IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

A Hybrid Proactive Caching System in Vehicular Networks based on Contextual Multi-armed Bandit Learning

QIAO WANG, *Student Member, IEEE*, **DAVID GRACE** *Senior Member, IEEE*
Communication Technologies Research Group, Institute for Safe Autonomy and School of Physics, Engineering and Technology, University of York, York YO10 5DD, United Kingdom (e-mail: qiao.wang@york.ac.uk, david.grace@york.ac.uk)

ABSTRACT Proactive edge caching has been regarded as an effective approach to satisfy user experience in mobile networks by providing seamless content transmission and reducing network delay. This is particularly useful in rapidly changing vehicular networks. This paper addresses the proactive edge caching (at the roadside unit (RSU)) problem in vehicular networks by mobility prediction, i.e., the next RSU prediction. Specifically, the paper proposes a distributed *Hybrid cMAB Proactive Caching System* where RSUs act as independent learners that implement two parallel online reinforcement learning-based mobility prediction algorithms between which they can adaptively finalize their predictions for the next RSU. The two parallel prediction algorithms are based on *Contextual Multi-armed bandit* (cMAB) learning, called *Dual-context cMAB* and *Single-context cMAB*. The hybrid system is further developed into two variants: *Vehicle-Centric* and *RSU-Centric*. In addition, the paper also conducts comprehensive simulation experiments to evaluate the prediction performance of the proposed hybrid system. They include three traffic scenarios: *Commuting traffic, Random traffic* and *Mixed traffic* in Las Vegas, USA and Manchester, UK. With the different road layouts in the two urban areas, the paper aims to generalize the application of the system. Simulation results show that the hybrid Vehicle-Centric system can reach nearly 95% cumulative prediction accuracy in the Commuting traffic scenario and outperform the other methods used for comparison by reaching nearly 80% accuracy in Mixed traffic scenario. Even in the completely Random traffic scenario, it also guarantees a minimum accuracy of nearly 60%.

INDEX TERMS proactive edge caching, reinforcement learning, multi-armed bandit learning, mobility prediction, vehicular networks, roadside units(RSUs)

I. INTRODUCTION

THE automobile industry has been making road vehicles more and more intelligent over the past decade, thanks to the development in electronics and communication technologies. Vehicles are embedded with onboard units (OBUs) and able to communicate with road infrastructures e.g., roadside units (RSUs), and even with other vehicles. What is even more incredible is the upcoming era of electric and autonomous vehicles. This means that the vehicle no longer provides just transportation in the traditional sense but will become a mobile information and entertainment center [1], [2]. All of these are essential elements of vehicular networks that are considered as one of the most important enabling technologies of the next generation intelligent transportation system [3].

However, such a revolution also poses unprecedented challenges to conventional vehicular networks from the perspective of content transmission. Currently, tremendous data demands from vehicular users are satisfied by the remote content provider through network infrastructure such as RSUs. This inevitably causes problems such as high network latency and poor quality of experience for the users, given the limitation of link capacity and bandwidth resources [4]. In addition to this, as fast-moving objects, vehicles may experience frequent intermittent connections with RSUs, which results in a rapidly changing vehicular environment. High-speed mobility causes frequent link re-connections, which means that a content transmission between a vehicular user and an RSU may not be completed within the coverage of the RSU and the user has to re-request the remaining content after reconnecting to a new RSU at a dramatically reduced data rate [3], [4].

The edge caching technique, which brings content closer to end users, is considered to be an effective approach to resolve the challenge of network latency and backbone network congestion due to a massive amount of remote requests to the content provider. On top of this, *proactive edge caching* has been recognized as a promising solution to the intermittent connectivity challenges caused by the highly dynamic vehicular network. It not only provides content close to the vehicular users but also predicts where they may need content in advance through prediction algorithms. Proactively caching the desired content at the future RSU(s) beforehand allows vehicles to continue their earlier incomplete content transmissions immediately after accessing the new RSU without having to request the content again from the remote server. Thanks to the rapid development of mobile edge intelligence, *mobile edge computing* (MEC) [5] servers deployed at the network edge (i.e., RSUs) are the key enabler of proactive caching by providing both local storage and computation functionalities, where the computation is crucial in regard to mobility prediction.

As the name implies, proactive caching relies on predictions. Since the focus of this paper is proactive edge caching at the targeted RSU, the problem then becomes predicting the next RSU that is most proper to perform proactive caching. For this purpose, machine learning (ML) techniques can be useful. In the literature, most studies using ML models for this purpose used recurrent neural networks (e.g., [4] and [6]). However, one disadvantage of them is the high reliance on the offline training stage, which limits their adaptability in a time-varying environment. Vehicular networks, however, give rise to a rapidly changing environment. Therefore, it is meaningful to find an online learning approach for the purpose of increasing the adaptability of the prediction algorithms. Reinforcement learning (RL) provides this option and in fact, this problem can be seen as a direct application of RL because every prediction is a decision to make. The *agent* in RL learns in a trial and error manner and tries to learn a *policy* that is usually associated with *states* and *actions*. None of the past work investigated the effectiveness of RL techniques in next-RSU prediction-based proactive caching, except for our previous paper [7] which proposed *Multi-armed bandit* (MAB) learning [8], [9] and contextual MAB (cMAB) learning algorithms to address the problem. MAB learning is a special instance of RL and it is single-state and model-free RL. The agents in MAB learning just have one state and no state transition (i.e., it is stateless), and do not have to build up a model of the environment. This significantly reduces the number of trials needed to learn a mature strategy, speeds up the learning process [7], [10], and solves the difficulty of representing every single state of the environment in the traditional RL, which is of great benefit to a dynamically changing vehicular environment. Excellent prediction performance of the single-context (i.e., one-dimensional) cMAB was achieved in [7] and we believe it is of great significance to keep exploring the potential of cMAB learning, which is the main motivation of this paper.

