IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 11 November 2022, accepted 11 December 2022, date of publication 15 December 2022,
date of current version 21 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3229645

RESEARCH ARTICLE

Proactive Edge Caching in Vehicular Networks: An Online Bandit Learning Approach

QIAO WANG, (Student Member, IEEE), AND DAVID GRACE, (Senior Member, IEEE)
Communication Technologies Research Group, Institute for Safe Autonomy, University of York, YO10 5DD York, U.K.
School of Physics, Engineering and Technology, University of York, YO10 5DD York, U.K.

Corresponding author: Qiao Wang (qiao.wang@york.ac.uk)

ABSTRACT Proactively caching content at the network edge is particularly effective in high-mobility vehicular networks, where intermittent connection is the major challenge for seamless content transmission. The objective of this paper is to achieve proactive caching in vehicular networks by mobility prediction, specifically by predicting the next roadside unit (RSU) for a vehicle with reinforcement learning techniques. The paper proposes two proactive caching algorithms based on *multi-armed bandit* (MAB) learning, *non-contextual MAB* and *contextual MAB*, respectively. This paper fills the void in the literature regarding the application of MAB learning to mobility-prediction based proactive caching. Their feasibility, superiority, and applicability are evaluated with simulation in two modern cities: Las Vegas, USA with a grid road layout, and Manchester, UK with a more historical layout. Additionally, this paper is the first that proposes to investigate the uncertainty associated with proactive caching systems in the form of entropy with a specifically extended *Subjective Logic* framework, in order to provide an insight into the underlying link between prediction accuracy and uncertainty.

INDEX TERMS Proactive edge caching, reinforcement learning, multi-armed bandit, mobility prediction, vehicular networks, uncertainty analysis, entropy, subjective logic.

I. INTRODUCTION

Past decades have witnessed a rapid growth of the automobile industry and its economic and societal impacts continue to expand. With the rapid development in electronics and communications, vehicles will be able to communicate with each other, forming a large communication network, i.e., vehicular networks [1]. In addition, the upcoming era of autonomous vehicles means that vehicles will soon not only act as a simple means of transportation but also become moving entertainment centers where passengers are able to entertain themselves while traveling in the car [2], transferring the vehicular network to content-centric, e.g., streaming videos. On the other hand, different from general mobile networks, a peculiarity of vehicular networks is high-speed mobility. As fast-moving objects, vehicular users consequently experience short intermittent connectivity with roadside units (RSUs) more frequently than ordinary mobile users [3], [4]. Such frequent link re-connections due to vehicles' high-mobility means that vehicles may not be able to finish consuming the content before leaving the connected RSU, meaning that they have to re-establish the connection to the remote server for the remaining parts at a drastically reduced data rate [4], [5]. This is the main factor that causes intermittent content transmissions and a degraded quality of experience (QoE) [6].

To address the above challenge, proactive edge caching has been a promising technique to achieve seamless content transmission. This technique allows the vehicular network to pre-cache the (unfinished) content at the proper places in the network so that vehicular users can reduce the frequency of accessing content from content providers located in the core network. Therefore, a higher data rate can be achieved. In order to anticipate where to pre-cache the content, mobility-prediction techniques can be applied. Such techniques generally require computing resources to perform machine learning in the network. Recently, thanks to the development of mobile edge intelligence, mobile edge

The associate editor coordinating the review of this manuscript and approving it for publication was Yiqi Liu.

computing (MEC) units can be installed on the network edge i.e., RSUs, which enables them to perform both storage and computation functionalities [7], [8], hence the key enabler of mobility-prediction based proactive caching. Instead of predicting the exact position of vehicles, predicting the network edge node (i.e., RSU) and pre-caching the desired content at the node in advance allows vehicles to obtain immediate satisfaction after entering a new coverage area.

Effective proactive caching at the targeted RSU relies on effective prediction. For prediction purposes, the rapid development of machine learning (ML) and deep learning (DL) has played an important role. However, conventional ML and DL models require a great amount of training data and a training phase is normally indispensable. Thus, these models cannot well adapt to time-varying vehicular environments. Therefore, an online learning approach is needed. In fact, predicting the next RSU as a proactive caching node is a direct application of reinforcement learning (RL) because every prediction is a decision to make. The key feature of RL is that it does not require any prior knowledge of the environment, which makes it a suitable solution to online learning and enables the potential to achieve high adaptability in intelligent vehicular networks.

Nevertheless, in systems that do not have to be represented by states, the learning problems become stateless decision problems and the learning agent becomes stateless, which significantly reduces the number of trials needed to learn a mature strategy and speed up the learning process [9]. This is of great help in a dynamically changing vehicular environment. *Multi-armed bandit* (MAB) problem [10], [11] are basic instances of RL problems or to be specific, single-state model-free RL problems, where a learning agent does not have to build up a model of the environment. This feature makes it efficient in dealing with the variable vehicular environment. It has also attracted significant attention in various applications, from recommendation systems and information retrieval to healthcare and finance, thanks to its excellent performance combined with certain attractive properties, such as learning from fewer feedback [12]. In a bandit problem, the agent, i.e., the bandit, takes an action to achieve an immediate reward without states being involved, aiming to maximize the total amount of rewards.

Uncertainty is inextricably linked to learning algorithms and their models and is an important concept in machine learning methodology [13]. Assessing and quantifying uncertainty helps understand more precisely the benefits that models can bring. Reducing uncertainty will inevitably give us more accurate prediction results. *Subjective logic* [14], [15] has emerged as an effective method for uncertainty evaluation. This formalism allows us to express specific forms of probability distributions by generating a *multinomial opinion* over a discrete set of elements. It provides a concise formalism to represent Dirichlet-multinomial and Dirichlet-categorical models [16] and therefore, the opinion induces a categorical distribution over the element set that allows evaluation of the overall uncertainty as the entropy of the distribution. This model has also been recently used to assess uncertainties in deep networks [17], [18]. This will be further developed in this paper.

The purpose of the paper is to achieve effective mobility-prediction based proactive caching in vehicular networks. The approach to mobility prediction is by predicting the next RSU that a vehicle is about to access through MAB learning. We treat this as a decision-making process and investigate the feasibility and prediction performance of bandit learning. We designed two original prediction algorithms for proactive caching systems: *non-contextual MAB* and *contextual MAB*. The motivation for exploring two MAB-based algorithms is to further investigate the benefits of introducing *context* in contextual MAB on the prediction performance. Another purpose of the work is to investigate the uncertainty behind the proposed proactive systems with the Subjective Logic framework. The motivation behind this is that uncertainty is inseparably connected to learning algorithms, and we aim to verify and support the superiority of the proposed systems from the theoretical viewpoint of uncertainty. Our work fills the void of using MAB learning to solve proactive caching problems in such scenarios. Specifically, the main contributions of the paper can be summarized as follows:

- We design both non-contextual MAB-based and contextual MAB-based algorithms to address proactive caching at the next RSU. Despite the many applications of MAB in a range of fields such as ad placement and packet routing, we show how it can be used, for the first time, in pre-caching problems.
- We implement the proposed algorithms with online learning in a distributed way on individual RSUs, to realize instant learning and prediction, whilst previous similar work in [4] and [19] was based on centralized and offline approaches. Besides, the performance comparison with the baseline systems shows the advantages of using MAB learning in solving the proactive caching problem. Particularly, the contextual MAB with only a single context shows a faster convergence and better accuracy than the conventional sequence prediction model applied in [20].
- We extend the subjective logic framework specifically to proactive caching systems to analyze, using entropy, the overall uncertainty behind the bandit learning based systems as well as two baseline systems. By doing this, we aim to investigate the uncertainty variation and its connection with the prediction accuracy of different proactive caching systems.
- We experiment with the test data of two cities with significantly differing characteristics, Las Vegas and Manchester, from USA and UK respectively. The results show the scalability and adaptability of MAB-based approaches in proactive caching problems with different road layouts.

The rest of the paper is structured as follows. The related works regarding proactive caching and MAB applications in vehicular networks are summarized in Section II. Section III mainly discusses the network architecture and system model. The proposed algorithms are introduced in Section IV, and in Section V, the uncertainty analysis model is provided. Section VI shows the simulation results. Section VII discusses the theoretical analysis, time complexity, and convergence of the proposed algorithms. Section VIII concludes the paper.

II. RELATED WORK

This section discusses some relevant studies and is divided into two parts: *Proactive Caching in Vehicular Networks* and *Reinforcement Learning and Uncertainty*.

A. PROACTIVE EDGE CACHING IN VEHICULAR NETWORKS

Proactive edge caching in the literature can roughly be categorized as what to cache and where to cache. The former mainly relies on content popularity prediction. For example, the authors in [21] proposed a two-level prediction model for predicting video popularity to pre-cache popular videos in the content delivery network, and in the survey [22], the authors summarized the studies on popularity-based video caching techniques in cache-enabled networks. However, most popularity prediction methods require RSUs to collect vehicles' data which may contain sensitive information. We believe that this will become increasingly difficult for network operators given the increasing restrictions on security and privacy. In addition, they are not very effective because vehicles are fast-moving objects and this causes validity issues of the prediction. The latter, on the other hand, depends on how well the system is able to anticipate where a vehicle is going. This is more manageable and applicable for network operators and also essential in the rapidly changing vehicular environment. Therefore, this paper is interested in where to cache.

Predicting where to cache at the network edge via mobility prediction may include caching at vehicle nodes and RSU nodes. Past work in [23] and [24] focused on vehicle node caching, but they both require numerous data for offline training and the vehicles needed to send their trajectories to every RSU they visit in [24], which inevitably raised concerns on transmission overhead and privacy issues. Therefore, this paper concentrates on RSU node edge caching. Khelifi et al. [19] put forward a proactive caching scheme based on vehicular mobility prediction on top of a Named Data Networking architecture. The authors used Long Short Time Memory (LSTM) to first predict the moving direction of the vehicle and then estimated the next possible RSU, instead of directly predicting the next RSU. Similarly, the work in [4] also used LSTM to predict the direction of a vehicle by training realistic traffic data, after which Q-learning is applied to determine how much content to cache. The work in [25] applied the Markov chain model to predict the next RSU. The authors of [20] proposed a sequence-prediction based proactive caching system to address the problem. Their model is based on a sequence prediction algorithm, Compact Prediction Tree+ [26], by training vehicle-specified simulated traffic traces. Nevertheless, all these previous studies require an offline training phase with massive labeled data for a proper model. This is the first fundamental difference from this work where we concentrate on online learning, which can improve the adaptability of learning models in a time-varying environment. Besides, the prediction models in these studies are considered in a centralized way, that is prediction is made by a central node for a vehicle after the offline training stage. In contrast, this paper has considered a distributed system where RSUs learn and make predictions independently, which is the second substantial difference.

B. REINFORCEMENT LEARNING AND UNCERTAINTY

One of the most widely used model-free RL techniques is Q-Learning proposed by Watkins [27]. It is an off-policy method where the policy is updated based on the best possible future scenario, in contrast to its on-policy counterpart State–action–reward–state–action (SARSA) [28] that takes into account what actually happens after an action is taken for policy updating. However, given a highly dynamic vehicular environment and discrete action set in our problem, it is applicable and practical to consider it as stateless instead of its classical counterpart, which can dramatically reduce the number of Q-values needed for estimation by the learning agent, thereby potentially reducing the number of trials needed for it to learn a mature strategy and improving the adaptability of RL-based cognitive devices (e.g., RSUs). The MAB model is a single-state model [11] with no state transitions (i.e., stateless). While it has been widely used and proven to be effective in areas such as ad placement, computer game-playing, etc., its application in vehicular networks seems to be more limited. Dai et al. [29] proposed a multi-armed bandit learning algorithm called Utility-table based Learning to solve the distributed task assignment problem in a MEC-empowered vehicular network. The work in [30] focused on task caching problems in the edge cloud. The authors proposed an intelligent task caching algorithm based on a multi-armed bandit algorithm and evaluated its benefits in task latency performance. Authors of [31] discussed the potential of using a MAB problem in future 5G small-cell networks as well as its applications and future research directions. A detailed example of using a MAB model for energy-efficient small cell activation in 5G networks has been provided in [31]. Xu et al. [32] investigated collaborative caching problems in small-cell networks by learning the cache strategies directly at small base stations online by utilizing multi-agent MAB.

