schemes for content distribution is available (e.g. [7] and reference therein). Among these, of special relevance are *probabilistic spatial content caching schemes* based on opportunistic content replication, such as Hovering Information [8], Anchored Information [9], and Floating Content (FC) [10]. Their goal is to achieve probabilistic content persistence and a target hit ratio within a predefined geographical area, while minimizing the amount of system resources employed [11]. However, being designed for sparse topologies, they are highly inefficient in contexts where node clusters are a significant portion of the network, and they do not support content retrieval with tight delay constraints.

A promising approach to address both the aforementioned shortcomings of NDN and the limitations of DTN schemes in vehicular settings is to combine NDN for intra-cluster content exchanges with Delay-Tolerant Routing (DTR) schemes for inter-cluster content dissemination [12]–[14]. However, several of these works apply to settings where networks have relatively stable topologies and mobility patterns, and they are thus unfit for vehicular networks. Moreover, they all leave open the crucial issue of how to orchestrate communications in order to deliver a given target performance (e.g. in terms of hit ratio and maximum delay) in a fragmented and dynamic network, in a resource-efficient manner.

The present work represents a first attempt at tackling these issues. We consider scenarios in which infrastructure support to V2V communications (in the form of collection of data on user mobility, and orchestration of content replication) is ubiquitous. The main contributions of this paper are:

- We propose DeepNDN, a communication scheme for content retrieval in vehicular networks, based on the joint application of NDN and of probabilistic spatial content caching, and capable of adapting to a wide range of network topologies and to very dynamic settings.

- We present a data-based approach for dynamic management of DeepNDN, which achieves a target minimum hit ratio in a resource-efficient manner, by proactively adapting the content replication and availability to local conditions. The approach employs a Convolutional Neural Network (CNN) architecture, in order to effectively capture the complex relations between spatio-temporal patterns of mobility and the performance of the content delivery service. A flexible cost function allows accounting for heterogeneity in the node population (e.g., in spatio-temporal patterns and resource availability) and for a variety of resource cost models.
- We assess numerically our approach on a set of realistic scenarios, showing that it substantially outperforms schemes based only on NDN, both in terms of resource efficiency and in the ability to satisfy tight QoS constraints.

The paper is structured as follows. Section II presents the system model, followed by the problem formulation in Section III. Our deep learning management algorithm is illustrated in Section IV and assessed numerically in Section V. In Section VI we review the state of the art. Finally, Section VII concludes the paper.

II. System Model

We consider a set of nodes (modeling vehicles and pedestrians) moving on a road grid. We assume that each node knows its position and embeds a wireless network interface (such as IEEE802.11p, IEEE802.11bd, Bluetooth, or cellular D2D [15]) to communicate directly with other nodes in its vicinity. We say that two nodes are in *contact* when they are in range to exchange directly information. In addition, each node is endowed with a cellular network interface. Coherently with the 5G (and beyond) paradigm, we assume that a coordination and management function (e.g., possibly implemented by a Software Defined Network controller - SDNC [16]) within the cellular access network periodically collects data about node mobility in order to optimize operations.

We assume that the road grid is partitioned into I road segments, generally of different size and shape (as shown in Figure 1b). The number and shape of each segment are based on the tradeoff between computational complexity and accuracy of our approach. Given a specific content, we consider a time interval T, denoted as *observation interval*, corresponding to the time period during which a population of users might request that content. With respect to a given content, we assume that each node is either *neutral*, when it does not possess the content nor it has requested it; *requester*, when it has requested the content but it does not possess it yet; or *producer*, when it possesses the content. A requester becomes a producer if it received the content by a *maximum delay* d, which is generally different for each content type. Its duration is a function of the specific application, beyond which the information requested is useless. Indeed, we assume that if the requester for a content does not get it by d seconds, it becomes neutral again.

All nodes entering the considered road grid are without any content. Every time a node enters a segment i, it becomes a requester with probability μ_i. This allows modeling a process of content requests which is driven by context. Indeed, the need for specific information (e.g., about a point of interest) is more likely to arise when people get close to it.

Table I
Main notation used in the paper

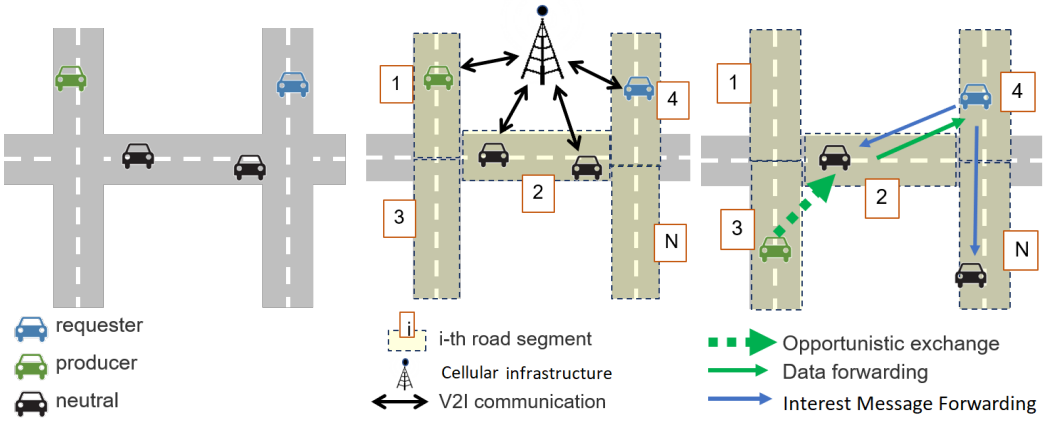
Name	Description
I	Total number of road segments in the grid
T	Observation interval
μ_i	Prob. for a node to become a requester in segment i
d	Maximum delay for a request to be satisfied
b_i	Replication probability in segment i
k_i	Caching probability in segment i
a_i	Mean content availability in segment i
N_i	Mean number of vehicles in segment i
h	Max number of hops to forward an Interest Message
γ_i	Total number of content transfers in segment i
r	Mean hit ratio

A. DeepNDN operation

In what follows, we describe the proposed DeepNDN communication paradigm, based on combining NDN routing mechanisms with strategies for probabilistic localized content caching. At the beginning of the observation interval, a non-empty subset of nodes in the scenario are producers (they might have produced the content themselves, or downloaded from the infrastructure), while the others are all neutral. We assume contents are identified unambiguously by their names.