The objective of the paper is to address the proactive caching problem in vehicular networks using cMAB learning. Specifically, we develop a distributed *Hybrid cMAB Proactive Caching System* where each RSU in the network is an independent learning agent and predicts the next RSU for proactive caching for every connected vehicle as required. In the system, each independent RSU learner is enabled with adaptive prediction between its two underlying prediction algorithms: *Dual-context cMAB* and *Single-context cMAB*. Despite the earlier work in [7] which focused only on the single-context cMAB learning system, the motivation here is to design a hybrid system that can fully exploit the potential of both dual-context and single-context cMAB in order to seek better proactive caching performance in a variety of scenarios. This paper further fills the gap in the studies on using *independent multi-agent* contextual MAB to solve proactive caching problems. Specifically, the main contributions of the paper can be summarized as follows:

- We propose a *Hybrid cMAB Proactive Caching System* with a specifically designed switching mechanism to allow RSUs to adaptively finalize their predictions between the dual-context and single-context cMAB algorithms. The system is further developed into two variants: *Vehicle-Centric* System that realizes vehicle-level switching and *RSU-Centric* System with RSU-level switching, for comprehensive performance comparison.
- We design a Dual-context cMAB algorithm that utilizes vehicle ID and the previous RSU together as two-dimensional context. Together with the Single-context cMAB algorithm that uses previous RSU as context, they serve as the two underlying parallel prediction algorithms in the hybrid system. The hybrid system outperforms the single-context cMAB system proposed in [7] in various experimental scenarios.
- We extend traffic scenarios on top of [7] into *Commuting traffic, Random traffic, Mixed traffic*, in order to evaluate the system performance in a more comprehensive way. They are generated in two urban areas in Las Vegas, USA, and Manchester, UK with significantly differing road planning characteristics. The results demonstrate the adaptability of the proposed algorithms and systems to different road layouts.

The rest of the paper is structured as follows. In Section II, some related studies on proactive caching in vehicular networks and the applications of MAB in relevant fields are discussed. Section III introduces the architecture of the MEC-enabled vehicular network that this work is based on. The proposed hybrid cMAB system and the two parallel cMAB-based prediction algorithms are elaborated in Section IV. Section V discusses the simulation setup, traffic scenarios, and performance evaluation and analysis. Section VI conducts an extended study on an additional traffic scenario and provides in-depth insight into the proposed learning systems. Section VII concludes this 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 MAB in mobile networks*.

A. PROACTIVE CACHING IN VEHICULAR NETWORKS

Research on the problem of proactive caching in mobile networks can be broadly classified into two categories: what to cache and where to cache. To anticipate what to cache in advance mostly depends on content popularity prediction. Hassine *et al.* [11] used a two-level prediction model for video popularity prediction to pre-store popular videos in a content delivery network. Popularity-based video caching techniques in cache-enabled networks have been summarized in [12]. Nevertheless, the reliance on collecting vehicular users' personal data makes these methods less effective given the increasing restrictions and users' attention on security and privacy. Therefore, this paper focuses on solving where to cache problems by predicting where a vehicle is going next, more precisely the next RSU it is going to access. From the network operator's point of view, this is more manageable and applicable.

The most recent work on next-RSU proactive caching based on offline learning is in [13] where the authors proposed a sequence-prediction-based proactive caching system to address the problem. The model is based on the Compact Prediction Tree plus (CPT+) model [14], a sequence prediction algorithm, by training vehicle-specified simulated traffic traces. Hou *et al.* [4] and Khelifi *et al.* [6] both used the Long Short Time Memory (LSTM), a deep neural network model, to predict the direction of a vehicle is going and thus infer the next RSU instead of directly predicting it. For a similar purpose, Zhao *et al.* [15] used a hybrid Markov chain model for future RSU prediction, depending on the availability and quality of vehicles' traces. Yao *et al.* [16] also proposed using Prediction by Partial Matching (PPM), a tree-based Markov chain model, for mobility prediction of reaching different hot spot regions, but they concentrated on caching on individual vehicle nodes. Despite these meaningful studies, the first fundamental difference is that they all rely on massive offline training with labeled data in order to get a proper prediction model, which is the main limitation of their adaptability in a fast-changing environment. This work, however, focuses on online learning with a model-free learning algorithm. Additionally, in contrast to the centralized way of prediction in [4], [6], and [15], our approach considers a distributed system where RSUs are learning and predicting independently.

B. REINFORCEMENT LEARNING AND MAB IN MOBILE NETWORKS

One of the most widely used model-free RL techniques is *Q-Learning* proposed by Watkins [17]. However, a challenge of traditional Q-learning is its applicability to realistic vehicular environments. As mentioned earlier, traditional RL techniques are required to represent the states of the learning agent and this restricts its adaptability in mobile networks including vehicular networks. Therefore, it is helpful to consider the agents with a discrete action set as stateless in vehicular networks as this will potentially reduce the number of trials needed to learn a sophisticated strategy and improve the adaptability of RL-based cognitive devices (e.g., RSUs).