The uncertainty associated with learning models has attracted significant attention. *Subjective logic* first proposed in [14] has emerged as an effective method for uncertainty evaluation. The work in [16] used a subjective logic framework to solve bandit problems, where the action selection is based on sampling the multinomial opinion over the action set. They quantified the overall uncertainty of the proposed system with the entropy of the categorical distribution. The authors in [33] argued that Beta distribution and subjective logic are isomorphic in terms of fusion while finding the equivalence between uncertainty and entropy of Beta models. It has also been used for assessing uncertainty in deep networks as studied in [17] and [18].

Despite the benefit of proactive caching and the potential of the MAB, to the best of our knowledge, there are no studies focused on the problem of applying MAB to predict the next RSU for proactive caching. In addition, no one has studied the uncertainty of these systems so far. We believe this area is worth more investigation.

III. NETWORK ARCHITECTURE AND PROBLEM STATEMENT

We consider a MEC-enabled vehicular network where RSUs at the network edge are capable of caching and computing, as shown in Fig. 1. Vehicles in the network frequently request and download the content they are interested in. The intelligent RSUs are able to learn and predict the next possible RSU a vehicle will connect to and send a pre-caching request to that RSU. In this way, seamless content transmission can be achieved and user experience can be improved.

Consider an area G in an urban area deployed with N RSUs in a set $\mathcal{R} = \{r_1, r_2, \ldots, r_N\}$. There are residential areas and workplaces in G where vehicular users from the set $\mathcal{V} = \{v_1, v_2, \ldots, v_M\}$ travel to and from on a daily basis. A content database (content provider), $\mathcal{C} = \{c_1, c_2, \ldots, c_K\}$, exists in the backhaul network providing K types of content with various sizes for vehicles to request. The content is comprised of f_{c_k} ($c_k \in \mathcal{C}$) fragments and each fragment is a constant size F_c. Each RSU $r_i \in \mathcal{R}$ is MEC enabled so that they are capable of making mobility predictions and proactively caching content. There may be m neighboring RSUs of r_i and hence the next potential pre-caching node is selected from the neighbors. Besides, there is a central server that is responsible for sending the outcome of the earlier proactive caching decision to RSUs so that they can refine their learning models.

The communication model characterizes only the key elements needed to study the problem, given that the interest of the work is to predict where to cache accurately. Simply, a vehicle $v_j \in \mathcal{V}$ in the network always accesses the closest RSU, because the RSU access criteria do not influence the MAB learning algorithms. v_j may request a type of content c_k from the connected RSU r_i in a random way. r_i then starts to transmit c_k to v_j from its cache directly or through the content provider in the backhaul network or both, depending on the dwelling time and data rate. In order to fully focus on the mobility prediction task for proactive caching, the following assumptions are made: 1) the underlying issues arising at the physical and MAC layers e.g., packet loss, interference, and re-transmissions are not considered in vehicular communications so, the transmission rate e is a constant; 2) the dwell time of the vehicles in the coverage area of an RSU is extracted from the test trace being simulated and is known so that the number of content fragments can be derived; 3) the system is completely proactive, meaning that reactive caching is not enabled; 4) a vehicle does not request new content until it finishes consuming the current one and the system keeps a record of content consumption so that when handover occurs, vehicles continue to download the remaining of its previously requesting content; 5) despite the architecture of MEC, the computing and caching resource is assumed to be unlimited. These assumptions are legitimate because they do not influence the prediction performance of the proposed algorithms, which is the primary focus of this paper. Moreover, the abstraction of fragmented content is also a valid application in reality when delivering large data files or streaming content (e.g., videos) over many sources such as HTTP-based streaming such as Dynamic Adaptive Streaming over HTTP (DASH) [25], [34].

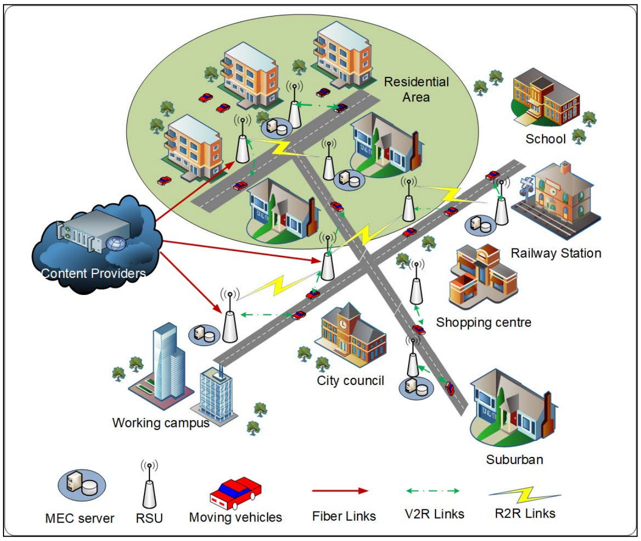


FIGURE 1. Architecture of MEC-enabled vehicular network.

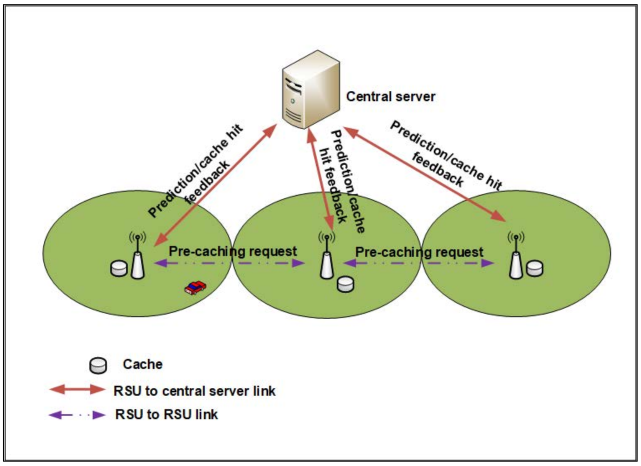

Central server
Prediction/cache hit feedback
Prediction/cache hit feedback
Prediction/cache hit feedback
Pre-caching request
Pre-caching request
Cache
RSU to central server link
RSU to RSU link

FIGURE 2. Distributed structure of proactive caching system.

A typical proactive caching process is shown in a diagram in Fig. 2. The current associated RSU may face a prediction decision involving a few neighboring RSUs when a vehicle requests content from it. Assuming the vehicle $\hat{v}$ sends its request for a content $\hat{c}$ right after it enters the coverage of the RSU $\hat{r}$, $\hat{r}$ serves $\hat{v}$ with its cached fragments of $\hat{c}$

TABLE 1. List of abbrevations.

CPT	Compact Prediction Tree
CPT+	Compact Prediction Tree+
CP	Content Provider
cMAB	Contextual Multi-armed Bandit
DL	Deep Learning
DASH	Dynamic Adaptive Streaming over HTTP
EQ	Equal-probability based proactive caching system
HTTP	Hypertext Transfer Protocol
LSTM	Long Short-Term Memory
MAB	Multi-armed Bandit
MAC	Medium Access Control
MEC	Mobile Edge Computing
ML	Machine Learning
PB	Probability-based proactive caching system
RL	Reinforcement Learning
SL	Subjective Logic

if available or through a content provider in the backhaul network otherwise, which would cause some delay or both ways. Whichever way, $\hat{r}$ evaluates how many fragments of $\hat{c}$ it can transmit to $\hat{v}$ before handover. If $\hat{c}$ has a relatively small size and/or $\hat{v}$ would stay connected rather long, $\hat{c}$ can be fully transmitted (consumed) and therefore, no proactive caching in the next RSU is required. Otherwise, $\hat{r}$ will predict the next RSU and request it to perform proactive caching on the remaining number of fragments F_R of c_K from fragment No. f_r, by sending the *proactive caching request* message. A transmission delay μ would be introduced if f_r is not found in the actual next RSU, via $\frac{F_R \times F_c}{\omega}$ where ω is the backhaul link rate. The role of the central server that connects multiple RSUs is to transmit *prediction/cache hit feedback* message which acts as *rewards* in the learning algorithms.

Problem Statement: In vehicular networks with proactive caching enabled, the goal of this feature is to realize seamless content transmission, which addresses the challenge caused by intermittent connectivity. If the next possible RSU that a vehicle is about to access can be accurately predicted, a higher cache hit ratio can be achieved. Therefore, how to predict the next RSU node for a connecting vehicle as accurately as possible through multi-armed bandit learning is the primary problem studied in this paper. Additionally, how the uncertainty of prediction associated with proactive systems evolves during learning is another problem to be investigated.

IV. ALGORITHM DESIGN

This section will briefly introduce some background of *multi-armed bandit problem*, following which the learning algorithms designed for proactive caching will be discussed.

A. MULTI-ARMED BANDIT PROBLEM

Multi-armed bandit (MAB) problem, sometimes also known as k-armed bandit problem, is a special instance of reinforcement learning (RL). Different from a full RL problem where a learning agent may have multiple states associated with the environment (e.g., positions in a game), it only has a single state in the MAB problem [11] (i.e., no state transitions). From this perspective, MAB is essentially identical to *stateless Q-Learning* [35] and can also be treated as a model-free reinforcement learning technique. A well-known scenario of the bandit problem is where a gambler in a casino sits in front of a slot machine with one or multiple arms (referred to as a one-arm bandit and k-armed bandit respectively) and tries to get payoffs by pulling the arm(s). The ultimate goal of the gambler is to achieve the highest cumulative rewards through learning the inherent reward pattern of each lever and gradually concentrating on the best lever. During the learning process, the gambler will face the *exploration-exploitation* dilemma [36]: where the gambler tries out the potential arms that may return high payoffs (exploration) or pulls the arm that has yielded the highest reward from the past experiments (exploitation). This is a non-trivial process and carefully balancing exploration and exploitation is crucial in MAB problems.

A MAB problem can be formally given as a tuple [16]: $\langle \mathcal{A}, \mathcal{R} \rangle$, where $\mathcal{A} = \{a_1, a_2, \ldots, a_k\}$ is the a set of k actions (i.e., arms) and $\mathcal{R} = \{\theta_1, \theta_2, \ldots, \theta_k\}$ associates action a_i with its reward probability distribution defined by θ_i. There are a number of variants of MAB problems and it is out of the scope of the paper to cover all of them. Therefore, in the following a canonical example of MAB - the Bernoulli bandit problem and the contextual bandit will be discussed as they are closely related to the problem here and the proposed learning algorithm for proactive caching. In addition, the approaches to resolve the exploration-exploitation dilemma in MAB problems are plenty such as ϵ-greedy, upper-confidence bound algorithm, Thompson sampling [36], etc. The aim of this paper is not to find out a sophisticated way to balance exploration and exploitation so the most straightforward ϵ-greedy is adopted here.

1) BERNOULLI MULTI-ARMED BANDIT

Consider a k-armed bandit problem $\langle \mathcal{A}, \mathcal{R} \rangle$. The agent takes actions from action set $\mathcal{A}$ and any action played will generate an outcome: success (reward 1) or failure (reward 0). Action $a \in \mathcal{A}$ produces a success with probability $\theta \in \mathcal{R}$. In other words, for an action a, a reward $r = 1$ is produced with probability θ and $r = 0$ with probability $1 - \theta$. In this case, θ can be viewed as the expected reward of taking action a, is unknown to the agent, and is invariant in a *stationary* MAB problem. One way to estimate such θ is to use *sample-average* method [11] by averaging the rewards actually received. The estimate of θ for action a at timestep t can be denoted as:

$$Q_t(a) = \frac{\text{sum of rewards when } a \text{ taken prior to t}}{\text{number of times } a \text{ taken prior to t}} = \frac{\sum_{i=1}^{t-1} r_i \cdot \mathbb{1}_{A_i=a}}{\sum_{i=1}^{t-1} \mathbb{1}_{A_i=a}} \quad (1)$$

where A_i is the action taken at timestep i, $\mathbb{1}_{condition}$ equals to 1 if *condition* is true and 0 if not, and $r_i = \{1, 0\}$ is the reward of i-th selection of action a. According to the law of large numbers, Equation (1) converges to θ as the

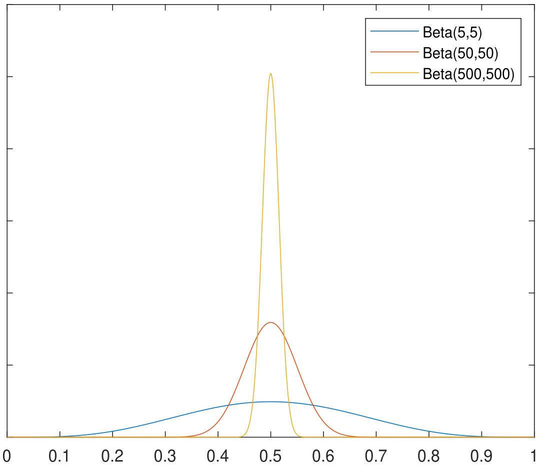


FIGURE 3. Example of the sample-average process shown with Beta distribution.

denominator tends to infinity. A more intuitive way to illustrate this is through the probability density function of $Beta(\alpha = successes, \beta = failures)$ distribution as shown in Fig. 3. Consider a 95% confidence interval, in the late stage of the sample-average process after 1000 trials with 500 successes and 500 failures, the range that captures the true probability θ is [0.469, 0.531], i.e., $P(0.469 < \theta < 0.531) = 0.95$. However, the intermediate stage with 100 trials (50 successes and 50 failures) returns a much wider range of [0.403, 0.597] for the same 95% confidence interval, and the initial stage with only 10 trials gives an even wider range of [0.212, 0.788]. Thus, the more trials, the more certain one can be about the approximation to the true probability θ. By taking the proper action with the associated action-selection strategy (e.g., ϵ-greedy), it is also to maximize the cumulative rewards $\sum_{t=0}^{T} r^t$ where T is the given time horizon.