When a node becomes a requester for a given content, if it is in contact with other nodes, it broadcasts an Interest Message (IM) for that content. Then every T_f seconds, if it has not received yet the content, it broadcasts again the IM to all nodes in its range. A node receiving the IM checks whether it possesses the content. If this is not the case, it records in the Pending Interest Table (PIT) the node from which it received the IM, and it rebroadcasts the IM. When the node from which it received the IM is not anymore in range, the corresponding entry in the PIT table is removed. When the IM reaches the producer for that content, the producer replies by sending back the requested content in a Data Message (DM), which is routed back to the requester by exploiting the information stored in the PIT of the nodes traversed by the IM. In order to limit the overhead and prevent broadcast storms, we assume that the IM cannot be forwarded to more than h-hops from the requester which originated it. We assume each IM is valid until d seconds have passed from the time at which the originating node became a requester. After d seconds, the IM is not forwarded anymore, and at each node, all the PIT entries corresponding to that IM are removed. In this work, our primary focus is on the content distribution. To reduce the effect of constant IM broadcasts, we limit the IM travel to h hops. This allows broadcasting to be limited only in particular clusters (areas) instead of occupying the whole network. Moreover, FC works in the premise of opportunistic communications, something that we suggest of doing via broadcasting of IMs.

In what follows, we assume that the content can be exchanged as the payload of a single layer 2 packet (i.e., as a single *Data Message*). When this is not the case, the content is exchanged as a sequence of data messages, each



(a) (b) (c)

Figure 1. DeepNDN Overview. a) Road grid and node states. b) Road grid partitioning, data collection and dissemination. c) Content retrieval.

retrieved and routed independently through the network. For ease of exposition, we assume that the Data Message can fit into a single frame. But when this is not the case, our approach allows the retrieval of several content objects related to the same Data message. When a producer, residing in the i^{th} road segment, comes in contact with a neutral node residing in the j^{th} road segment, the content is replicated to the neutral node with probability b_i. When a node in the i^{th} segment receives a content, it checks the PIT, and it forwards the content to the correspondent entries (as shown in Figure 1c). Then it keeps it with probability k_i. If the PIT is empty, and the node decides to keep the content, for each node with which it is in contact it replicates the content with probability b_i (independently for each node). Note that we assume content exchanges are always unicast (one-to-one). However, the proposed scheme can be easily extended to include the effects of multicasting and broadcasting.

Let $\boldsymbol{b} = \{b_i\} \in \mathbb{R}^n$, and $\boldsymbol{k} = \{k_i\} \in \mathbb{R}^n$. The set of parameters $(\mathbf{b}, \mathbf{k})$ completely describes a DeepNDN scheme for a given content, as each entry identifies the content replication and caching strategies over all segments during the whole observation interval. Parameters $(\mathbf{b}, \mathbf{k})$ determine the likelihood for the given content to persist probabilistically in the road grid for the whole duration of the observation interval.

A key performance metric for the DeepNDN scheme is the *hit ratio*, which is the fraction of requesters that get the content within the maximum delay.

III. Problem Formulation

Besides system parameters, the performance of the DeepNDN communication scheme critically depends on the values of the replication and caching probabilities associated with each road segment. Such a scheme requires a coordination and management function which, given a minimum hit ratio r and a maximum delay for a content, modulates $(\mathbf{b}, \mathbf{k})$ over space satisfying the given performance constraints and content persistence over the observation interval.

In this section, we perform a first step towards the design of such a management function. We formulate the problem of deriving the parameters $(\mathbf{b}, \mathbf{k})$ that achieve a target minimum hit ratio r_0 during the observation interval T while minimizing a cost function accounting for the amount of host resources (bandwidth and storage) employed by the scheme.

The cost function we adopt is the sum of two components. The first is a *user device storage* function $S(\boldsymbol{b}, \boldsymbol{k})$, and it accounts for the amount of storage at the user device employed by the scheme. Its expression is given by the sum, over all segments, of the mean number of producers in each segment, averaged over the observation interval, and multiplied by the content size L.

$$S(\boldsymbol{b}, \boldsymbol{k}) = L \sum_{i=1}^{I} N_i a_i(\boldsymbol{b}, \boldsymbol{k}) \tag{1}$$

where N_i is the mean amount of nodes on the i^{th} segment, while a_i is the *availability* of the given content in that segment, and it is equal to the fraction of nodes in that segment which possess the content.

The second component is the *user communication resources* function Γ:

$$\Gamma(\boldsymbol{b}, \boldsymbol{k}) = \sum_{i=1}^{I} \gamma_i(\boldsymbol{b}, \boldsymbol{k}) \tag{2}$$

where γ_i is the total number of content transfers in road segment i which have taken place during the observation interval T (a content transfer is counted as taking place in the segment of the transmitter node).

In the present work, we assume the content size to be typically several orders of magnitude larger than the size of control messages, as cellular network offloading is one of the main applications of the proposed scheme. The resulting cost function does not account for the amount of resource cost due to the exchange of IM, nor to the exchange of information implemented through the infrastructure (e.g., through the cellular network), for the collection of data on mobility, and for communicating the caching and replication parameters to each node at the beginning of the observation interval. However, our cost function (and our approach) can be easily generalized to account for such contributions.

Given the high volatility and dynamism of the settings to which DeepNDN applies, there is always a nonzero probability that the content disappears from the road grid before the end of the observation interval. This is undesirable in our scheme, as it would entail the use of communication and storage resources of the infrastructure (i.e., the cellular network) for reseeding the content and delivering it, and this would defeat the purpose of having DeepNDN in the first place. Let $P(T)$ denote the probability of content disappearance during an observation interval of T. Moreover, let r denote the mean hit ratio across the whole road grid, averaged over the observation interval. An optimal DeepNDN scheme is the solution of the following optimization problem:

Problem 1.

$$\min_{\boldsymbol{b},\boldsymbol{k}} \Gamma(\boldsymbol{b},\boldsymbol{k}) + \beta S(\boldsymbol{b},\boldsymbol{k}) \tag{3}$$

Subject to:

$$r \geq r^0 \tag{4}$$

$$P(T) \leq \epsilon \tag{5}$$

ϵ is the target maximum value of the probability of content disappearance, while coefficient β modulates the relative weight of the two cost components. By varying β it is possible to adapt the cost function to settings with different resource availability and to different incentive schemes to resource sharing and cooperation.

Note that the optimal strategy is derived by solving Problem 1 separately for each different content, as each comes with its own performance constraints (i.e., target hit ratio, maximum delay, and observation interval). However, our approach can be easily extended to a formulation that optimizes the overall cost for several contents.