The MAB problem is representative of the stateless RL problem. While it has attracted significant attention in various applications ranging from recommendation systems and advertisement replacement to healthcare and finance [18], its application on proactive caching in vehicular networks and other mobile networks seem to be rare. To the best of our knowledge, our previous work in [7] is the only study that proposed two proactive caching schemes in vehicular networks based on MAB and cMAB. RSUs in [7] act as independent stateless learning agents and observe the previous RSU as the context in cMAB scheme. In addition, there are some applications of MAB to other aspects of mobile networks. Dai *et al.* [19] proposed a Utility-table based Learning algorithm based on MAB to solve distributed task assignment problem in a MEC-empowered vehicular network. The authors in [20] proposed an intelligent task caching algorithm based on MAB and evaluated its benefits to task caching latency performance in the edge cloud. Xu *et al.* [21] investigated collaborative caching problems in small-cell networks by learning the cache strategies directly at small base stations online by utilizing multi-agent MAB.

Despite the advantages of MAB learning, we believe it is worth more investigations in the area of proactive caching in vehicular networks. In particular, it is meaningful to exploit the potential of cMAB with contexts from different dimensions i.e., dual-context. Meanwhile, it is also practical to develop a hybrid system that can fully exploit the advantages of cMAB algorithms with different context dimensions. To the best of our knowledge, no prior study has focused on these technical aspects. The novelty of the present work is the proposed adaptive hybrid cMAB proactive caching system that exploits both dual-context cMAB and single-context cMAB algorithms, and the evaluation of system performance using this approach under various realistic-like traffic scenarios.

III. NETWORK ARCHITECTURE

The vehicular network considered in the paper is deployed with RSUs that are MEC-enabled, as depicted in Figure 1. The RSUs are capable of edge computing and caching with MEC servers. With computing units, they are intelligent to learn and predict the next possible RSU a vehicular user may connect to next and the caching units enables them to pre-caching content when the pre-caching request is received from other RSUs. Vehicular users frequently request content from RSUs after they enter the network. Despite the equipped MEC servers, computing resource consumption and content replacement techniques are out of the scope of this paper.

Consider a vehicular network $\mathcal{G}$ in an urban area with M RSUs in a set $\mathcal{M} = \{m_1, m_2, ..., m_M\}$. There are residential areas and workplace areas in $\mathcal{G}$ where L vehicles in the set

$\mathcal{V} = \{v_1, v_2, ..., v_L\}$ depart and arrive on a daily basis. An RSU $m_i \in \mathcal{M}$ has neighboring RSUs and it predicts the next RSU by selecting one of its neighbors. In addition, a central node is available to help coordinate RSUs in a distributed way. One of its main responsibilities is to observe the result of a previous prediction and feedback a reward to a prior RSU so that the RSU can refine its learning policies (which shall be discussed in the next section). Furthermore, a content database $\mathcal{C} = \{c_1, c_2, ..., c_K\}$ exists in the Content Provider that stores K types of content with various sizes, represented by $f_{c_k \in \mathcal{C}}$ fragments, each of which is of size F_c.

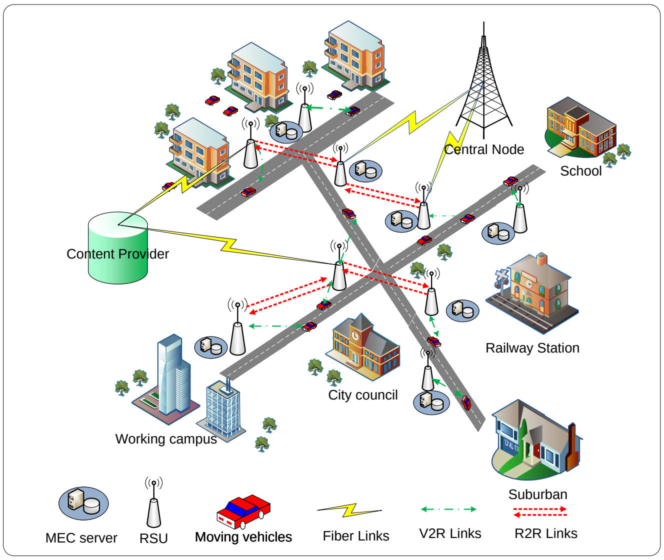


FIGURE 1: Architecture of MEC-enabled vehicular network (redrawn from [7])

The communication model implemented in this paper only characterizes some basic features of transmission because the goal of the work is to anticipate where to cache precisely. Therefore, the following assumptions and simplifications are made:

- A vehicle connects to the geographically closest RSU
- Problems such as interference and re-transmission in the underlying layers (e.g., physical and medium access control layers) in vehicular communications are not considered in this paper and thus the transmission rate e is a constant
- The dwell time of vehicles in an RSU is extracted from the test trace being simulated and is known so that the number of content fragments can be derived
- The network is completely proactive which means that content will not be cached in a reactive way
- A vehicle will not request new content until it finishes consuming the current one; when handover occurs, the vehicle continues its unfinished transmission