2) CONTEXTUAL BANDITS

As an extension of the above multi-armed problem, the contextual bandit problem associates actions with side information or context [37]. In such problems, the agent aims to learn a policy that maps contexts to actions, that is, $\pi(a_i \mid s_j)$ where s_j is one of the contexts. Another viewpoint is that it now consists of multiple independent MAB tasks associated with contexts, and the agent aims to learn the best policy under various contexts. Every time an agent is assigned a MAB task (possibly with a certain probability), it will be given a ''clue'' (i.e., context) and learn what the best action is under this clue. In general, the agent can do better with the presence of context information that distinguishes one bandit problem from another [11]. Despite the fact that contextual bandit problems involve learning policies, they still resemble the general MAB tasks, as the action taken only affects the immediate reward, and makes no difference to the future situations as well as their rewards. Therefore, it is an intermediate between the MAB problem and the full RL problem.

B. MAB-BASED PROACTIVE CACHING ALGORITHMS

The primary focus of the proactive caching problem in this paper is where to pre-store relevant content in the immediate future. Therefore, it is vital for an RSU to predict as accurately as possible the next potential RSU a vehicle is about to hand over to. Intuitively, this may not seem to be closely related to a MAB problem so in the following we will first demonstrate how to match them together.

1) MAPPING OF PROACTIVE CACHING TO MAB PROBLEMS

As discussed in the last subsection, the MAB problem mainly consists of agents, actions, rewards, and contexts as in a contextual bandit problem. An agent aims to maximize its cumulative rewards by taking appropriate actions from the action set in a given period of time. Regarding the next-RSU prediction based proactive caching in vehicular networks, an RSU assists a vehicle to successfully hit the content that was previously being transmitted. They resemble each other in terms of node selection and success or failure (reward). Therefore, we model this problem as a MAB problem using the following mappings:

- **RSUs as bandit learning agents**: Any RSU in the vehicular network acts as a learning agent, and its neighboring RSUs are equivalent to its actions. Predicting the next RSU as a proactive caching node is actually making a decision on one of the agent RSU's neighbors.
- **Stateless RSU**: In general, the state of a reinforcement learning agent is associated with the environment. Since the interaction of an RSU with the vehicular environment can be extremely dynamic and complicated to represent, the single-state feature of MAB resolves this problem. In other words, an agent RSU is single-state or stateless which means that it does not transfer to a new state by taking an action.
- **Action selection as next RSU prediction**: The agent RSU will either exploit its current knowledge to select the *greedy* action/neighbor or explore other non-greedy actions that may return a higher reward depending on the exploration-exploitation scheme adopted.
- **Reward generation**: When handover happens, the system will return a reward to the previous agent RSU. This is achieved by determining whether there is pre-cached content in the RSU after the handover, or alternatively whether the RSU is the previously predicted one. The reward in return helps an agent RSU compute the estimated values of its actions.
- **Previous RSU as context**: The agent RSU may also make use of contexts for its action selection as in a *contextual bandit problem*. By identifying the previous RSUs that the visiting vehicles coming from as contexts, it can map such contexts into various bandit tasks and perform more effective learning. Technical details about the contextual information will be discussed shortly.

In a vehicular network with multiple RSUs, the problem becomes a *Multi-agent Multi-armed bandit* (MAMAB) problem where each individual RSU is an independent player and learns its own best action or best policy. On this basis, we designed two algorithms to address mobility-prediction based proactive caching: ***non-contextual MAB*** and ***contextual MAB***.

2) ADDRESS PROACTIVE CACHING WITH BANDIT LEARNING

We will elaborate on the two bandit learning algorithms that address proactive caching from three aspects: *action selection and value estimation*, *reward function*, and *context information*.

a) **Action selection and value estimation** Two critical elements in MAB problems are action selection and action value update. Given the estimated action values $Q(a)$ of actions in action set $\mathcal{A}$, the ϵ-greedy method is used to make a selection: the best action is selected with probability $1-\epsilon$; otherwise, actions will be selected randomly with a small probability ϵ regardless of their action values, as demonstrated in Equation (2) where A_t is the predicted action at timestep t.

$$A_t = \begin{cases} \arg\max_a Q(a), & 1-\epsilon \\ random, & \epsilon \end{cases} \tag{2}$$

Another important method is the action value estimation, also known as *action-value method* in the literature. Recall in a Bernoulli multi-armed bandit, the true success probability θ of action a is its expected reward, defined as $\theta \doteq E[r \mid A = a]$. The sample-average approximation method for action-value estimation shown in Equation (1) can have a more compact representation with incremental implementation [11]. For action a which has been selected for n times, the estimated value is:

$$\begin{aligned} Q_{n+1} &= \frac{1}{n} \cdot \sum_{i=1}^{n} r_i \\ &= \frac{1}{n}\left(r_n + (n-1)\frac{1}{n-1}\sum_{i=1}^{n-1} r_i\right) \\ &= \frac{1}{n}\left(r_n + (n-1)Q_n\right) \\ &= Q_n + \frac{1}{n}\left(r_n - Q_n\right) \end{aligned} \tag{3}$$

An important parameter in the incremental value updating rule of Equation (3) is $\frac{1}{n}$, the *step-size*. As can be noted from Equation 3, this step-size declines as n grows. In fact, this is effective in a ***stationary*** bandit problem where the reward probabilities (i.e., θ) remain unchanged over time. Vehicular networks, however, are dynamic environments with varying traffic densities and may result in a ***non-stationary*** bandit problem. Therefore, recent rewards should be given more weight when updating action values. This is often achieved using a constant step-size denoted with $\alpha \in [0, 1]$ and Equation (3) therefore becomes:

$$Q_{n+1} = Q_n + \alpha(r_n - Q_n) \tag{4}$$

A more general form of Equation (4) that is adopted in our algorithms is:

$$Q(a) \leftarrow (1-\alpha)Q(a) + \alpha r \tag{5}$$

where $Q(a)$ is the quality value of action a, named Q-value as in Q-learning [9], [35], r is the reward associated with the most recent trial and is determined by a reward function, and $\alpha \in [0, 1]$ is the step-size or *learning rate* .

b) **Reward function** The reward function R is used to generate a reward associated with the action taken previously when an outcome is observed. Given an action a taken at time step t and the observed outcome as b (may occur immediately), its reward can be computed with $r_t = R(b)$. In a Bernoulli MAB problem discussed earlier, the reward function R is actually the outcome itself (1 or 0), meaning that $r_t = R(b) = b$. In order to introduce punishment to wrong predictions or cache misses, we referred to the reward function that has been successfully applied in the Dynamic Spectrum Access problem in [9]:

$$R(b) = \begin{cases} 1, & b = \text{True} \\ -1, & b = \text{False} \end{cases} \tag{6}$$

As mentioned earlier, the outcome b is determined by observing whether a vehicle switches to the predicted RSU, equivalent to a cache hit or miss if a pre-caching request was sent to the RSU. The relevant reward will then be generated with Equation (6) and fed back to the earlier decision-making RSU. With Equation (3), (5) and (6), the learning agent aims to update its estimate of each action $Q(a) = E[r_t]$, make an action selection and maximize its cumulative rewards $max \sum r_t$. Notably, due to the constant α adopted in Equation (5) and the negative reward introduced in Equation (6), $Q(a) \in [-1, 1]$ is no longer a probability i.e., it is not an estimate of θ as in the sample-average method (Equation (1)), but directly represents the expected reward of the action a.

c) **Contextual information** The above methods for updating actions' Q-values, selection, and reward function can be applied to both non-contextual and contextual bandit problems. However, the agent in the general non-contextual MAB learning could face the dilemma where two or more of its actions may converge to very close estimated Q-values, which poses great uncertainty when predicting an accurate next RSU node. Therefore, the motivation for proposing a contextual MAB algorithm is to resolve this as best as possible. The agent in a contextual MAB problem maps *contexts* to its action set and associates a specific Q-table with each individual context and aims to learn a policy under different them. In the vehicular network, vehicles may come from various directions which can be useful contextual information. If the agent RSU can make use of it and split it to separate bandit tasks, it is likely to improve the overall accumulated rewards.

Specifically, the context we introduced on top of a non-contextual MAB-based algorithm is the previous RSU that a vehicle connected to before the current agent RSU. The rationale behind this is that the previous RSU is a very easily accessible context and this does not require additional effort on signaling extra information, compared to other types of context e.g., road information, vehicle angel, etc. Once a vehicle connects to an RSU and starts to request content from it, the agent RSU needs to predict the next RSU (action selection) and inform it to pre-cache the needed content if necessary. In contextual MAB, the agent RSU now needs to first identify the previous RSU as context and learn the action values associated with it so that decisions are properly made under a particular context. The equivalent equations to Equations (2) and (5) for action selection and Qvalue updating in contextual MAB become:

$$A_t = \begin{cases} \arg\max_a Q_t(a \mid s_j), & 1-\epsilon \\ random, & \epsilon \end{cases} \tag{7}$$

$$Q(a \mid s_j) \leftarrow (1-\alpha)Q(a \mid s_j) + \alpha r \tag{8}$$

where s_j is the detected context at t.

Algorithm 1 Non-Contextual Multi-Armed Bandit

```
Initialization (if not done): For RSU $m \in \mathcal{M}$ with the number of actions (RSU neighbors) $\mathcal{A}_m$, their Q-values are initialized to $Q(a) = 0$ for $a \in \mathcal{A}_m$
while not the end of the test do
    if Content transmission is happening whilst in RSU $m$ then
        Predict the next RSU by:
        $a^* \leftarrow$ selection decision based on Eq. (3)  Precaching content at $a^*$ if needed;
    end if
    if Handover happens then
        $r^* \leftarrow$ observe the reward according to Eq. (6)
        Update $Q(a^*)$ with Eq. (5);
    end if
end while
```

We sum up the above in Algorithm 1 and Algorithm 2 for non-contextual and contextual bandit learning respectively, which have been applied to our proactive caching problem. Additionally, a general flowchart of MAB-based proactive caching is integrated and shown in Fig. 4, though contextual MAB may also involve identifying the context and updating its action values correspondingly.

V. UNCERTAINTY ANALYSIS MODEL

In decision-making problems, reducing uncertainty is deemed to be vital as less uncertainty means that an agent is likely to make more accurate decisions. Thus, it is meaningful to assess and quantify the uncertainty in a learning problem.

Algorithm 2 Contextual Multi-Armed Bandit

```
Initialization (if not done): For RSU $m \in \mathcal{M}$ with the number of actions (RSU neighbors) $\mathcal{A}_m$, their Q-values are initialized to $Q(a) = 0$ for $a \in \mathcal{A}_m$
while not the end of the test do
    if Content transmission is happening whilst in RSU $m$ then
        $s \leftarrow$ detect the previous RSU $s$ before $m$  if $s$ is a new detection then
            Create an entry of $s$ to its action values  Initialize $Q(a \mid s) = 0$ for $a \in \mathcal{A}_m$;
        end if
        Predict the next RSU by:
        $(a^* \mid s) \leftarrow$ selection decision based on Eq. (7)
        Precaching content at $a^*$ if needed;
    end if
    if Handover happens then
        $r^* \leftarrow$ observe the reward according to Eq. (6)
        Update $Q(a^* \mid s)$ with Eq. (8);
    end if
end while
```

In this work, we adopt *Subjective Logic* framework [15] and particularly adjust it to investigate uncertainty in bandit learning based proactive caching systems. The motivation behind this is to provide a more insightful analysis model for the performance of proactive caching systems and how uncertainty evolves during the learning process. We also aim to give a greater insight as to how MAB-based systems outperform the others and how the context introduced by the contextual MAB algorithm could benefit the whole system. This subsection will introduce some background and discuss how we achieve this.