Problem 1 involves performance parameters (such as mean content availability and the probability of content disappearance) whose dependence on system parameters and input constraints is complex to model analytically in a heterogeneous setting, without relying on strong assumptions that limit the accuracy of the resulting model.

To address this issue, in the next section, we present a data-based approach to the optimization problem, relying on the application of a CNN architecture.

IV. A Deep Learning algorithm for resource-efficient management

In this section, we describe our approach to solving Problem 1, based on a type of CNN which is adapted to model data with grid-like topology [17]. The choice of a deep learning approach is due to the high complexity of the relationships between the system parameters, the caching and replication strategies, and the performance of our NDN scheme in non-homogeneous settings. In addition, a data-based approach is made possible by a large amount of data (on its own operation and on user patterns of demand, of mobility) which is collected by the cellular infrastructure in the 5G paradigm and beyond. The specific choice of a CNN architecture is due to its ability to capture the complex relations between elements of a multidimensional set of system parameters. In particular, when applied to realistic, measurement-based scenarios, CNN has shown a higher accuracy and efficiency in capturing the correlations of spatial correlations (in our case, between spatial features of the road grid) than other learning approaches, such as Decision Tree or Random Forest [18].

We assume that the management function in the cellular infrastructure, on the basis of data collected on node mobility and on patterns of the content request, elaborates forecasts on the spatio-temporal patterns of the process of requests for each content, as well as on host resource availability, and mobility patterns. In this way, it identifies opportunities for resource optimization via content pre-fetching and caching at the user.

Then, possibly according to considerations of minimization of resource utilization for the cellular network, and of availability of resources (the specific criteria used is however out of the scope of the present work), it decides to offload pre-fetching and delivery of a set of contents to the DeepNDN communication scheme. The accuracy of such forecasts is out of the scope of this work. However, our solution accounts for data quality injected providing a conservative solution.

The setup and operation of our DeepNDN communication scheme is divided into three phases:

- **Data collection and elaboration**. The management function collects and records node mobility traces over time across the whole road grid, on a regular basis. The training set is generated, by associating the communication features to the measured mobility features. Finally, the CNN is trained.
- **Computation of a DeepNDN strategy**. When a decision of offloading the delivery of a content to the DeepNDN communication scheme is taken, the management function in the cellular infrastructure (e.g., an SDN controller) determines the target hit ratio, maximum delay, and operational interval, based on the application QoS requirements. Then it uses the trained CNN to compute in real-time a set of parameters $(\mathbf{b}, \mathbf{k})$ that allows satisfying the target minimum hit ratio r_0 and the constraint (5) on content absorption probability while minimizing the cost function.
- **Deployment**. The system provides to all nodes in the considered road grid, and to all those which enter the road grid during the observation interval, the parameters $(\mathbf{b}, \mathbf{k})$ which orchestrate the DeepNDN scheme for the given content. If mobility and/or request patterns deviate significantly from the forecasted ones during operations, new forecasts are elaborated, and a new set of coefficients are computed and adopted by all nodes in the road grid.

In what follows, we describe in detail the three phases, and the algorithms involved.

A. Data collection and elaboration

The system collects, on a regular basis, data about vehicle mobility in the road grid, recording the trajectories of each user, and the time and location in which it becomes a requester. Specifically, we assume that the system partitions time into intervals. Then starting from user trajectories, for each segment i and interval z, the system computes a set of aggregate metrics

Table II
FEATURES COLLECTED IN EACH SEGMENT.

Array	Notation	Feature
Mobility	$\mathbf{m}$	average number of nodes average node speed [m/s] average number of nodes in contact mean request arrival rate [s^{-1}]
Communications	$\mathbf{c}$	average number of requesters average number of producers average number of transmitting nodes mean hit ratio [s^{-1}] mean request delay [s]

relative to node mobility and to wireless communications. With $\mathbf{m}_z$, we denote the *mobility array* for the z^{th} time interval, consisting of the values of these aggregate parameters for the whole grid. For each segment, the mobility array contains the average node speed, the average number of nodes, the average number of nodes that, in a given time instant, are in contact with a given node (the list of vehicles able to exchange beacons according to the given communication protocol [19]), and the mean rate at which nodes become requesters in the given segment. Such a choice of parameters represents only one of many possibilities. However, these parameters have been chosen as they are typically used as input to the main existing models (both analytic and data-based) of probabilistic content caching based on opportunistic replications [20]–[22]. Moreover, such a choice has shown in our evaluations to enable a high degree of accuracy in accounting for the peculiarities of node mobility patterns, and their effects on the performance of content delivery.

1) Label generation: In this step, a randomization procedure is applied over the collected data. Specifically:

- For each $\mathbf{m}_z$, a set of random schemes $(\mathbf{b}_j, \mathbf{k}_j)$ $j = 1, ..., J$ is generated. In addition, a random process of request arrivals is generated, based on the measured average rates.
- For each set of parameters $(\mathbf{m}_z, \mathbf{b}_j, \mathbf{k}_j)$, a simulation is performed based on the random strategy $(\mathbf{b}_j, \mathbf{k}_j)$, the user trajectories during the observation interval, and the given request arrival process. The parameters measured, for each segment, are the mean number of requesters present in the segment, the mean hit ratio, the mean number of producers, the mean time required to satisfy a content request, and the mean number of nodes that are transmitting at a given time instant. These parameters constitute the *communications feature vector* $\mathbf{c}_{j,z}$, and they are the basis for the estimation of the resource utilization and their associated costs.

For each j, z, $(\mathbf{m}_z, \mathbf{c}_{j,z})$ denotes the *Segment Features Vector* associated with $(\mathbf{b}_j, \mathbf{k}_j)$ and time interval z (Table II). Finally, each vector $(\mathbf{m}_z, \mathbf{c}_{j,z})$ is normalized, in order to avoid numerical issues in the subsequent phases of the process. To create the data set containing unbiased quadruplets $(\mathbf{m}_z, \mathbf{c}_{j,z}, \mathbf{b}_j, \mathbf{k}_j)$, which not differ in any systematic way, we apply a covariate adaptive randomization [23], which uses the method of minimization by assessing the imbalance of sample size among several covariates. This technique ensures no a priori knowledge of group assignment and it prioritizes the configuration sampling over the feature. Indeed, for a given feature set $\mathbf{m}_z$ several configurations $(\mathbf{b}_j, \mathbf{k}_j)$ are tested with respect to the covariance of the measured hit ratio and mean time for satisfying a request. Once collected such dataset, it has been split into a training set and a test sets using the stratified technique [24], [25], which holds the same ratio of classes across each split. In order to avoid numerical issues, we have normalized the data set and filled missing data with their median values. The output of this phase is thus a data set composed by quadruplets $(\mathbf{m}_z, \mathbf{c}_{j,z}, \mathbf{b}_j, \mathbf{k}_j)$, $\forall z, j$. Note that this phase can be performed entirely offline.