A representative proactive caching procedure can be described as follows. After a vehicle $v_i \in \mathcal{V}$ accesses an RSU $m_i \in \mathcal{M}$, m_i uses the prediction algorithm to predict the next RSU that the vehicle is likely to access next, say $m_j \in \mathcal{M}$. While the vehicle is in this network, it may request content transmissions from RSUs in a random way. Now say v_i requests a new transmission $c_k \in \mathcal{C}$ from m_i. m_i then starts requesting the content from the content provider to transmit c_k to v_i. If m_i calculates that v_i cannot complete this transmission within the dwelling time, then m_i sends the *proactive caching request* message to m_j to ask it to perform proactive caching on the remaining fragments f_r of c_k. Next, v_i hands over to a new RSU. If this new RSU happens to be m_j, then this is a correct prediction and the pre-cached content is hit. In this case, m_j satisfies the remainder of v_i's previous transmission by its cache instead of having to request that from the content provider, hence realizing seamless transmission and reducing network delay. Otherwise, the new RSU has to finish the remaining transmission through the content provider via the backhaul network. A transmission delay μ is thereby introduced via $\frac{f_r \times F_c}{\omega}$ where ω is the backhaul link rate. In either case, a *prediction feedback* message (positive reward or negative reward) is sent back to m_i via the Central Node (depicted in Figure 1) so that it can update its prediction policy.

IV. SYSTEM AND ALGORITHM DESIGN

The first focus of this section is to introduce the designed hybrid cMAB proactive caching system. Then the underlying dual-context cMAB and single-context cMAB prediction algorithms will be elaborated on in more detail. The section starts with a brief theoretical background of cMAB problems.

A. BACKGROUND OF CONTEXTUAL MULTI-ARMED BANDIT PROBLEM

The contextual multi-armed bandit (cMAB) problem is a useful extension of the general multi-armed bandit (MAB) problem which is a special instance of reinforcement learning. Different from a full RL problem where a learning agent may have multiple states associated with the environment (e.g., positions in a game) and may transfer from one state to another, it only has a single state in the MAB problem [9] (i.e., no state transitions). From this perspective, MAB is essentially identical to *stateless Q-Learning* [22] and can also be treated as a model-free reinforcement learning technique. Despite the additional context used in cMAB to assist the decision-making process, it shares many common features with the general MAB problem including single-state agent, action selection and update strategy, *exploration-exploitation* dilemma [23], etc. A well-known scenario of the MAB problem is where a gambler in a casino sits in front of a slot machine with multiple arms and tries to get payoffs by pulling one of them. The ultimate goal of the gambler is to achieve the highest cumulative rewards through learning the inherent reward pattern of each arm and gradually concentrating on the best lever. During this process, the gambler will face the exploration-exploitation dilemma: 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). cMAB under the gambling scenario can be thought of as if the gambler has been given a "clue" and this is used to learn the best action.

A cMAB problem can be formally given as a tuple: $\langle \mathcal{A}, \mathcal{S}, \mathcal{R} \rangle$, where $\mathcal{A} = \{a_1, a_2, ..., a_k\}$ is a set of k actions (i.e., arms), $\mathcal{S} = \{s_1, s_2, ..., s_j\}$ is a set of j contexts, and $\mathcal{R} = \{\theta_{1-1}, \theta_{2-1}, ..., \theta_{j-k}\}$ associates action a_k and context s_j with its reward probability distribution defined by θ_{j-k}. This is formally formulated as follows:

- *Contextual multi-armed bandit* Consider a cMAB problem $\langle \mathcal{A}, \mathcal{S}, \mathcal{R} \rangle$. The aim of any agent in the cMAB problem is to learn a policy that maps contexts to actions, that is, $\pi(a \in \mathcal{A} \mid s \in \mathcal{S})$. Another viewpoint is that they now become multiple independent MAB tasks associated with contexts, and the agent aims to learn the best policy under these various contexts. Every time an agent is assigned a MAB task (possibly with a certain probability), it will observe context, take the action by looking at the current context, and eventually learn the best action. The agent takes an action a_k from its action set $\mathcal{A}$ under context $s_j \in \mathcal{S}$ and this will generate a success (reward 1) or failure (reward 0). The action $a_k \in \mathcal{A}$ produces a success with probability $\theta_{j-k} \in \mathcal{R}$. In other words, for an action a_k reward $r = 1$ is produced with probability θ_{j-k} and $r = 0$ with probability $1 - \theta_{j-k}$. In this case, θ_{j-k} can be seen as the expected reward of taking action a_k at situation s_j and is unknown to the agent. We can denote the estimated value of θ_{j-k} at time step t as $Q_t(a_k \mid s_j) = \frac{\text{sum of rewards when } a_k \text{ is taken under } s_j \text{ prior to t}}{\text{total number of times } a_i \text{ is taken under } s_j \text{ prior to t}}$. The cumulative rewards are now to be maximized across $\mathcal{S}$ over a certain amount of time T.

Generally, the agent can do better in cMAB than in a non-contextual MAB with the assistance of context that distinguishes one bandit problem from another [7], [9]. 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 [23], etc. The purpose of this paper is not to find out a sophisticated way to balance exploration and exploitation so the most straightforward ϵ-greedy is adopted. Despite the fact that cMAB involves learning policies, it still resembles the general MAB tasks, as the action taken only affects the immediate reward, and makes no difference to the next situations, as well as their rewards. Therefore, it is an intermediate between the MAB problem and the full RL problem.

B. HYBRID CMAB PROACTIVE CACHING SYSTEM

The topic of this subsection is to introduce the design of the proposed *Hybrid cMAB Proactive Caching System* (HCPC) used for proactive caching. The basic concept behind the hybrid system is that it implements a switching mechanism that allows an RSU to adaptively finalize its prediction between two cMAB-based prediction algorithms: ***Single-context cMAB*** and ***Dual-context cMAB*** algorithms.