A. UNCERTAINTY

In the field of machine learning and statistics, a reliable estimation of uncertainty plays an important role in order to create reliable statistical models [16]. In [13], uncertainty in statistical models is classified as *aleatoric* and *epistemic*. Given a set of observed data samples $\mathcal{D} = \{d_1, d_2, \ldots, d_n\}$ that are generated by an unknown stochastic process P, if the task is to fit a model $p(\mathcal{D} \mid \Theta)$ that describes the observation $\mathcal{D}$, the set of parameters Θ needs to be learned from the collected observations. Apparently, the uncertainty that affects the accuracy of model $p(\mathcal{D} \mid \Theta)$ comes from both the internal randomness of process P and the limitation of the number of observations used to estimate the model. Therefore, these two types of uncertainty can be described as:

- *Aleatoric uncertainty* is inherent randomness in the data generation process P which can be reflected by the variability in the outcome of a trial. A typical example is coin flipping. For this type of uncertainty, however much data is provided, the uncertainty of the final fitted model $p(\mathcal{D} \mid \Theta)$ is unlikely to be less than the underlying model P [16].

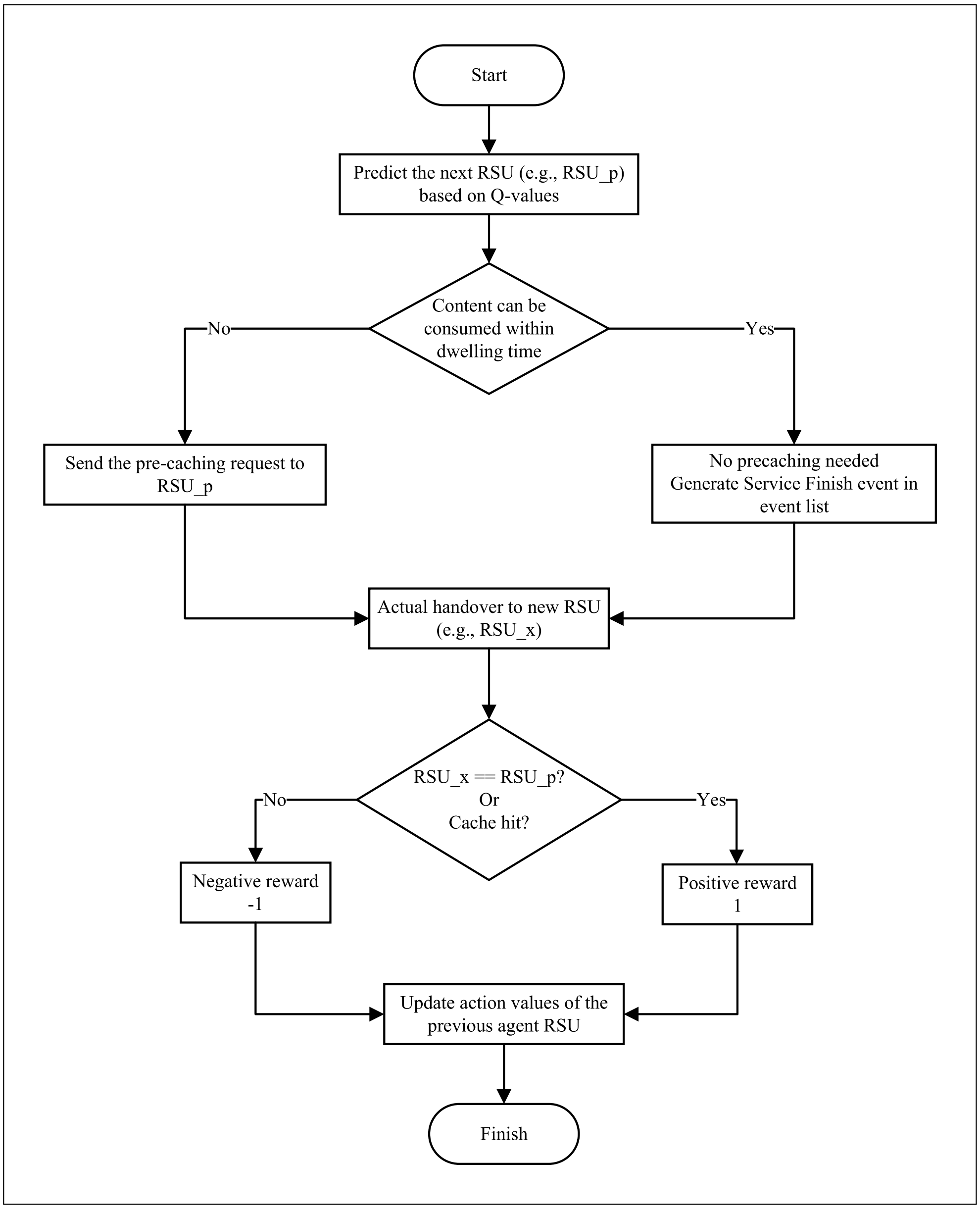

Start
Predict the next RSU (e.g., RSU_p) based on Q-values
Content can be consumed within dwelling time
No
Yes
Send the pre-caching request to RSU_p
No precaching needed Generate Service Finish event in event list
Actual handover to new RSU (e.g., RSU_x)
RSU_x == RSU_p? Or Cache hit?
No
Yes
Negative reward -1
Positive reward 1
Update action values of the previous agent RSU
Finish

FIGURE 4. Flowchart of MAB-based proactive caching algorithm. This is a general cycle of an agent RSU serving a connecting vehicle, from ***Start*** when it receives a content request from a connecting vehicle, to ***Finish*** when its action-value table is successfully updated with corresponding rewards.

- *Epistemic uncertainty* on the other hand, is due to the lack of knowledge about the best model such as finite sample size. Different from aleatoric uncertainty, epistemic uncertainty can be improved by having more samples or trials.

The present study concentrates on the overall uncertainty of bandit learning algorithms, accounting for both aleatoric and epistemic uncertainties, which can be computed as the entropy of the relevant distribution under the subjective logic framework.

B. SUBJECTIVE LOGIC

Subjective logic [15] has been a promising approach to evaluating uncertainties in a statistical model. It is a compact formalism to represent specific forms of probability distributions (Dirichlet-multinomial and Dirichlet-categorical models) [16]. Specifically, given a discrete domain $\mathcal{X} = \{x_1, x_2, \ldots, x_k\}$ with k elements, there exists an *multinomial opinion* for the domain:

$$o = (b, u, c)\,, \quad \text{subject to } u + \sum_{i=1}^{k} b = 1$$

- $b \in \mathbb{R}^k_{\geq 0}$: *belief vector* that represents the degree of certainty over the k elements
- $u \in \mathbb{R}_{\geq 0}$: *uncertainty scalar* that shows the degree of certainty on belief vector
- $c \in \mathbb{R}^k_{\geq 0}$: *base rate vector* which often expresses the prior probability distribution of the k elements

According to [16], the belief vector acquires the *first-order* uncertainty of the distribution of beliefs over the domain mapping to the aleatoric uncertainty whereas u maps to epistemic uncertainty capturing the *second-order* uncertainty about the belief model. In such a model, the probability of an element x_i in the domain $\mathcal{X}$ with opinion o can be computed with:

$$p\left(x_i \mid o\right) = b_i + uc_i \tag{9}$$

An existing mapping between an opinion $o = (b, u, c)$ and an evidential Dirichlet pdf $s = Dir_e(e)$ [15], [16]:

$$\begin{cases} e_i = \dfrac{Wb_i}{u} & \text{if } u \neq 0 \\ e_i = \infty & \text{otherwise} \end{cases} \tag{10}$$

whose reverse is:

$$\begin{cases} b_i = \dfrac{e_i}{W + \sum_{i=1}^{k} e_i} \\ u = \dfrac{W}{W + \sum_{i=1}^{k} e_i} \end{cases} \tag{11}$$

where W is a non-informative prior weight normally specified equal to 2 for consistency.

Equations (10) and (11) form a theoretical foundation for the uncertainty analysis in the present work. Most importantly, Equation (11) allows building the multinomial opinion o over actions of RSUs with experiment observations (i.e., evidence). Hence we are able to obtain the probabilities of actions and overall uncertainty in the form of entropy accordingly. How we define evidence and the overall uncertainty calculation will be discussed in the following.

C. UNCERTAINTY EVALUATION OF PROACTIVE CACHING SYSTEMS

Similar to the uncertainty in decision-making theory, two sources of uncertainty exist in proactive caching systems, corresponding to aleatoric and epistemic uncertainties. On the one hand, for an RSU, the right decision depends on the proactive caching scheme as well as the randomness in the system. These are all inherent aleatoric uncertainty. On the other hand, epistemic uncertainty in such systems comes from the lack of visits of the RSU or the lack of chances for it to make decisions, which should be reduced as more observations are collected.

To form an opinion over an action set, the evidence of the set needs to be collected, with which the corresponding belief vector and uncertainty scalar of the opinion tuple can be obtained through Equation (11). The probability of individual action can be achieved accordingly via Equation (9). For an arbitrary RSU with m actions, the subjective opinion $o^t = \left(b^t, u^t, c\right)$ at an arbitrary timestep t conveys:

- the belief of an agent on action a_i being the best action with b_i^t
- the global uncertainty over the beliefs with u
- the prior belief c which is constant

Therefore, at timestep 0 or at the beginning of the learning process, the initial values of the three elements are:

$$\begin{cases} b_i^0 = 0 & \forall i \in [1, m] \\ u^0 = 1 & \\ c_i = \dfrac{1}{m} & \forall i \in [1, m] \end{cases}$$

which means that the agent has no knowledge about which of its actions is likely to be the best and they have equal probabilities. The uncertainty at this point is the maximum, 1.

The rule we used to collect evidence that supports the belief that action a^t could be the best is straightforward:

$$e_i^{t+1} = e_i^t + \mathbb{1}\left[a^t = a_i\right]$$

where the evidence is updated by adding one piece if $\left[a^t = a_i\right]$ is true. Thus, the evidence at any timestep t forms the opinion o^t and with Equation (9) a categorical distribution of the action set can be induced: $p\left(a \mid o^t\right) = Cat(b^t + u^t c)$. From this distribution, the overall uncertainty can be calculated as the entropy of the distribution:

$$H = -\sum_{i=1}^{m} p(a_i \mid o^t) \log_2 p(a_i \mid o^t) \tag{12}$$

For the non-contextual MAB algorithm, Equation (12) can be applied directly because of its single-state feature. In contrast, for contextual MAB, the entropy computation needs to consider the number of contextual situations.

Given an agent that has n contextual situations denoted by $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$ with m actions, each of these situations is an independent bandit task as mentioned earlier. As a consequence, we can compute their entropy called *context entropy* as:

$$H(s_j) = -\sum_{i=1}^{m} p(a_i \mid o^t, s_j) \log_2 p(a_i \mid o^t, s_j) \tag{13}$$

For the agent, the global uncertainty in terms of entropy then becomes:

$$H = \frac{\sum_{j=1}^{n} H(s_j)}{n \log_2 m} \tag{14}$$

This draws on the *Exploration Entropy* in a full reinforcement learning problem [38] where multiple *states* are associated with an agent.

In the proactive caching system, the actions of an agent RSU have their own success probability, which is a source of the aleatoric uncertainty. As mentioned earlier, even the optimal model cannot have less uncertainty than the true process. The MAB-based algorithms cannot remove such intrinsic uncertainty but aim to form a belief vector b over the actions that best describe it. For non-contextual MAB, sufficient learning (trials) allows the agent RSU to have the best model for the aleatoric uncertainty, compared to other non-contextual baseline systems (which we shall see in the results section). In other words, enough evidence results in a small epistemic uncertainty u, and a smaller overall uncertainty means a better-fitted model. Contextual MAB (cMAB), on the other hand, introduces a context (i.e., previous RSU) to further disaggregate the problem into context-related ones. The aleatoric uncertainty under each context s may be substantially reduced in contrast to the non-contextual case. Therefore, after sufficient learning, the agent RSU will have the best model for the aleatoric uncertainty associated with each context s, thereby less overall uncertainty.