This data set is enriched over time with new elements, as long as the system measures new patterns of requests and of mobility and it performs new simulations. Note that, the contribution to the given data set of those elements derived through simulations plays a key role in enabling the system to effectively configure a DeepNDN scheme for relatively infrequent scenarios, such as road accidents or disasters, for which few measured data are typically available.

B. Computation of a DeepNDN strategy and deployment

As already stated, we assume that the management function in the cellular infrastructure is constantly monitoring mobility patterns and requests for contents. Based on these data, it is also computing forecasts, in order to decide which contents to offload to direct device-to-device communications. The mechanism by which these forecasts are implemented is out of the scope of the present paper. For each of those contents to offload, the management function establishes a time interval during which the offloading should be active, and it elaborates detailed forecasts of mobility patterns and requests. In addition, it establishes a target hit ratio and a maximum delay for receiving a requested content based on the application requirements. Let $\mathbf{m}$ be the forecasted mobility feature array for the given observation interval. Given these inputs, the CNN computes the replication and caching strategies $(\mathbf{b}, \mathbf{k})$ that achieve the target hit ratio while minimizing the resource cost for the whole observation interval. These parameters are then communicated to all nodes in the scenario. In those cases in which, during the operation of our scheme, new forecasts of mobility become available, the system computes a new strategy $(\mathbf{b}, \mathbf{k})$ for the remaining portion of the observation interval based on such forecasts, and it injects the new values of replication and caching parameters to all the nodes. Similarly, if patterns of mobility and of content request vary significantly during the duration of the observation interval, the management function may split such interval in sub-intervals (defined in such a way as to have a sufficient uniformity of the mobility patterns within each of them), and apply the dimensioning procedure described to each of them.

C. Convolutional Neural Network Architecture

In this section, we describe the overall structure of the CNN at the basis of our DeepNDN dimensioning approach. Figure 2 presents the structure and operation of our CNN. The number of layers in each of the three steps determines

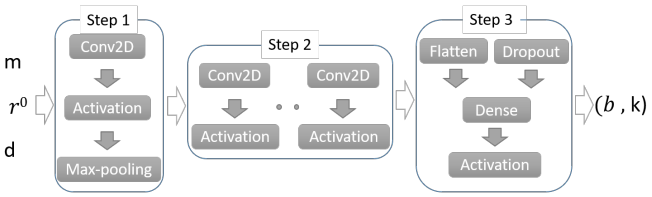


Figure 2. Architecture of the Convolutional Neural Network used to compute resource-efficient replication and caching strategies $(\mathbf{b}, \mathbf{k})$ for the DeepNDN communication scheme. Its inputs are the mobility features m, the target hit ratio r^0, and the maximum delay d

the *depth* of the learning architecture, and hence the level of complexity of the correlations which it is able to capture. Such correlations between different road segments are the result of wireless propagation effects and of the spatio-temporal patterns of mobility and of content request, and of their mutual interactions. Indeed, the ability of CNN to capture both intra-segment and inter-segment relations between features, and in particular, to model complex correlations even among segments which are distant in space, makes such a learning approach a particularly good fit for the problem of optimally orchestrating a DeepNDN scheme in realistic settings.

In step 1, our CNN learns local features correlations, such as the spatio-temporal correlations between speed and node density or the ratio between requester and producer. More specifically, the convolution layer (Conv2D) captures the correlation between features of the same road segment; the Activation layer selects the most relevant local features; finally, the max-pooling layer enhances computational performance by reducing the data dimensionality [17]. Its output are the most relevant local features per road segment.

The goal of step 2 is to enable the CNN to learn the structure of complex inter-segment dependencies, particularly those involving segments which are distant in space among them, thus improving model accuracy. Therefore, step 2 is composed by multiple instances of the convolutional layer, each followed by an activation layer, which selects the most relevant inter-segment features. The choice of the number of Conv2D layers, as well as of the kernel size used in these layers, depends on a tradeoff between computational complexity and accuracy of the CNN.

Step 3 processes the previous results in order to reduce data dimensionality and produce the strategies for DeepNDN management. It is composed by a Flatten layer for data dimensionality reduction, by a Dropout layer which contributes to avoiding overfitting issues, and by a Dense layer which reshapes data. The last layer is an Activation layer, which selects the best strategy among a set of possible candidates, providing as output the coefficients $(\mathbf{b}, \mathbf{k})$. The number of iterations (i.e., epochs) of the above three steps has been set according to the CNN learning rate derivative [17]. A crucial aspect of the performance of our approach is the computational load required by the three phases which compose it. Given the potentially large amount of data to be collected and pre-processed, the data collection is the most computationally intensive. The amount of computation required is directly related to the size of the training set, as well as to the number of segments and features considered. In order to reduce such a computational load, we have adopted a discretization of the caching and replication parameters, casting Problem 1 into a classification problem [17]. However, all the operations in this phase can be executed offline (i.e., before the need for content offloading and pre-fetching arises). Hence, their computational complexity has no impact on the performance of the scheme. The computations required to derive the DeepNDN strategy for a specific content are performed at the moment in which an offloading decision is made by the infrastructure. As this decision is not made in real-time but based on forecasts (of say, a few hours at least), it does not have an impact on the time required to implement the caching and replication strategies.

V. Numerical Evaluation

In this section, we numerically assess the performance of our approach to management of the DeepNDN communication scheme, in both synthetic and measurement-based scenarios, and we characterize the spatio-temporal strategies emerging from our deep-learning-based management approach. Finally, We compare our approach with vanilla NDN mechanism and NDN full supported by opportunistic communications.