In general, the agents in cMAB problems use context to help choose which action to play in the current iteration. The context observed is actually an *N-dimensional* context, where each dimension is a source of side information that may or may not be the same type. Therefore, single-context cMAB is a *one-dimensional* cMAB problem where the agent only observes one source of information (e.g., previous RSU) to consider as context. The agent in dual-context cMAB, on the other hand, is able to detect information from two sources (e.g., previous RSU and vehicle ID), together forming a *two-dimensional* context.

The single-context cMAB algorithm which makes use of the *previous RSU* has been exploited in [7] and excellent prediction performance was achieved. As one of the two underlying prediction algorithms in the HCPC system, it is enhanced in this paper with a *Win-or-Learn-Fast* variable learning rate to increase the robustness of the algorithm. In contrast to dual-context cMAB, the advantage of single-context cMAB is that it has sufficient learning opportunities for every related context s in the early stage of learning, but in some situations, it may suffer from a similar dilemma as in the non-contextual MAB problem as described in [7], hence the limitation in prediction performance. On the other hand, the dual-context cMAB designed in this paper utilizes two-dimensional context which consists of *vehicle ID* and *previous RSU*. It reinforces the single-context cMAB and could result in a more explicit context for an agent RSU to distinguish different tasks. Nevertheless, its disadvantage is the shortage of learning samples in the early stages, since a vehicle passes through an RSU from a particular prior RSU only once a day. Therefore, the motivation behind the HCPC system is to combine the advantages of both in order to ensure the accuracy of the prediction as much as possible. The designed switching mechanism is the enabler of adaptive selection between single-context and dual-context, depending on the comparison of their historical prediction performance. In the meantime, it guarantees a lower bound on its prediction performance, i.e., single-context cMAB.

A complete procedure of an RSU predicting the next RSU as the proactive caching node in the HCPC system starts when a vehicular user connects to the RSU. It makes two predictions (performs two action selections) with dual-context and single-context cMAB algorithms, respectively, denoted as P_D and P_S. It then performs the switching mechanism to finalize its decision $P_F \in \{P_D, P_S\}$ and sends its proactive caching request to the predicted RSU (i.e., P_F). In other words, the final decision can also be seen as the result of either dual-context cMAB or single-context cMAB.

The key point in the switching mechanism is the way to compare the historical prediction accuracy of the two cMAB algorithms. One thing to consider in the comparison is whether the RSU extracts its past predictions made for all the vehicles that have connected to it or just the prediction data of the current vehicle, which corresponds to ***RSU-Centric*** and ***Vehicle-Centric***, respectively. In the HCPC RSU-Centric system, the RSU finalizes its prediction (P_D or P_S) for all of the connecting vehicles, once it computes which cMAB algorithm may benefit its overall prediction performance in the current simulation cycle. On the other hand, the RSU in the HCPC Vehicle-Centric system does this on a vehicle level. It uses the past prediction performance of this particular vehicle to compute and determine what is the best option for the vehicle in the current cycle. The advantage of the Vehicle-Centric system is that it allows “customization" for different vehicular users, which will intuitively benefit individual users because the best decision is customized for them. The two systems use different *window sizes* (WS) for backtracking length to calculate past prediction performance because for the Vehicle-Centric system, to obtain a similar past prediction sample size it needs longer backtracking length i.e., larger WS than RSU-Centric system. We summarize the switching mechanism of HCPC system in **Algorithm 1** and meanwhile, a comprehensive flow of the system in the flowchart is shown in Figure 2.

Algorithm 1: Switching mechanism in Hybrid cMAB Proactive Caching System

```
while not the end of the test do
    if Vehicle $V_u$ connects to RSU $m$ then
        Predictions by parallel algorithms:
        $P_D \leftarrow$ Dual-context cMAB;
        $P_S \leftarrow$ Single-context cMAB;
        Finalize prediction $P_F$ - switching scheme:
        Vehicle-Centric System: Extract past
        predictions of $V_u$ made by RSU $m$ in the last
        $WS$ tests;
        RSU-Centric System: Extract past predictions
        of all vehicles made by RSU $m$ in the last $WS$
        tests;
        Compute cumulative average accuracy:
        $Acc_D \leftarrow$ Dual-context cMAB;
        $Acc_S \leftarrow$ Single-context cMAB;
        if $Acc_D > Acc_S$ then
            $P_F \leftarrow P_D$;
        else
            $P_F \leftarrow P_S$ ;
        end
    end
end
```

In a proactive caching-enabled vehicular network, the objective is to realize seamless content delivery to vehicular users. This is achieved by a high cache hit ratio which relies on accurate mobility prediction. Therefore, achieving high prediction accuracy is the objective of the hybrid cMAB proactive caching system. In the following, the detailed implementation and design of the two parallel cMAB prediction algorithms will be discussed.

C. TWO PARALLEL CMAB-BASED MOBILITY PREDICTION ALGORITHMS

Finding the best RSU to pre-cache relevant content for a vehicular user is a matter of mobility prediction. It is crucial that the currently associated RSU is able to predict the next possible RSU the vehicle is about to access, as accurately as possible. As discussed earlier, a cMAB problem is composed of action set, context set, and rewards. By taking appropriate actions, the agent hopes to maximize its payoff eventually. In the next RSU proactive caching problem, the currently connected RSU helps a vehicle to continue the unfinished content transmission immediately when it reconnects to a new RSU, provided that the new RSU has the requested content. This completely depends on whether the last RSU predicts or selects the correct RSU from its neighboring RSUs. If it was a correct prediction, positive feedback is given; otherwise, negative feedback is generated. From this point of view, they resemble each other in terms of action (RSU) selection and reward (feedback) generation. How the mobility prediction is modeled as a single-context cMAB problem has been elaborated on in [7]. However, the proposed dual-context cMAB algorithm differs in terms of the dimension of context. The remainder of this subsection will focus on the composition of the context in the dual-context cMAB in contrast to the single-context cMAB, and introduces how to solve them with the variable learning rate proposed in this paper.