To sum up, Equation (12) will be applied to evaluate the overall uncertainty in the non-contextual MAB-based proactive caching algorithm, and Equation (13) and (14) will assess the contextual MAB-based algorithm.

VI. SIMULATION AND PERFORMANCE EVALUATION

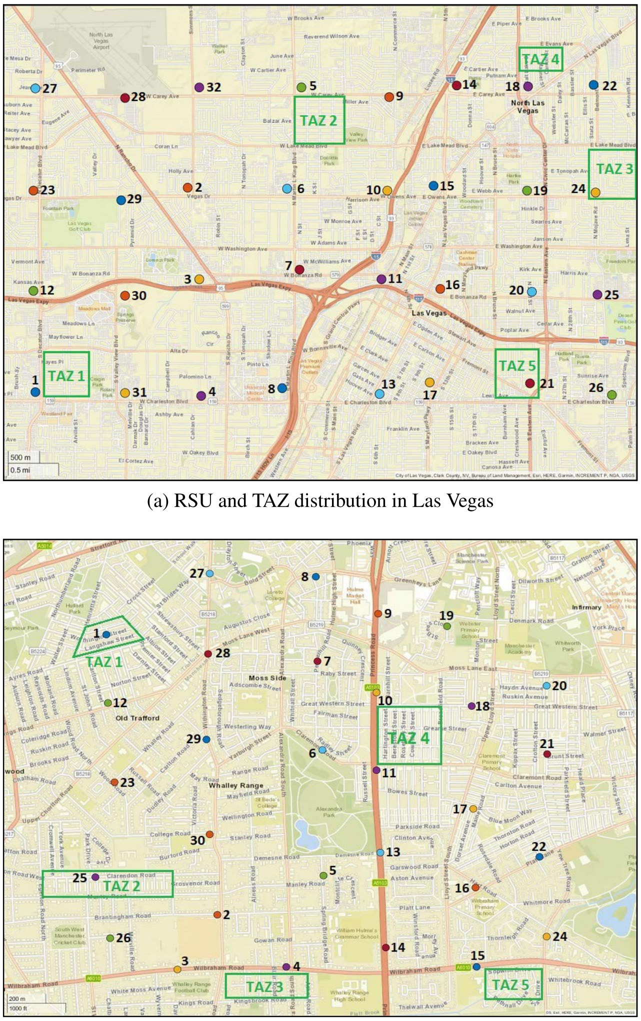
A. SIMULATION SETUP

Simulation in this work includes two parts: traffic simulation and network simulation. Vehicle traffic traces are generated by Simulation of Urban MObility (SUMO) [39] and they are processed with an event-driven network simulation program implemented in MATLAB [40].

1) TRAFFIC SIMULATION

SUMO is used to simulate a real transportation network discussed in Section III. The scenario we are interested in is the daily commuting routine of people living in a particular urban area. We focus on an area in Las Vegas as our primary city and Manchester as a secondary city to generalize the application of MAB-based schemes to two cities with different road planning. For both areas shown in Fig. 5, five traffic zones (TAZs) are defined in SUMO and in total 174 vehicles travel from and to these zones as their origins and destinations. The positions of these TAZs have been chosen in such a way as to allow vehicles to have the longest possible trips. These TAZs are designed to simulate realistic residential and office areas. We assume that a TAZ contains both residential and office areas. In order to simulate vehicles with the same daily routine, each vehicle has its own fixed departure and arrival zone. However, each vehicle may have different departure times and lanes (which may result in route differences) from test trace to test trace. Again, this is to imitate that people in reality may set off for work at various time slots, park at various places in an area, and take slightly different commuting routes, despite having the same workplace (TAZ).

200 files of test traces for each city have been generated to simulate 200 workdays and the simulation period in SUMO is between 8 am to 9 am. The vehicles' routes between two TAZs are defined by the tool *duarouter* and follow the Shortest or Optimal Path Routing rule. They depart at the *maxSpeed* and follow the default Car Following Model to keep the maximum speed which is safe in the sense of being able to stop in time to avoid a collision. Other road behaviors apply as well such as lane changing, accelerating/decelerating, intersections, etc. Technical details about these settings can be found in SUMO documentation.[1]



(a) RSU and TAZ distribution in Las Vegas

(b) RSU and TAZ distribution in Manchester

FIGURE 5. RSU and TAZ distribution in the two urban areas.

2) NETWORK SIMULATION

Discrete event-driven system simulation [41] allows vehicular network simulation to be performed through a series of events. Test traces are generated by SUMO and passed to the simulation system sequentially. The discrete event list corresponding to the test trace being tested is created at the beginning, which may include *departure* and *arrival* of vehicles, *content request*, *handover*, and *finishing of content consumption*. As the present work concentrates on online learning, a complete cycle of the simulation is testing 200 trace files and the learners (i.e., RSUs) make predictions as they learn throughout the simulation cycle and become increasingly knowledgeable as the simulation runs. Fig. 6 shows a structure of the modules mentioned and the relevant parameters are summarized in Table 2. The number and location of the RSUs were determined to cover as much of the area as possible while maintaining a distance of a few hundred meters between them. However, these factors do not affect the applicability and adaptability of the proposed algorithms,

[1] https://sumo.dlr.de/docs/

as will be discussed in Section VII on scalability issues. The closely related parameters to the MAB-based systems in Table 2 are learning rate α referenced in [42] and ϵ selected empirically. The network parameters such as transmission rate and backhaul link rate, are empirical values and they have no impact on the performance of proactive caching (i.e., prediction accuracy or cache hit ratio).

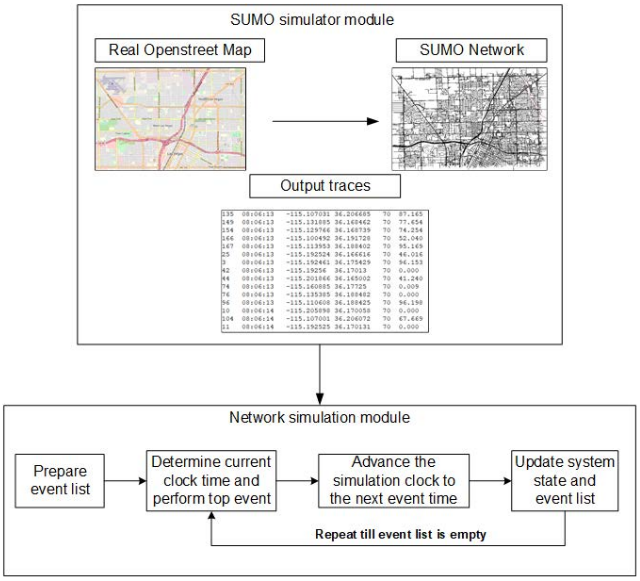

SUMO simulator module
Real Openstreet Map
SUMO Network
Output traces
Network simulation module
Prepare event list
Determine current clock time and perform top event
Advance the simulation clock to the next event time
Update system state and event list
Repeat till event list is empty

FIGURE 6. **Simulation modules.**

TABLE 2. **Simulation parameters.**

Parameter	Value
α for bandit learning	0.5
ϵ for bandit learning	0.05
No. of test traces	200
No. of Vehicles	174
SUMO Simulation Time	8:00 - 9:00
No. of RSUs	32 (Las Vegas) / 30 (Manchester)
Backhaul Link Rate	$\omega = 5Gbps$
Transmission rate	$e = 50Mbps$
Size of content database	$K = 30$
Fragment size	$F_c = 100MB$

B. PERFORMANCE EVALUATION

The performance of non-contextual and contextual MAB-based proactive caching systems is compared with three other proactive caching systems:

- *Equal Probability-based Proactive Caching System*: RSUs select a pre-caching node with equal weight from their neighbors. In other words, it is a random selection scheme.
- *Probability-based Proactive Caching System*: This allows RSUs to make the next pre-caching node decision based on their previous popularity using information from historical traces. This is an intuitive scheme where an RSU believes the neighbor with more frequent handovers deserves a higher weight to become the caching node.
- *CPT+ based Proactive Caching System*: This system is based on the sequence prediction algorithm CPT+. Different from the work [20], we have adjusted the algorithm to be used in an online mode. In brief, an RSU trains its prediction tree model with currently available vehicles' data and when predicting the next RSU for a vehicle, it matches all the past RSUs this vehicle has connected and gives out the most possible RSU (highest score).

Remark: the five systems are referred to and denoted in the following as *cMAB* and *MAB* for contextual and non-contextual bandit learning systems, respectively; *EQ*, *PB*, and *CPT+* represent for equal probability-based, probability-based, and CPT+ based systems, respectively.

1) EVALUATION

We mainly focus on the evaluation of the proactive caching performance of the systems. An action selection is considered correct when the predicted pre-caching RSU is the actual RSU to which the handover is made. In the systems considered, it is identical to a cache hit. Additionally, the extended subjective logic framework discussed in Section V is applied to the systems to provide an analysis of uncertainty except for CPT+. This is because CPT+ is a fundamentally different algorithm compared to the other four, in terms of its model and algorithm design. The variability of its action set and the difficulty of accessibility to "contexts" have made the extended uncertainty model inapplicable. The entropy calculation for EQ and PB systems is also based on Equation (12) as the non-contextual MAB. Furthermore, how proactive caching systems benefit the network is also considered.

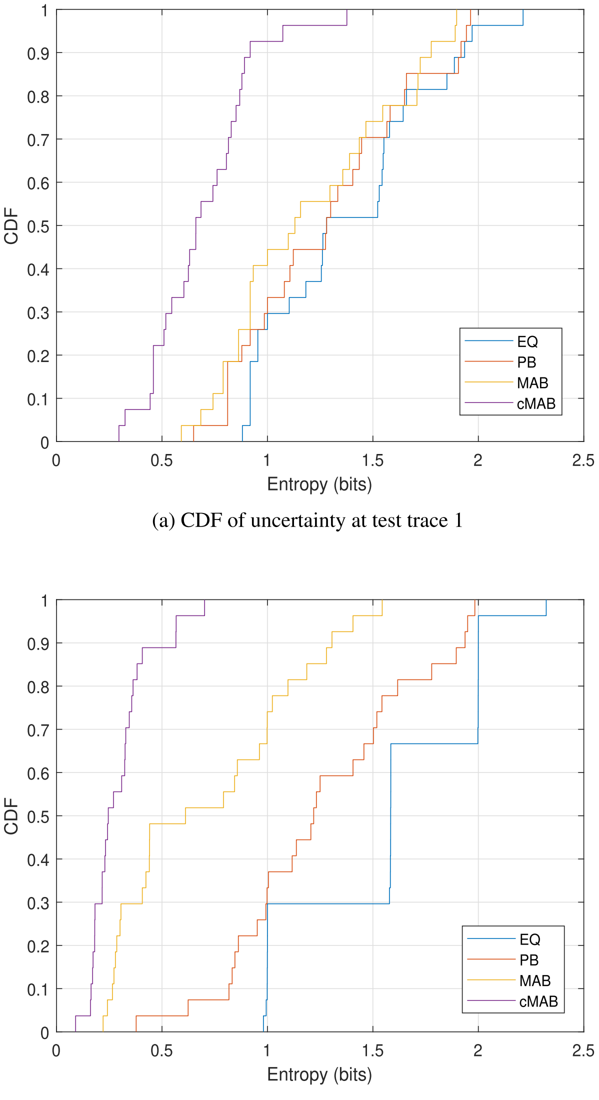
The following aspects will be shown in the results:

- *Cumulative prediction accuracy*: Denoting the total number of predictions as $Q_{prediction}$ and correct ones as $Q_{correct}$ of test trace n, the cumulative prediction accuracy PA up till trace n is defined as:

$$PA = \frac{\sum_{i=1}^{n} Q_{correct}}{\sum_{i=1}^{n} Q_{prediction}}$$

- *Cumulative distribution function (CDF) of uncertainty*: Aims to show uncertainty at the system level as well as some particular RSUs.
- *Proportion of Proactive Caching Content Fragments*: the proportion of the number of content fragments that are proactively cached and transmitted to vehicular users. This reflects the effectiveness of a proactive caching system.

We evaluate the network performance of the systems using *Proportion of Proactive Caching Content Fragments* as a figure of merit, instead of network delay, because the communication model considered in the paper does not model underlying transmission layers and backhaul links, given the focus of the paper is to find where to cache accurately.