A. System setup

We assume nodes embed the IEEE 802.11p wireless access protocol [15] with a maximum transmission power of 20 mW, a minimum signal attenuation threshold of −89 dBm, and a minimum path loss coefficient of 2, which correspond to typical settings in a vehicular environment [15]. We simulate content exchanges among nodes using Veins on OMNeT++ simulator [26], [27], whereas we use SUMO [28] for the vehicular mobility simulation over a road grid. For each segment, mobility features were measured using a sampling interval of 1 ms, in order to accurately capture the dynamics of mobility and content diffusion. In both scenarios, we used a training set size of $2 \cdot 10^5$, which has proven sufficient to achieve a high level of accuracy, and we performed a 10-fold cross-validation. For testing, we considered a test set size of $5 \cdot 10^3$, sufficient to achieve confidence intervals of at most 3 % in all settings considered. Unless stated otherwise, the coefficient β in the cost function in Problem 1 has been set to 1, in order to give equal weight to storage and communication costs. The default values of target hit ratio (0.9) and observation interval (1800 s) have been chosen to model applications with tight performance requirements, and in which the DeepNDN scheme must be effective over short time intervals.

B. Baseline scenario

In the first set of simulations, we considered a Manhattan grid of 10 by 10 square blocks, each of side 100 m. The partition into road segments has been such that every portion of road between two consecutive crossroads consists in two road segments (one per each roadway), plus a segment covering the center of each crossroad, for a total of 460 road segments. In this scenario,

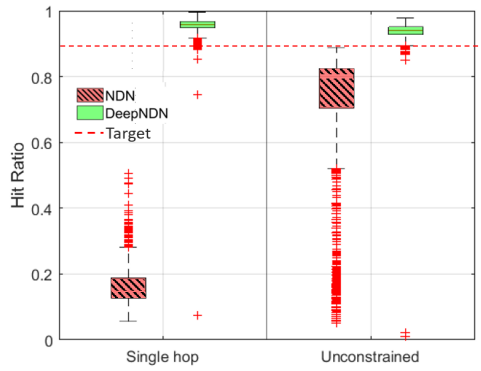


Figure 3. DeepNDN strategy, NDN scheme, for single-hop interest message forwarding and for the case without limitations. Manhattan setup.

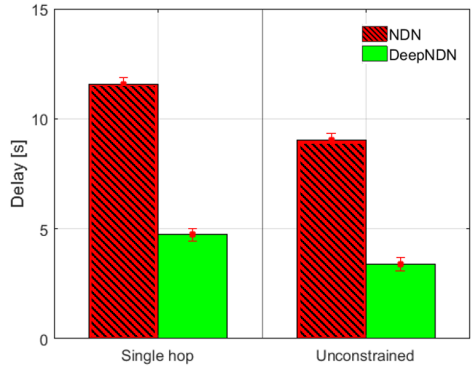


Figure 4. DeepNDN strategy, NDN only scheme, for single-hop IM forwarding and for IM forwarding without limits. Manhattan setup. Maximum delay 5 s.

nodes model vehicles moving according to the Manhattan Mobility Model with a turn probability of 0.25, equal for both left and right turns [29], and a speed uniformly distributed between 30 km/h and 50 km/h. Nodes enter the road grid from segments at the border, at a rate of $0.0024\,\mathrm{s}^{-1}$ per border segment. This resulted into an average of 100 vehicles in the whole scenario, with a mean sojourn time of 92.2 s. Throughout all the simulations, these settings never gave rise to a network topology consisting of a single connected component. Instead small, short lived clusters periodically formed at each crossroad, due to vehicles queuing. At the beginning of the observation interval, we assumed that each node has a 0.1 probability to possess the content, and that there are no requesters. Then during the observation interval, every time that a neutral node entered the road grid, it became a requester with a probability of 0.9. We use one content object, e.g. one Data Message, to model the exchange of short video trailer or of a multimedia advertisement. We assume that there is no need for fragment the content into multiple Data messages, hence, one IM can be satisfied by a single Data message. We leave the scenario with multiple fragmented messages for the same content object for future work. The maximum request delay has been set to 5 s, to model applications which rely on very short-term predictions of content requests.

For what concerns the propagation of the IM, we have considered two configurations. In a first one (denoted as *unconstrained*) there has been no limitations to the maximum number of hops which an IM can be forwarded. In addition, we considered the case in which an IM can only be forwarded from the requester to nodes that are in contact with it, i.e. we set the IM max number of hops to 1 (the *single hop* strategy). This second strategy aims at minimizing the communication overhead due to forwarding of IM, which in scenarios with large node clusters may jeopardize the communication channels, significantly affecting the performance of our scheme.

Table III
MEAN RESOURCE UTILIZATION OF DEEPNDN IN THE MANHATTAN SCENARIO. 98% CONFIDENCE INTERVAL OF 3%.

Forwarding Strategy	Mean Availability	Transmitting Nodes
Single hop	0.49	8.2
Unconstrained	0.32	8.3

As the box-plots in Fig. 3 and Fig. 4 show, in both configurations our DeepNDN management strategies are able to satisfy the target hit ratio with the given constraint on maximum delay. In particular, these results show that our approach is able to tune content replication and caching in such a way as to bring the likelihood to not satisfying the performance constraints to very low values (in over $5 \cdot 10^3$ simulations, less than 3 % violated this constraint) while using only a small portion of system resources. Table III shows the availability (mean fraction of nodes with content), as well as the mean number of simultaneous transmissions, for the two considered configurations. As results show, restricting IM forwarding to a single hop does not affect the ability of our approach to satisfy the target performance, though it comes at the cost of a 53% increase in the storage resources used.

The fact that our paradigm is effective in our scenario even when limiting the IM forwarding to a single hop might lead to assume that the mechanisms for NDN content retrieval alone could be sufficient to achieve the target performance, and that the contribution to the performance of DeepNDN scheme given by opportunistic content replication plays only a marginal role. In order to verify this, we have considered the performance of another scheme, denoted as NDN, and derived by the original DeepNDN scheme by setting to zero the opportunistic replication parameters and to one the caching probability. In this scheme, the content is cached only when nodes route it from the source to the requester. As Fig. 4 show, the NDN scheme systematically fails to deliver the content within the maximum delay of 5 seconds. In addition, in Fig. 3 we see that even when assuming no limitations to the delay with which a request can satisfied, and even when caching all contents forwarded, content delivery based only on NDN is not able to achieve the target hit ratio. In addition, while no content disappearance has been observed with the DeepNDN scheme, the percentage of contents disappeared from the scenario before the end of the observation interval in the NDN scheme has been of 11 % for the unconstrained case, and of 87 % in the single hop case.