1) **Context in cMAB** In cMAB problems, a specific Q-table that consists of multiple actions' quality values (Q-value) is associated with specific context $s \in \mathcal{S}$. The agent aims to learn a Q-table of s. Generally, the purpose of introducing context is to help the agent make better decisions compared to a general MAB problem (i.e., non-contextual MAB). The effectiveness of single-context cMAB with the previous RSU as the context has been proved in [7] since this information is a useful source to help RSUs distinguish the incoming directions of vehicles. Despite its excellent performance, there may still exist occasions where the RSU's actions have close Q-values, which results in high uncertainty and limits the prediction accuracy. Therefore, it is meaningful to investigate the performance of cMAB with additional context from a different dimension and this motivates the proposal of the Dual-context cMAB-based algorithm.

   Specifically, the context in the ***Dual-context cMAB-based Mobility Prediction*** algorithm combines two-dimensional context i.e., *vehicle ID* and *previous RSU*. As in single-context cMAB, the information of previous RSUs is easily accessed and used as a reference to such directions compared to other sorts of information e.g., road information, vehicle angle, etc. Moreover, the use of vehicle IDs, sometimes referred to as OBU IDs in literature e.g., [15], as additional contextual information is also legitimate as the IDs are important and useful identifiers in the next-generation vehicular networks. In