(a) CDF of uncertainty at test trace 1

(b) CDF of uncertainty at test trace 200

FIGURE 7. Cumulative distribution function (CDF) of the overall uncertainty in Las Vegas. The figure demonstrates the reduction in uncertainty of the four proactive caching systems in the form of CDF of entropy.

2) EXPERIMENTAL RESULTS

As Las Vegas is the primary city, its results will first be discussed, followed by a more general demonstration of the secondary city Manchester. Fig. 7 shows the uncertainty analysis of four proactive caching systems in Las Vegas at a system level. It is the cumulative distribution of the uncertainty (entropy) of 32 RSUs at the end of test trace 1 and 200, respectively. These results illustrate performance before and after learning. The two bandit learning schemes, non-contextual MAB-based (MAB) and contextual MAB-based (cMAB), outperform the other two baseline schemes in terms of the reduced amount of uncertainty in decision-making. Both MAB and cMAB have dramatically reduced the uncertainty level through sufficient learning after 200 traces. The proportion of RSUs with entropy less than 0.5 bits has increased from 0% to 49% and 20% to 90%, respectively. The superiority of the cMAB-based system over its counterpart benefits from introducing the context information. Uncertainty distributions of bandit learning schemes were close to PB and EQ systems in the initial stage of simulation, but this gap has been enlarged in the end. The percentages of RSUs with less than 1-bit entropy are 100%, 80%, 40%, and 0% for cMAB, MAB, PB, and EQ respectively. The PB scheme has not experienced a significant change from this perspective because of its nature. Since the test traces simulate vehicles following their own daily commuting routines, the transition probability matrix or the weights used by PB scheme for decisions does not vary too much in the end of trace 1 and 200. By contrast, despite the fluctuations in the initial stage of simulation due to the lack of samples, the EQ scheme is constantly the one with the highest overall uncertainty and converges to a stable state finally. This makes sense from the viewpoint of information theory [43] as the entropy of an RSU with m neighbors is maximized to $\log_2 m$ with equal probability $\frac{1}{m}$ among the neighbors.

Fig. 8 shows the prediction accuracy (or hit ratio) in a cumulative way over the test traces. The accuracy superiority of bandit learning schemes over the PB and EQ is closely related to uncertainty reduction. Another point to explain this is that in bandit learning based schemes, RSUs make their decision on Q-values and the goal is to maximize the rewards. Therefore, fewer attempts are wasted on those actions that are less likely to be successful, whereas PB and EQ schemes, especially the latter one, attempt "bad" decisions more frequently. We shall see this in individual examples later. In addition, CPT+ is also shown in the figure, whose prediction performance is in between cMAB and MAB. In contrast to MAB, this makes sense since CPT+ relies greatly on a vehicle's past RSUs as a kind of context and this reduces the prediction uncertainty. However, it is outperformed by the cMAB as a model-free scheme with only one context (i.e., previous RSU) required. The MAB scheme reaches its limitation of 53% at a much earlier stage compared to cMAB with an upper bound of nearly 80%. CPT+ seems to have an increasing trend after test trace 200 and we can infer that it would reach the performance of cMAB perhaps at test trace 500 because the performance of CPT+ depends on its model: the more data, the better model. However, this is also its limitation in terms of adaptability and flexibility. It is also observed that the introduction of contextual information helps RSUs make more accurate decisions throughout the simulation cycle and meanwhile, it takes relatively longer to fully train the model and converge due to this fact.

Although Fig. 7 and Fig. 8 have demonstrated the potential interaction between prediction accuracy and uncertainty reduction, different RSUs may show significantly different variations on these two metrics. In Fig. 9 we have selected 4 types of RSUs according to the number of their actions/neighbors. From the top to the bottom row, they are RSUs with 5 actions, 4 actions, 3 actions, and 2 actions, respectively. The left column is the uncertainty CDF of relevant RSUs in an aggregated way. For example, there

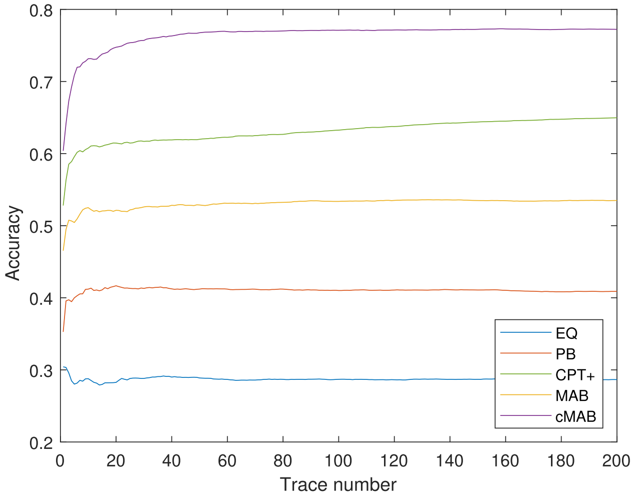


FIGURE 8. Cumulative prediction accuracy of the proactive caching systems in Las Vegas.

are 6 RSUs with 4 actions in our system. To achieve the plot on the left-hand side, we have collected their uncertainty at the end of each test trace, resulting in 200 by 6 samples for the CDF plot. Note that there is only 1 RSU with 5 actions. Similarly, the right-hand column shows the cumulative prediction accuracy of the corresponding RSUs also in an aggregated way. The prediction accuracy of test trace 10 of 4-action RSUs is $\frac{\sum_1^6 \sum_1^{10} Q_{correct}}{\sum_1^6 \sum_1^{10} Q_{prediction}}$. Both columns share the same legend shown in the bottom left corner. In general, the distribution of uncertainty of test traces still supports the inner connection seen in Fig. 7 and 8. Although it may be difficult to quantify the benefits of the reduction in uncertainty to prediction accuracy at this point, it helps visualize such benefits.

Even with the same number of neighbors, RSUs may show completely different performance in terms of uncertainty and prediction accuracy, possibly depending on their geographical location, traffic patterns, connectivity patterns, etc. In Fig. 10, we have selected RSU 2 and RSU 22 from the map in Fig. 5a, both of which have two neighboring RSUs (actions to be more precise) with unbalanced traffic. Over the 200 test traces, there are 73% and 27% of the 1116 handovers from RSU 2 to its two neighbors respectively, and RSU 22 also has the same proportion based on 2872 handovers. Despite this, proactive caching schemes have shown significantly different performance on these two RSUs and we have summarized in the table of Fig. 10 some statistical data at the end of the simulation. Without additional context introduced, we believe there is an unknown inherent success rate of each action for non-contextual schemes (EQ, PB, and MAB), denoted as θ^*. For the action 1 of RSU 2, θ^* can be approximately 80% according to the table as the success rates of all the three schemes tend to converge to 80%. For action 2, however, there does not seem to have a clear converging success rate, but we can infer that it could be 18% as in EQ scheme. The reason that PB and MAB have higher performance for action 2 is because they have fewer selections on action 2 than EQ, referred to as the "Count" row. Precisely because of this, during the learning process, MAB leans towards action 1 as it tends to have a better Q-value than action 2, and hence much fewer wrong decisions are made, resulting in a 78% overall accuracy. On the other hand, θ^* for action 1 and action 2 of RSU 22 is tending to converge to somewhere around 50%. Consequently, the MAB scheme is unable to tell which action would be a better one as they both have similar Q-values and it shows basically the same prediction performance as EQ and PB.

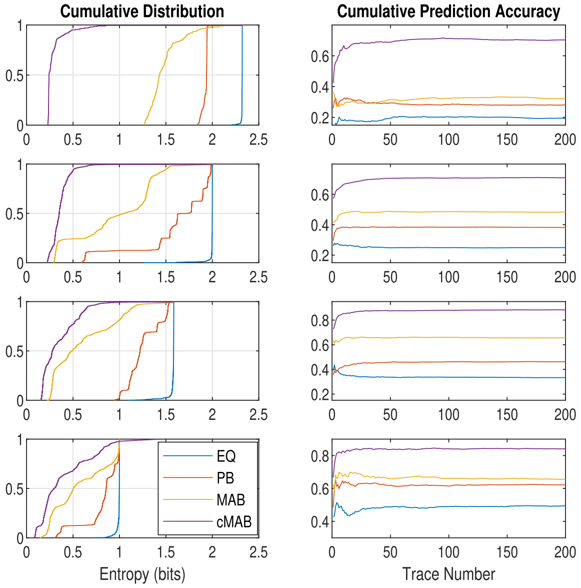
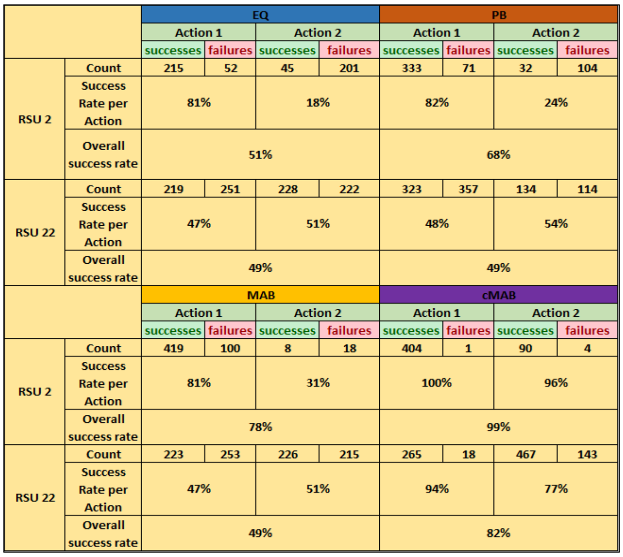
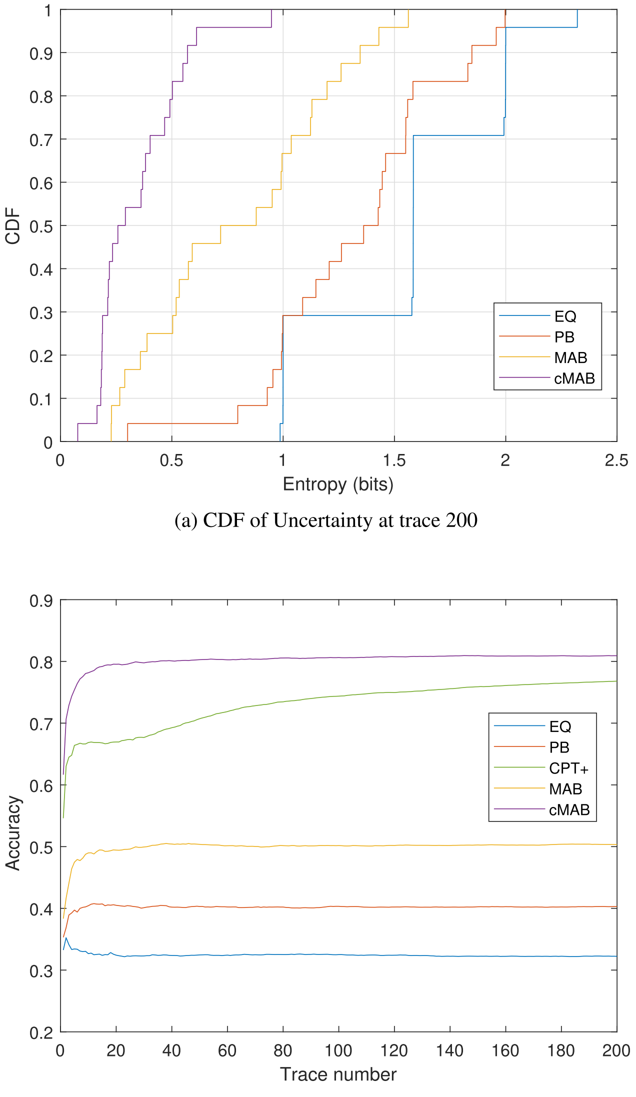


FIGURE 9. Performance of RSUs with different numbers of actions at the end of the simulation in Las Vegas. From the top to bottom, they are RSUs with 5, 4, 3, and 2 actions, respectively. The left column is the CDF of uncertainty (entropy) of these RSUs and the right column is the cumulative accuracy. The same legend is shared by two columns. The significance of the figure is that it demonstrates that smaller uncertainty results in higher accuracy (horizontally).