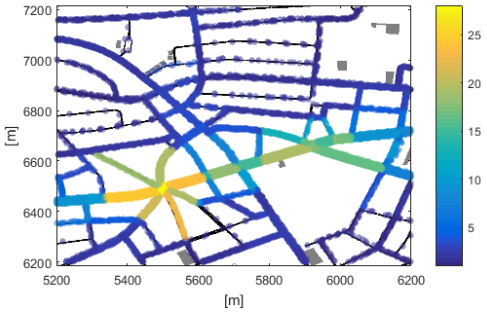


Figure 5. Mean number of neighbors per node. Luxembourg City.

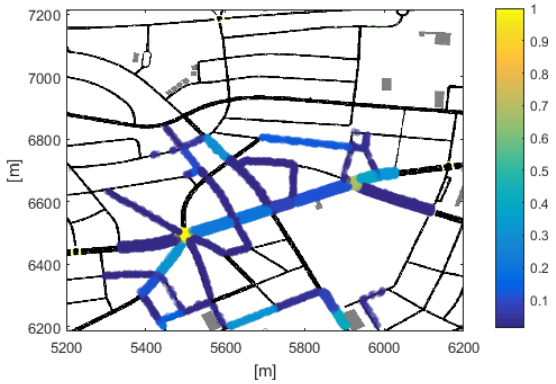


Figure 6. Density Map of requesters. Luxembourg City setup, 7 AM - 7:30 AM. The Point of Interest is the region in light yellow.

C. Luxembourg Scenario

In order to perform a more realistic assessment of the performance of our approach, we considered a scenario corresponding to a square area of side 1 km in the center of Luxembourg City (Fig. 5). The road grid, as well as its partition in road segments, is derived from OpenStreetMap [30]. The partition is performed in a similar way as in the Manhattan grid, with at least two segments for each portion of the road between two crossroads, for a total of 234 segments. The measurement-based vehicular mobility traces have been derived from [31], and they refer to a time interval with rush traffic (7 AM to 7:30 AM), with an average of 82.7 nodes present in the area in the considered operational interval. Figure 5 shows the mean number of nodes which are in contact with a node in the considered area. The map indicates that in the given time interval, despite the rush hour, a high density of nodes (and a potentially a high likelihood of clustering) is present only in limited portions of the considered area.

Unlike the baseline scenario, to make the considered scenario more realistic, we have assumed that a point of interest (e.g., a cinema) is present in the map (light yellow spot in Fig. 5), and that the likelihood of becoming a requester when entering a given segment increases when the distance from the point of interest decreases. The resulting average spatial distribution of requesters is shown in Fig. 6.

In such a scenario, we have evaluated the performance of our DeepNDN strategy with a maximum request delay of 10 s, and we have compared it with the NDN strategy, in which no

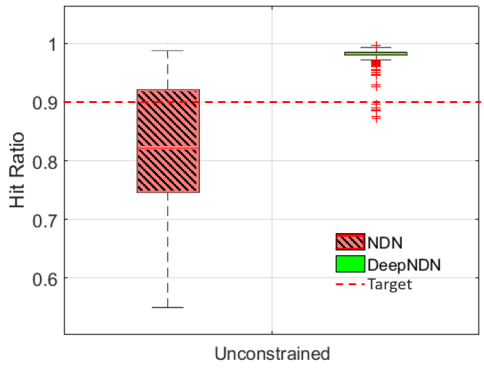


Figure 7. DeepNDN strategy, NDN only scheme, and without limit hops.

Table IV
RESOURCE UTILIZATION LUXEMBOURG SCENARIO. 98 % CONFIDENCE INTERVAL OF 3 %.

Approach	**Mean Availability** (%)	**Transmitting Nodes** (%)
NDN	52.5	43.9
All-on	62.4	44.7
DeepNDN	30.8	46.7

opportunistic content replication takes place. In both strategies, no limit has been set on IM forwarding. In this setup, our DeepNDN strategy managed to achieve a hit ratio larger than 0.9 in all the simulations. In the NDN only scheme instead, and despite the highly favourable conditions for NDN performance (in terms of correlation in space between zones of a high user density and zones with high density of requesters), only 18.3% of requesters received the content within the maximum delay. This result suggest that in realistic vehicular scenarios, and even in conditions of peak vehicle density and high node clustering around the point of interest, only a combination of NDN with a strategy for opportunistic content replication and caching allows achieving the target minimum hit ratio. In order to gain insights into this result, in Fig. 8, we have put in evidence those road segments where contents are exchanged only through opportunistic replication (in green), only through NDN (in blue), and with both mechanisms (in cyan). The figure shows that in those regions which surround the area with a high density of requesters, content is retrieved based only on opportunistic replications. In these areas, opportunistic replications have also the function of increasing the amount of content redundancy around the point of interest, by delivering content to those users who are approaching it. Instead, in those segments in which nodes tend to form clusters, and particularly around the point of interest, content is delivered through a combination of NDN and opportunistic content replications, instead of only via NDN. In these segments, the role of opportunistic replications is to maintain a sufficiently high amount of content redundancy for the NDN-based content retrieval to be effective within the given maximum retrieval delay. In addition, they make sure that the likelihood of content disappearance remains very low throughout the scenario.

Table IV reports, for the Luxembourg scenario, the amount

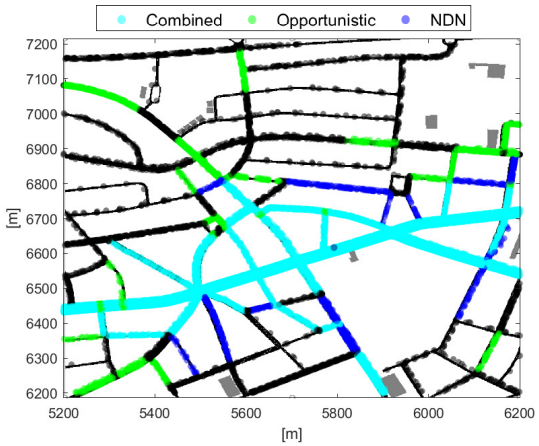


Figure 8. Opportunistic replication (in green), only through NDN (in blue) and with both mechanisms (in cyan). Luxembourg City, 7 AM - 7:30 AM.

of resources employed by our DeepNDN scheme, as well as by the NDN only scheme. In addition, we have also considered the *all-on* scheme, derived by DeepNDN by setting all caching and replication coefficients are set to one. As we have verified, both the DeepNDN and the all-on approach satisfy the performance targets in the considered scenario. However, our deep learning based management approach is able to achieve those performance goals while substantially decreasing the amount of content redundancy required in the network with respect to both all-on and NDN only schemes. Note that communication costs of the all-on scheme are substantially similar to the DeepNDN case, as in the all-on case a higher availability brings a lower rate of content transfers, seen that a higher fraction of the opportunistic contacts are among nodes which have the content.