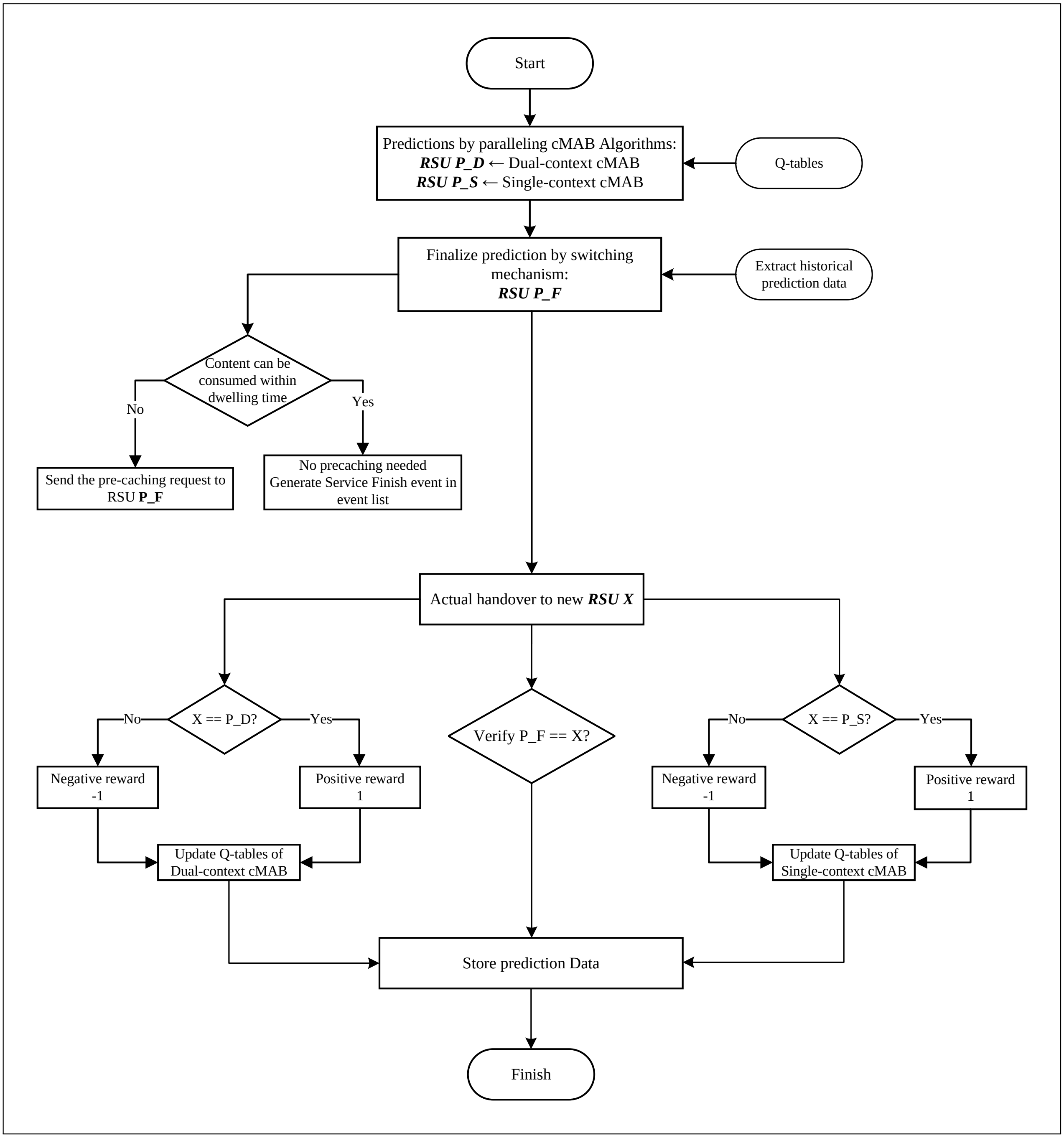

Start
Predictions by paralleling cMAB Algorithms:
RSU P_D ← Dual-context cMAB
RSU P_S ← Single-context cMAB
Q-tables
Finalize prediction by switching mechanism:
RSU P_F
Extract historical prediction data
Content can be consumed within dwelling time
No
Yes
Send the pre-caching request to RSU P_F
No precaching needed
Generate Service Finish event in event list
Actual handover to new RSU X
X == P_D?
No
Yes
Verify P_F == X?
X == P_S?
No
Yes
Negative reward
-1
Positive reward
1
Negative reward
-1
Positive reward
1
Update Q-tables of Dual-context cMAB
Update Q-tables of Single-context cMAB
Store prediction Data
Finish

FIGURE 2: Flowchart of the Hybrid cMAB Proactive Caching System: this is a general cycle of an agent RSU serving a connecting vehicle, from *Start* when a vehicle connects to the RSU, to *Finish* when its action-value table is successfully updated with corresponding rewards and relevant prediction data is stored sufficiently.

both algorithms, when the agent RSU needs to predict the next RSU (action selection) for a newly connected vehicle, the vehicle's relevant context will first be identified, which corresponds respectively to vehicle ID plus previous RSU as dual context or previous RSU only as single context. The task of the agent RSU is to learn the action values associated with the identified context through trial and error. This enables the agent RSU to solve separate bandit tasks associated with them, thereby guaranteeing a more effective policy learned. Since the dual-context cMAB solution is tailored to a specific vehicle, in principle it is likely to provide a more accurate prediction than single-context cMAB. The context of dual-context cMAB can be described as follows:

- Given an RSU $m \in \mathcal{M} = \{m_1, m_2, ..., m_M\}$ and a connecting vehicle with ID $v \in \mathcal{V} = \{v_1, v_2, ..., v_L\}$, the RSU m can detect vehicle v's previous connected RSU n which should be one of the neighbors of RSU m, i.e., $n \in \mathcal{N} = \{n_1, n_2, ..., n_N\}$. Combined with the already known vehicle ID v, the dual-context s can be identified as $s \rightarrow \langle v, n \rangle$

- With the identified context s, the RSU m can retrieve the Q-table associated with s so that an action can be predicted properly according to the action selection algorithm. If there does not exist such Q-table, it will initialize one for the combined context s and perform its decision.

2) **Mobility prediction** Mobility prediction (i.e., next RSU prediction) in the modeled cMAB-based prediction algorithms is essentially an action decision for an agent RSU. Action selection plays an important role in solving cMAB problems and is fundamentally based on the estimated true values of actions. In a cMAB problem, the learning agent learns its actions' quality values corresponding to a type of context through trial and error. We use $Q(a \mid s)$ to denote this value and name it Q*-value* as in Q-learning [10] [22], where $a \in \mathcal{A}$ and $s \in \mathcal{S}$. The agent then uses the corresponding exploration-exploitation scheme (i.e., ϵ-greedy) to select the appropriate action based on their Q-values: the best action is selected with a probability of $1 - \epsilon$; Otherwise, with small probability ϵ, actions will be selected randomly with equal probability regardless of their Q-values.

$$A = \begin{cases} \arg\max_a Q(a \mid s), & 1 - \epsilon \\ \quad random, & \quad\epsilon \end{cases} \tag{1}$$

3) Q**-value update** For economy and clarity, we use the simplified term $Q(a)$ of $Q(a \mid s)$ to denote the Q-values of the actions under context s. In [7], we have derived the recursive action-value updating formula using *incremental implementation* [9]:

$$Q_{n+1} = Q_n + \frac{1}{n}(r_n - Q_n) \tag{2}$$

where Q_{n+1} is the value after the action a has been selected for n times.

Equation (2) is further generalized as follows by replacing the so-called *step-size* $\frac{1}{n}$ with a constant learning rate α. This is because vehicular networks are dynamic environments with varying traffic densities, which results in a *nonstationary* bandit problem. Therefore, recent rewards should be given more weight when updating action values.

$$Q(a) \leftarrow (1 - \alpha)Q(a) + \alpha r \tag{3}$$

The Q-values of actions under a particular context $s \in \mathcal{S}$ are hence updated according to Equation (3).

The agent RSU accepts a reward after taking an action and observing its relevant outcome. The outcome is translated to a reward through the reward function R. In other words, given an action a taken at time step t and the observed outcome as b (which may or may not occur immediately), its reward can be computed with $r_t = R(b)$. In the cMAB-modeled mobility prediction problem, the outcome of an agent RSU predicting one of its neighboring RSU as the next possible RSU is either $b = True$ or $b = False$. In order to introduce punishment for a wrong prediction and inspired by the reward function used in the Dynamic Spectrum Access problem in [10], the reward function R adopted by this work is:

$$r = R(b) = \begin{cases} 1, & b = \text{True} \\ -1, & b = \text{False} \end{cases} \tag{4}$$

4) **Win-Or-Learn-Fast variable learning rate** The learning rate α is a key parameter for any RL problems including cMAB. It has a significant influence on the dynamics of the learning process. A fixed learning rate for both the positive outcome and the negative outcome is often seen in the literature such as [24] and [25]. Bowling and Veloso proposed Win-Or-Learn-Fast (WoLF) method in [26] and provided the method to adapt different learning rates when different outcomes are observed. The principle behind this method is that the authors stated that the learning agent should learn faster when it is losing and more slowly when winning. This principle of learning faster when unsuccessful or “cautiously" when successful is also relevant in dynamic vehicular environments, e.g., when a change in network topology or traffic distribution requires the RSUs to readjust their learned policies. Besides, this feature of WoLF also encourages exploration in the early stage of learning and is important in terms of avoiding rapid convergence towards a local optimum at the beginning of the learning process.