		EQ				PB			
		Action 1		Action 2		Action 1		Action 2	
		successes	failures	successes	failures	successes	failures	successes	failures
RSU 2	Count	215	52	45	201	333	71	32	104
	Success Rate per Action	81%		18%		82%		24%	
	Overall success rate	51%				68%			
RSU 22	Count	219	251	228	222	323	357	134	114
	Success Rate per Action	47%		51%		48%		54%	
	Overall success rate	49%				49%			
		MAB				cMAB			
		Action 1		Action 2		Action 1		Action 2	
		successes	failures	successes	failures	successes	failures	successes	failures
RSU 2	Count	419	100	8	18	404	1	90	4
	Success Rate per Action	81%		31%		100%		96%	
	Overall success rate	78%				99%			
RSU 22	Count	223	253	226	215	265	18	467	143
	Success Rate per Action	47%		51%		94%		77%	
	Overall success rate	49%				82%			

FIGURE 10. Statistics of two RSUs in Las Vegas. The table in this figure shows the accuracy of two RSUs with two actions as well as the improvement in prediction accuracy among the four proactive caching systems.



(a) CDF of Uncertainty at trace 200

(b) Cumulative Prediction Accuracy

FIGURE 11. **Cumulative distribution function (CDF) of the overall uncertainty of four proactive caching systems at the end of simulation and prediction accuracy of all the systems in manchester.**

It is obvious that the introduction of additional contextual information in cMAB has dramatically increased not only the success rate of each action of RSU 2 and RSU 22 but also their overall prediction accuracy to 99% and 82%, respectively. In particular, compared to its counterpart MAB, it has resolved the dilemma with RSU 22 where both actions have similar inherent θ^*. In stead of "hesitating" between the two actions, RSU 22 learns policy under different contexts in cMAB and becomes more certain about which action is likely to be correct. This is even more convincing for the case of RSU 2, where both actions have over 96% accuracy.

The system performance of Manchester is shown in Fig. 11, as a secondary city for generalizing the application. Similarly, we show the distribution of uncertainty among RSUs of four systems in the end of test trace 200 in Fig. 11a and the cumulative prediction accuracy of all five systems in Fig. 11b. Bandit learning-based schemes still show comparable benefits to that in Las Vegas, especially cMAB whose prediction accuracy has reached 80%. The performance has successfully demonstrated the adaptability of the proposed bandit learning schemes in a relatively more complex transportation network. One of the reasons for this is that the proposed algorithms only rely on information from the vehicular network itself for proactive caching decisions instead of taking additional information from the road network. Despite the advantages over the other two non-contextual systems (EQ and PB) as before in Manchester, we clearly notice the performance limitation of non-contextual MAB in contrast to its counterpart MAB scheme. CPT+ still shows similar relative performance to cMAB and MAB but has a faster growth rate compared to Las Vegas. This might be because of the relative area size and traffic pattern difference between the two cities (which will be explained in detail shortly).

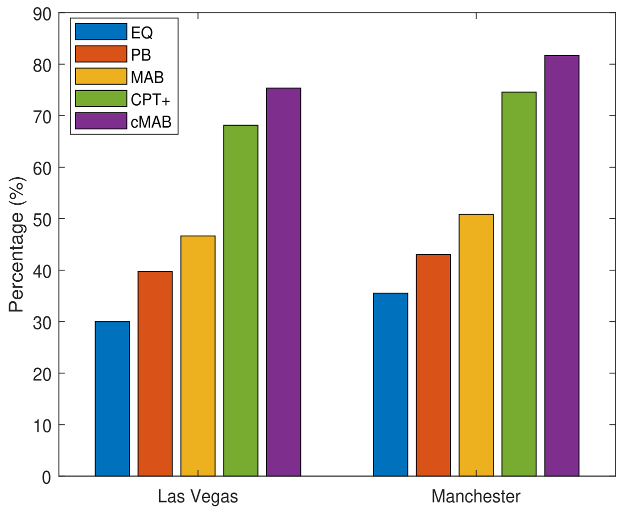


FIGURE 12. **Percentage of the average number of fragments served by proactive caching at the end of the simulation. The figure illustrates that higher cumulative prediction accuracy results in better proactive caching performance reflected by higher proportions of content fragments through caches.**

One of the major goals of proactive caching in vehicular networks is to provide vehicular users with seamless content delivery by bringing the content close to them accurately. We measure the number of fragments transmitted directly from RSU caches to vehicular users and plot a bar chart of the *proportion of the average fragments served by proactive caching* for each of the proactive caching schemes of two cities in Fig. 12. Overall, the proportions of both cities are consistent with the cumulative prediction accuracy, and the cMAB scheme demonstrates remarkable superiority over the other four. On average, it has achieved 75% in Las Vegas and 81% in Manchester, nearly double that of EQ and PB systems. We can also conclude that our proposed proactive caching schemes perform similarly irrespective of the road topology. Note-worthily, the proportions in the two cities are based on the different absolute total number of fragments transmitted to vehicles (around 1300 in Las Vegas and 750 in Manchester, varying trace by trace). This is because a) the Manchester area is relatively smaller than the Las Vegas area as a whole, b) the connectivity patterns of the two cities are distinct, and c) vehicles' content request pattern and frequency are different from test trace to trace of two cities. However, as the relative size of the center of the two areas has been kept on a similar level, this is still an effective contrast.

VII. DISCUSSION

A. ANALYSIS OF THE ADVANTAGE OF MAB-BASED ALGORITHMS

We can seek the theoretical accuracy of the previous prediction algorithms. Assume a vehicle v connecting to an RSU m with N actions. There exists an unknown probability distribution of v actually going to the N neighbors after m, denoted as $\mathcal{A} = [a_1, a_2, \ldots, a_n], n \in N$ and $\sum_{n\in N} a_n = 1$. If the RSU m makes a prediction with $\mathcal{B} = [b_1, b_2, \ldots, b_n], n \in N$, then the chance that this is a correct prediction can be computed by $\mathcal{P} = \mathcal{A} \cdot \mathcal{B} = \sum_{n\in\mathcal{N}} a_n \times b_n$. Depending on which algorithm, $\mathcal{B}$ is different. In the most under-performed one i.e., Equal-probability algorithm, $\mathcal{B}$ is uniform distribution i.e., $b_1 = b_2 = \cdots = b_n = \frac{1}{N}$ and thus $\mathcal{P} = \sum_{n\in\mathcal{N}} a_n \times b_n = \frac{1}{N} \times \sum_{n\in\mathcal{N}} a_n = \frac{1}{N}$. In the Probability-based algorithm, $\mathcal{B}$ is the transition probabilities derived from previous traces, where $b_1 \neq b_2 \neq \ldots \neq b_n$, and therefore $\mathcal{P}$ remains to be $\mathcal{P} = \sum_{n\in\mathcal{N}} a_n \times b_n$. If the traffic pattern through RSU m does not change significantly over time, we can assume $a_n = b_n$, so $\mathcal{P} = \sum_{n\in\mathcal{N}} b_n^2$. In non-contextual MAB, $\mathcal{B}$ depends on Q-values and action selection algorithm (i.e., ϵ-greedy). Therefore, the probability b_n of its neighbor $n \in N$ to be predicted as the next RSU is: $b_n = \begin{cases} 1-\epsilon, & \text{if } n \text{ has the highest } Q\text{-value} \\ \epsilon \cdot \frac{1}{N}, & \text{Otherwise} \end{cases}$. Take an example of an RSU of two action choices (neighbors) with uneven traffic patterns (e.g., 80% vs 20%). Its theoretical accuracy with the Equal-probability algorithm is 50% since it has 2 neighbors. Because it has an uneven traffic pattern where one of its neighbors has approximately 80% traffic, the theoretical accuracy with Probability-based algorithm $\mathcal{P} = 80\% \times 80\% + 20\% \times 20\% = 68\%$. It is because of this traffic pattern that MAB has a dominant action and therefore, the overall theoretical accuracy is $\mathcal{P} = 80\% \times (1-\epsilon) + 20\% \times \frac{\epsilon}{2} = 77\%$, where $\epsilon = 0.05$. The cMAB algorithm further expands the advantage of MAB and reduces uncertainty by breaking it down into context level, hence resulting in an even higher optimal boundary. The simulated result of RSU 2 in Fig. 9 is consistent with the theoretical values and this can be extended to other RSUs with different numbers of actions.

Furthermore, another notable advantage of the proposed cMAB and MAB algorithms is their natural capabilities of coping with sudden major changes in the topology or vehicular environment by rapidly adjusting Q-tables and policies, whereas Probability-based and CPT+ based algorithms, become very clumsy in this regard due to high reliance on past data to establish their models.

B. ALGORITHM COMPLEXITY AND SCALABILITY

1) SPACE COMPLEXITY

Theoretically, the two proposed MAB-based algorithms have advantages in terms of space/computational complexity. For the three non-contextual algorithms i.e., MAB, Equal-probability, and Probability-based, the Probability-based one has the highest computational complexity. This is because RSUs in this algorithm requires some extra computational resources to store historical traffic information in order to establish a probability distribution over their actions. However, MAB algorithm is a localized algorithm where the RSUs' Q-tables get in-place updates, and individual RSUs have their own fixed probability distribution for prediction in the Equal-probability algorithm. Although cMAB algorithm is also a localized algorithm as MAB, it does require RSUs to build context-related Q-tables and therefore, needs slightly more space than MAB. Nevertheless, this is worthwhile given the significantly reduced uncertainty and improved prediction accuracy by cMAB. CPT+ based algorithm, however, consumes the most resources because it requires building a large prediction tree model to achieve a certain prediction accuracy, which is still outperformed by cMAB. Such an advantage also makes it practical for the implementation of MAB and cMAB algorithms.

2) TIME COMPLEXITY

The three main functionalities in the proposed MAB and cMAB algorithms are: A - *Next RSU selection* (including ϵ -greedy), B - *Pre-caching content* and C - *Q-table updating with rewards*. From the perspective of actual code implementation, for MAB algorithm, an agent RSU with k actions requires $O(k)$, $O(1)$, and $O(k)$ time complexity for function A, B and C respectively. This is because functions A and C require action set traversal whereas B only needs insertion manipulation with a vector. In addition, as functions A, B and C are executed sequentially, they account for a $O(k)$ complexity. The system may have multiple RSUs but due to the nature of event-driven simulation, only one of them is "working" at a time. Therefore, assuming the largest action set of these RSUs is K, then the overall performance of N-length test can be represented by $O(NK)$. The major difference between the two lies in the additional context s. Specifically, functions A and C are executed based on s once it is detected. But this works in the same way as in a non-contextual MAB and therefore, their complexity is identical to that in MAB for an arbitrary RSU. Function B remains the same as well. Apart from this, cMAB algorithm also involves context detection and creation and these additional manipulations account for $O(1)$ complexity. Thus, cMAB has the same overall complexity, that is $O(NK)$ as above.

3) SCALABILITY

The proposed MAB-based algorithms in a network with a large number of RSUs are very scalable. This is because MAB learning is a model-free stateless framework.

A learning agent normally has a finite set of actions and does not need to estimate the states of the environment. This is also true for individual RSUs in a vehicle network because no matter how large the network is, a RSU always has a finite set of neighboring RSUs. Even for the proposed contextual cMAB learning algorithm where RSUs need to detect the previous RSU that a vehicle hands off from, the set of context is also finite since the previous RSU is also one of the neighboring RSUs. Therefore, this property results in very good scalability of the proposed algorithms.

C. CONVERGENCE

The cumulative prediction accuracy in Fig. 8 and Fig. 11b demonstrates the convergence of the proposed MAB-based proactive algorithms. Although a cumulative way to show this may not be perfect, it is still sufficient to demonstrate the performance boundary in the commuting traffic scenario considered in this paper. From the system level, theoretically, the cMAB algorithm should converge slower than the non-contextual MAB because given a statistically fixed number of Q-table updates (identical test traces) for an RSU fewer updates are allocated to each individual context in cMAB in contrast to MAB where all the updates are used for only one Q-table. This difference in convergence can be found in the previously mentioned results.

During the learning process, Q-values or Q-tables of individual RSUs may converge to rather different values depending on the traffic pattern through it. For example, in the non-contextual MAB algorithm, we have noticed that a high-accuracy RSU (over 90%) with 4 actions have a converged Q-table with values: $\langle -0.9375, -1, -0.9961, 1\rangle$ at an early stage of the learning process. This demonstrates a convergence to the last action and that there may exist very deterministic routes for all the vehicles through this RSU. On the other hand, it has also been found that an average-accuracy RSU (approximately 50%) with the same number of choices has a Q-table with values: $\langle -1, -1, -0.5643, -0.4379\rangle$. Throughout the learning process, the RSU tried to converge to the best action by trial and error but failed to do so because the last two actions are almost evenly good. This implies the dilemma in non-contextual MAB and should be resolved by contextual MAB exploiting the additional contexts available.