VI. Related Work

A number of works propose approaches which enable NDN to cope with some of the effects of node mobility and of network fragmentation. [5], [32] focus on the specific issue of receiver and source mobility, while [6] considers disruptions due to changes in the network topology. [14] proposes a version of NDN based on epidemic forwarding which proves effective in sparse scenarios. These approaches apply to scenarios with relatively infrequent configuration changes, and they imply a significant increase in communication overhead. Moreover, they leave open the issue of content retrieval across node clusters and in sparse settings, which is the focus of the present work.

Ascigil et al. [33] developed an opportunistic NDN content discovery, which allows nodes to localize cached copies of data in off-path vehicles using the trails left from data messages when delivered from the data source to the requesters. Gündoğan et al. [34] introduce QoS support for NDN networks by proposing a technique to regulate the NDN tables and probabilistic-caching parameters to favour either prompt or reliable content forwarding. The study discussed in [12] considers disaster scenarios, with a network topology which, though fragmented, is relatively stable. It proposes a scheme which integrates NDN and DTN routing with the notion of node clusters and of "data mules" for inter-cluster content retrieval. [13] proposes a protocol stack integrating the NDN and DTN architecture, addressing some interoperability issues. These solutions however do not address the issue of how to guarantee a target performance (in terms of delivery ratio and maximum delay) in a fragmented and dynamic network, in a resource-efficient manner. Our proposed scheme tackles these issues by adapting to dynamic network topologies and managing network resources in an effective way.

In vehicular NDN, several studies exploit the fixed network infrastructure in order to cope with intermittent connectivity between vehicles [35]–[38]. Wang et al. [35] propose a Road Side Unit (RSU) controlled traffic information dissemination system, in which RSUs are used to find alternative paths when there is no connectivity between vehicles. Authors in [36] employ RSUs for sending the IM and retrieving the DM. [37] exploits the global network topology view that the infrastructure can provide to proactively inject the content on selected vehicles in order to improve the overall content delivery. In [39], the infrastructure is used for exchanging content among different areas, while the NDN mechanism is used to deliver the content object within each area. In all these approaches however, the infrastructure caches and directly delivers content to at least a portion of the users. The fraction of content deliveries performed by the infrastructure depends on the specific approach and setting considered, taking a potentially heavy toll on infrastructure resources. Instead, in our approach content delivery is delegated completely to vehicular ad-hoc communications. Authors in [38] propose the concept of *vehicular micro clouds*, clusters of vehicles cooperating to provide services and resources, similar to FC. [40] proposes a content-centric dissemination scheme. Its solution is based on a policy which sets the order of content exchanges on a contact between two nodes, and the probability for a node to drop a content in a way which tries to maximize the total delivery rate over a set of contents of different popularity. In Grassi et al. [41], the considered region is partitioned into areas. The proposed solution connects the producer and requester local areas by geocasting the content object in a delay-tolerant manner. These approaches however do not optimize resource usage, and they do not address the issue of content persistence within the given network. In [32], FC is used together with NDN in order to address the specific issue of mobility of producer. However, no global resource optimization is performed, and the impact of content floating on the overall performance of the content delivery process is not considered. In our work, we exploit the global knowledge of the infrastructure to optimize content replication and caching probabilities over time, ensuring content persistence in the given area.

VII. Conclusion

In this work we have proposed DeepNDN, a communication scheme based on the joint application of NDN and of probabilistic spatial content caching, for content retrieval in fragmented and dynamic vehicular ad-hoc networks. We have presented a data-based approach for dynamic management of DeepNDN,

capable of achieving a target hit ratio in a resource-efficient manner, by locally modulating the content diffusion process. As a followup, we intend to extend our management approach to include the resource cost of infrastructure support, and to scenarios with heterogeneous node populations, including cars, drones and pedestrians. Moreover, we want to extend DeepNDN in several vehicular scenarios such as highway, residential district, and industrial area.

VIII. Acknowledgment

This work was undertaken under the CONTACT project, CORE/SWISS/15/IS/10487418, funded by the National Research Fund Luxembourg (FNR), and by the Swiss National Science Foundation (SNSF), project no. 164205. This work was partially supported by Hasler MOBNET, and by COST RECODIS projects.

References

[1] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang, et al., "Named data networking," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 66–73, 2014.

[2] C. Felipe, A. Boukerche, L. Villas, A. Viana, and A. Loureiro, "Data Communication in VANETs: A Survey, Challenges and Applications," *Ad Hoc Networks*, Mar. 2014.

[3] G. Tyson, J. Bigham, and E. Bodanese, "Towards an information-centric delay-tolerant network," in *2013 IEEE Conference on Computer Communications Workshops*, IEEE, 2013, pp. 387–392.

[4] J. M. Duarte, T. Braun, and L. A. Villas, "MobiVNDN: A distributed framework to support mobility in vehicular named-data networking," *Ad Hoc Networks*, vol. 82, pp. 77–90, 2019.

[5] M. F. Al-Naday, M. J. Reed, D. Trossen, and K. Yang, "Information resilience: source recovery in an information-centric network," *IEEE network*, vol. 28, no. 3, pp. 36–42, 2014.

[6] V. Sourlas, L. Tassiulas, I. Psaras, and G. Pavlou, "Information resilience through user-assisted caching in disruptive content-centric networks," in *IFIP Networking*, IEEE, 2015, pp. 1–9.

[7] A. McMahon and S. Farrell, "Delay-and disruption-tolerant networking," *IEEE Internet Computing*, vol. 13, no. 6, pp. 82–87, 2009.

[8] A. A. V. Castro, G. Di Marzo Serugendo, and D. Konstantas, "Hovering Information - Self-Organising Information that Finds Its Own Storage," in *IEEE SUTC*, 2008, pp. 193–200.

[9] E. Hyytiä, J. Virtamo, P. Lassila, J. Kangasharju, and J. Ott, "When does content float? Characterizing availability of anchored information in opportunistic content sharing," in *INFOCOM*, IEEE, Apr. 2011, pp. 3137–3145.

[10] J. Ott, E. Hyytia, P. Lassila, T. Vaegs, and J. Kangasharju, "Floating content: Information sharing in urban areas," in *PerCom 2011*, 2011, pp. 136 –146.

[11] G. Manzo, M. A. Marsan, and G. A. Rizzo, "Analytical models of floating content in a vehicular urban environment," *Ad Hoc Networks*, vol. 88, pp. 65–80, 2019.

[12] E. Monticelli, B. M. Schubert, M. Arumaithurai, X. Fu, and K. Ramakrishnan, "An information centric approach for communications in disaster situations," in *LANMAN*, IEEE, 2014, pp. 1–6.

[13] H. M. Islam, A. Lukyanenko, S. Tarkoma, and A. Yla-Jaaski, "Towards disruption tolerant ICN," in *ISCC*, IEEE, 2015, pp. 212–219.

[14] Y.-T. Yu, J. Joy, R. Fan, Y. Lu, M. Gerla, and M. Sanadidi, "DT-ICAN: A disruption-tolerant information-centric ad-hoc network," in *2014 IEEE Military Communications Conference*, IEEE, 2014, pp. 1021–1026.

[15] D. Jiang and L. Delgrossi, "IEEE 802.11p: Towards an International Standard for Wireless Access in Vehicular Environments," Jun. 2008.

[16] B. Nunes, M. Mendonca, X. Nguyen, K. Obraczka, and T. Turletti, "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks," *Communications Surveys Tutorials, IEEE*, vol. PP, no. 99, pp. 1–18, 2014.

[17] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press Cambridge, Mar. 2016.

[18] G. Manzo, S. Otálora, T. Braun, M. Ajmone Marsan, G. Rizzo, and H. Nguyen, "DeepFloat: Resource-Efficient Dynamic Management of Vehicular Floating Content," in *ITC 31*, 2019.

[19] M. Torrent-Moreno, J. Mittag, P. Santi, and H. Hartenstein, "Vehicle-to-Vehicle Communication: Fair Transmit Power Control for Safety-Critical Information," *IEEE Transactions on Vehicular Technology*, 2009.

[20] G. Manzo, R. Soua, A. Di Maio, T. Engel, M. R. Palattella, and G. Rizzo, "Coordination Mechanisms for Floating Content in Realistic Vehicular Scenarios," in *IEEE MobiWorld*, 2017.

[21] G. Manzo, M. A. Marsan, and G. Rizzo, "Performance modeling of vehicular floating content in urban settings," in *29th International Teletraffic Congress (ITC 29)*, IEEE, vol. 1, Sep. 2017, pp. 99–107.

[22] G. Manzo, J. S. Otalora, M. A. Marsan, and G. Rizzo, "A Deep Learning Strategy for Vehicular Floating Content Management," *ACM SIGMETRICS Performance Evaluation Review*, vol. 46, no. 3, pp. 159–162, Jan. 2019.

[23] N. Scott, G. Mcpherson, C. Ramsay, and M. Campbell, "The method of minimization for allocation to clinical trials. A review," *Controlled clinical trials*, Jan. 2003.

[24] M. Shahrokh Esfahani and E. R. Dougherty, "Effect of separate sampling on classification accuracy," *Bioinformatics*, Nov. 2013.

[25] "How many stratification factors are "too many" to use in a randomization plan?" *Controlled Clinical Trials*, 1993.

[26] C. Sommer, R. German, and F. Dressler, "Bidirectionally coupled network and road traffic simulation for improved IVC analysis," IEEE Transactions on Mobile Computing, 2011.

[27] A. Varga and R. Hornig, "An Overview of the OMNeT++ Simulation Environment," ICST Simutools, 2008.

[28] D. Krajzewicz, G. Hertkorn, C. Rössel, and P. Wagner, "SUMO (Simulation of Urban MObility), an open-source traffic simulation," MESM, 2002.

[29] A. Hanggoro and R. F. Sari, "Performance evaluation of the Manhattan mobility model in vehicular ad-hoc networks for high mobility vehicle," in *IEEE COMNETSAT*, 2013.

[30] OpenStreetMap contributors, *Planet dump retrieved from https://planet.osm.org*, https://www.openstreetmap.org, 2017.

[31] L. Codeca, R. Frank, and T. Engel, "Luxembourg SUMO Traffic (LuST) Scenario: 24 hours of mobility for vehicular networking research," IEEE VNC, 2015.

[32] J. M. Duarte, T. Braun, and L. A. Villas, "Source mobility in vehicular named-data networking: An overview," in *Ad Hoc Networks*, Springer, 2018, pp. 83–93.

[33] O. Ascigil, V. Sourlas, I. Psaras, and G. Pavlou, "Opportunistic off-path content discovery in information-centric networks," in *2016 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)*, 2016, pp. 1–7.

[34] C. Gündoğan, J. Pfender, P. Kietzmann, T. C. Schmidt, and M. Wählisch, "On the impact of QoS management in an Information-centric Internet of Things," *Computer Communications*, vol. 154, pp. 160 –172, 2020.

[35] S. Wang, J. Deng, W. Wu, and J. Zhou, "RSU Controlled Named Data Networking for Traffic Information Dissemination in Vehicular Networks," in *IEEE SmartWorld*, IEEE, Oct. 2018.

[36] E. Kalogeiton and T. Braun, "Infrastructure-Assisted Communication for NDN-VANETs," in *WoWMoM*, IEEE, 2018, pp. 1–10.

[37] I. Turcanu, T. Engel, and C. Sommer, "Fog Seeding Strategies for Information-Centric Heterogeneous Vehicular Networks," in *IEEE VNC*, Los Angeles, CA: IEEE, Dec. 2019, pp. 282–289.

[38] T. Higuchi, G. S. Pannu, F. Dressler, and O. Altintas, "Content Replication in Vehicular Micro Cloud-based Data Storage: A Mobility-Aware Approach," in *VNC*, Taipei, Taiwan: IEEE, Dec. 2018.

[39] X. Wang and X. Wang, "Vehicular Content-Centric Networking Framework," *IEEE Systems Journal*, vol. 13, no. 1, pp. 519–529, Mar. 2019.

[40] F. Neves dos Santos, B. Ertl, C. Barakat, T. Spyropoulos, and T. Turletti, "Cedo: Content-centric dissemination algorithm for delay-tolerant networks," in *ACM MSWIM*, ACM, 2013, pp. 377–386.

[41] G. Grassi, D. Pesavento, G. Pau, L. Zhang, and S. Fdida, "Navigo: Interest forwarding by geolocations in vehicular Named Data Networking," in *WoWMoM*, IEEE, Jun. 2015, pp. 1–10.