VIII. CONCLUSION

This paper studies how to achieve mobility-prediction based proactive caching in vehicular networks by predicting the next RSU of the vehicle's path. As a way of addressing this, the paper has proposed two bandit learning-based proactive caching algorithms: *non-contextual* MAB and *contextual* MAB and compared their performance with three other baseline schemes: *Equal Probability-based*, *Probability-based*, and *Compact Prediction Tree+ based* proactive caching strategy. In addition to this, the *subjective logic* framework has been extended to study the uncertainty associated with different proactive caching systems. With this model, we have analyzed in detail the overall entropy distribution of the systems as well as the distribution of representative RSUs. Furthermore, two urban areas of Las Vegas and Manchester with different road layouts have been tested to demonstrate the adaptability of the proposed schemes to a diverse set of road layouts.

Simulation results have shown the advantages of the proposed proactive caching algorithms over their counterparts. Contextual MAB-based scheme yields the highest benefit to the system thanks to the introduction of contextual information for uncertainty reduction. In both cities, the contextual MAB-based proactive caching scheme reaches a prediction accuracy of approximately 80% compared to roughly 50% of the non-contextual MAB-based scheme. As a result of this, the network performance is dramatically improved with contextual MAB in terms of the number of fragments directly transmitted by caches. The performance of bandit learning-based systems is similar in both cities regardless of road topology. Particularly, 75% and 81% content fragments are proactively served with contextual MAB algorithm and over 53% and 50% with non-contextual MAB algorithm in Las Vegas and Manchester, respectively.

REFERENCES

[1] H. Hartenstein and L. P. Laberteaux, "A tutorial survey on vehicular ad hoc networks," *IEEE Commun. Mag.*, vol. 46, no. 6, pp. 164–171, Jun. 2008.

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

[3] K. Zheng, L. Hou, H. Meng, Q. Zheng, N. Lu, and L. Lei, "Soft-defined heterogeneous vehicular network: Architecture and challenges," *IEEE Netw.*, vol. 30, no. 4, pp. 72–80, Jul./Aug. 2016.

[4] L. Hou, L. Lei, K. Zheng, and X. Wang, "A Q-learning-based proactive caching strategy for non-safety related services in vehicular networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4512–4520, Jun. 2018.

[5] N. Nakamura, Y. Niimi, and S. Ishihara, "Live VANET CDN: Adaptive data dissemination scheme for location-dependent data in VANETs," in *Proc. IEEE Veh. Netw. Conf.*, Dec. 2013, pp. 95–102.

[6] F. A. Silva, A. Boukerche, T. R. M. B. Silva, L. B. Ruiz, E. Cerqueira, and A. A. F. Loureiro, "Vehicular networks: A new challenge for content-delivery-based applications," *ACM Comput. Surv.*, vol. 49, no. 1, pp.1–29, Jun. 2016, doi: 10.1145/2903745

[7] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1657–1681, 3rd Quart., 2017.

[8] J. Zhang and K. B. Letaief, "Mobile edge intelligence and computing for the internet of vehicles," *Proc. IEEE*, vol. 108, no. 2, pp. 246–261, Feb. 2019.

[9] N. Morozs, T. Clarke, and D. Grace, "Distributed heuristically accelerated Q-learning for robust cognitive spectrum management in LTE cellular systems," *IEEE Trans. Mobile Comput.*, vol. 15, no. 4, pp. 817–825, Apr. 2016.

[10] A. Mahajan and D. Teneketzis, "Multi-armed bandit problems," in *Foundations and Applications of Sensor Management*. Springer, 2008, pp. 121–151, doi: 10.1007/978-0-387-49819-5_6.

[11] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[12] D. Bouneffouf and I. Rish, "A survey on practical applications of multi-armed and contextual bandits," 2019, *arXiv:1904.10040*.

[13] E. Hüllermeier and W. Waegeman, "Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods," 2019, *arXiv:1910.09457*.

[14] A. Jøsang, "Artificial reasoning with subjective logic," in *Proc. 2nd Austral. Workshop Commonsense Reasoning*. Princeton, NJ, USA: Citeseer, vol. 48, 1997, p. 34.

[15] A. Josang, *Subjective Logic: A Formalism for Reasoning Under Uncertainty* (Artificial Intelligence: Foundations, Theory, and Algorithms). Berlin, Germany: Springer, 2016. [Online]. Available: https://books.google.no/books?id=bJkJkAEACAAJ

[16] F. M. Zennaro and A. Jøsang, ''Using subjective logic to estimate uncertainty in multi-armed bandit problems,'' 2020, *arXiv:2008.07386*.

[17] M. Sensoy, L. Kaplan, and M. Kandemir, ''Evidential deep learning to quantify classification uncertainty,'' 2018, *arXiv:1806.01768*.

[18] L. Kaplan, F. Cerutti, M. Sensoy, A. Preece, and P. Sullivan, ''Uncertainty aware AI ML: Why and how,'' 2018, *arXiv:1809.07882*.

[19] H. Khelifi, S. Luo, B. Nour, A. Sellami, H. Moungla, and F. Naït-Abdesselam, ''An optimized proactive caching scheme based on mobility prediction for vehicular networks,'' in *Proc. IEEE GLOBECOM*, Dec. 2018, pp. 1–6.

[20] Q. Wang and D. Grace, ''Sequence prediction-based proactive caching in vehicular content networks,'' in *Proc. IEEE 3rd Connected Automated Vehicles Symp. (CAVS)*, Dec. 2020, pp. 1–6.

[21] N. B. Hassine, P. Minet, D. Marinca, and D. Barth, ''Popularity prediction–based caching in content delivery networks,'' *Ann. Telecommun.*, vol. 74, no. 5, pp. 351–364, 2019.

[22] H. S. Goian, O. Y. Al-Jarrah, S. Muhaidat, Y. Al-Hammadi, P. Yoo, and M. Dianati, ''Popularity-based video caching techniques for cache-enabled networks: A survey,'' *IEEE Access*, vol. 7, pp. 27699–27719, 2019.

[23] S. Yue and Q. Zhu, ''A mobility prediction-based relay cluster strategy for content delivery in urban vehicular networks,'' *Appl. Sci.*, vol. 11, no. 5, p. 2157, 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/5/2157

[24] L. Yao, A. Chen, J. Deng, J. Wang, and G. Wu, ''A cooperative caching scheme based on mobility prediction in vehicular content centric networks,'' *IEEE Trans. Veh. Technol.*, vol. 67, no. 6, pp. 5435–5444, Jun. 2018.

[25] Z. Zhao, L. Guardalben, M. Karimzadeh, J. Silva, T. Braun, and S. Sargento, ''Mobility prediction-assisted over-the-top edge prefetching for hierarchical VANETs,'' *IEEE J. Sel. Areas Commun.*, vol. 36, no. 8, pp. 1786–1801, Aug. 2018.

[26] T. Gueniche, P. Fournier-Viger, R. Raman, and V. S. Tseng, ''CPT+: Decreasing the time/space complexity of the compact prediction tree,'' in *Advances in Knowledge Discovery and Data Mining*, T. Cao, E.-P. Lim, Z.-H. Zhou, T.-B. Ho, D. Cheung, and H. Motoda, Eds. Cham, Switzerland: Springer, 2015, pp. 625–636.

[27] B. J. A. Kröse, ''Learning from delayed rewards,'' *Robot. Auto. Syst.*, vol. 15, no. 4, pp. 233–235, Oct. 1995.

[28] G. A. Rummery and M. Niranjan, *On-Line Q-Learning Using Connectionist Systems*, vol. 37. Cambridge, U.K.: Univ. of Cambridge, Department of Engineering, 1994.

[29] P. Dai, Z. Hang, K. Liu, X. Wu, H. Xing, Z. Yu, and V. C. S. Lee, ''Multi-armed bandit learning for computation-intensive services in MEC-empowered vehicular networks,'' *IEEE Trans. Veh. Technol.*, vol. 69, no. 7, pp. 7821–7834, Jul. 2020.

[30] Y. Miao, Y. Hao, M. Chen, H. Gharavi, and K. Hwang, ''Intelligent task caching in edge cloud via bandit learning,'' *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 1, pp. 625–637, Jan. 2021.

[31] S. Maghsudi and E. Hossain, ''Multi-armed bandits with application to 5G small cells,'' *IEEE Wireless Commun.*, vol. 23, no. 3, pp. 64–73, Jun. 2016.

[32] X. Xu, M. Tao, and C. Shen, ''Collaborative multi-agent multi-armed bandit learning for small-cell caching,'' *IEEE Trans. Wireless Commun.*, vol. 19, no. 4, pp. 2570–2585, Apr. 2020.

[33] T. Müller, ''An unforeseen equivalence between uncertainty and entropy,'' in *Proc. IFIP Int. Conf. Trust Manage.* Springer, 2019, pp. 57–72.

[34] T. Stockhammer, ''Dynamic adaptive streaming over HTTP–standards and design principles,'' in *Proc. 2nd Annu. ACM Conf. Multimedia Syst.*, 2011, pp. 133–144.

[35] C. Claus and C. Boutilier, ''The dynamics of reinforcement learning in cooperative multiagent systems,'' in *Proc. AAAI/IAAI*, vol. 1998, nos. 746–752, 1998, pp. 1–7.

[36] D. Russo, B. Van Roy, A. Kazerouni, I. Osband, and Z. Wen, ''A tutorial on Thompson sampling,'' 2017, *arXiv:1707.02038*.

[37] S. Bubeck and N. Cesa-Bianchi, ''Regret analysis of stochastic and non-stochastic multi-armed bandit problems,'' 2012, *arXiv:1204.5721*.

[38] B. Xin, H. Yu, Y. Qin, Q. Tang, and Z. Zhu, ''Exploration entropy for reinforcement learning,'' *Math. Problems Eng.*, vol. 2020, pp. 1–12, Jan. 2020.

[39] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, ''Microscopic traffic simulation using SUMO,'' in *Proc. 21st IEEE Int. Conf. Intell. Transp. Syst.*, Nov. 2018, pp. 2575–2582. [Online]. Available: https://elib.dlr.de/124092/

[40] *Version 9.4.0 (R2018A)*, MATLAB, MathWorks, Natick, MA, USA, 2018.

[41] G. A. Wainer and P. J. Mosterman, *Discrete-Event Modeling and Simulation: Theory and Applications*. Boca Raton, FL, USA: CRC Press, 2018.

[42] M. Bennis and D. Niyato, ''A Q-learning based approach to interference avoidance in self-organized femtocell networks,'' in *Proc. IEEE Globecom Workshops*, Dec. 2010, pp. 706–710.

[43] C. E. Shannon, ''A mathematical theory of communication,'' *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, Jul./Oct. 1948.

QIAO WANG (Student Member, IEEE) received the master's degree from the Tampere University of Technology (now Tampere University), Finland, in 2014. He is currently pursuing the Ph.D. degree with the Communication Technologies Research Group, Department of Electronic Engineering, University of York, under the supervision of Prof. David Grace. Since 2015, he has been working as a LTE System Developer with Ericsson, China. His current research interests include proactive caching, vehicular networks, mobility prediction, and reinforcement learning.

DAVID GRACE (Senior Member, IEEE) received the Ph.D. degree from the University of York, in 1999. His Ph.D. thesis was on ''Distributed Dynamic Channel Assignment for the Wireless Environment.'' From 2014 to 2018, he was a Non-Executive Director of Stratospheric Platforms Ltd. In 2000, he jointly founded SkyLARC Technologies Ltd., and was one of its directors. Since 1994, he has been a member of the Department of Electronic Engineering, University of York, where he is currently a Professor (Research), the Head of the Communication Technologies Research Group, and the Director of the Centre for High Altitude Platform Applications. He is also a lead investigator on H2020 MCSA SPOTLIGHT, U.K. Government funded MANY, dealing with 5G trials in rural areas, and HiQ investigating Quantum Key Distribution from high altitude platforms. He was a Technical Lead on the 14-partner FP6 CAPANINA Project that deal with broadband communications from high altitude platforms. He is the author of more than 280 papers and author/editor of two books. His current research interests include aerial platform-based communications, application of artificial intelligence to wireless communications, 5G system architectures, dynamic spectrum access, and interference management. He is the Former Chair of the IEEE Technical Committee on Cognitive Networks, from 2013 to 2014, and a Founding Member of the IEEE Technical Committee on Green Communications and Computing.