Therefore, a straightforward adaption of WoLF is to split the value of the learning rate α in Equation (3) into two cases, α_{win} and α_{lose}: the Q-value is updated with α_{win} if $r = 1$ and α_{lose} if $r = -1$. Therefore, the Equation (3) is rewritten using separate terms for Q-value estimates before $(Q(a))$ and after the update $(Q^{'}(a))$ as follows:

$$Q^{'}(a) = \begin{cases} (1 - \alpha_{win})Q(a) + \alpha_{win}, & r = 1 \\ (1 - \alpha_{lose})Q(a) - \alpha_{lose}, & r = -1 \end{cases} \tag{5}$$

Again, $Q^{'}(a)$ is still a simplified term of $Q^{'}(a \mid s)$ that omits the context s. The learning agent RSU updates Q-values of its actions for each independent context s using Equation (5).

As mentioned earlier, the single-context cMAB adopted in [7] is enhanced in this paper to accommodate the WoLF. To sum up, the two underlying parallel cMAB-based prediction algorithms in HCPC Vehicle-Centric system are summarized in **Algorithm 2**. They are referred to as dual-context cMAB and single-context cMAB, respectively.

Algorithm 2: cMAB-based Next RSU selection Algorithm

```
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 A vehicle connects to RSU $m$ then
        Context detection:
        Dual-context cMAB:
        1. Detect context $s_1 \leftarrow$ previous RSU before
          $m$;
        2. Detect context $s_2 \leftarrow$ Vehicle ID ;
        3. Dual context $s_D \leftarrow s_1 + s_2$ ;
        Single-context cMAB:
        Single context $s_S \leftarrow$ previous RSU before $m$
        if $s_*$ ($s_D$ or $s_S$) is a new detection then
            Create an entry of $s_*$ to its action values;
            Initialize $Q(a \mid s_*) = 0, \forall a \in \mathcal{A}_m$;
        end
        Predict the next RSU $a_*$ ($a_D$ or $a_S$) by:
        $(a_D \mid s_D) \leftarrow$ action taken based on Eq. (1);
        $(a_S \mid s_S) \leftarrow$ action taken based on Eq. (1);
    end
    if Handover happens then
        Reward $r_*$ ($r_D$ or $r_S$) generation:
        $r_D \leftarrow$ observe the reward of $a_D$ according to
          Eq. (4);
        $r_S \leftarrow$ observe the reward of $a_S$ according to
          Eq. (4);
        Update $Q$-tables of RSU $m$ with $r_D$ and $r_S$
          for Dual-context cMAB and Single-context
          cMAB by Eq. (5):
        if $r_*$ is 1 then
            $Q(a_* \mid s_*) \leftarrow (1 - \alpha_{win})Q(a_* \mid$
              $s_*) + \alpha_{win}$
        end
        if $r_*$ is -1 then
            $Q(a_* \mid s_*) \leftarrow (1 - \alpha_{lose})Q(a_* \mid$
              $s_*) - \alpha_{lose}$
        end
    end
end
```

V. SIMULATION AND PERFORMANCE EVALUATION

A. SIMULATION SETUP

1) Test Scenarios

Three vehicular test scenarios are designed in this paper to simulate realistic traffic scenarios and the corresponding test data is generated by Simulation of Urban MObility (SUMO) [27]. They are summarized as the following:

- **Scenario I - Commuting traffic**:

  This scenario aims to simulate daily commuters in reality. Normally, such commuting vehicles depart and arrive from one area in a city to another. We focus on two urban areas, Las Vegas as the primary city and Manchester as the secondary city to generalize the application of the proposed HCPC Vehicle-Centric system on two cities with two very different road layouts. 5 traffic zones (TAZs) are defined in SUMO to simulate realistic residential and workplace areas (assuming that a TAZ contains both areas) and each two of them form a TAZ pair, which results in 20 TAZ pairs. 10 vehicles commute between a TAZ pair, resulting in 200 vehicles in total. Figure 3a and Figure 3b show the distribution of the TAZs and RSUs in two cities.

  Another feature of commuting traffic is that commuters generally follow a point-to-point daily routine. Thus, to approximate this pattern, a specific vehicle traveling between two TAZs departs from a specific road in the originating TAZ as its home address and arrives at a specific road in the terminating TAZ as its workplace address, which is referred to as a "*departure trip*" and, conversely, as a "*return trip*". A "*departure test trace*" and a "*return test trace*" consist of 200 departure trips (i.e., vehicles) and 200 return trips, respectively. Furthermore, an individual vehicle is associated with an ID (ranging from 0 to 199 in this case) and its ID remains unchanged throughout all the test traces which reinforces the fact that they are commuters. Figure 3c and Figure 3d show an example of routes of all commuting vehicles in the two cities.

- **Scenario II - Random traffic**:

  This scenario is an extremely random scenario where vehicles randomly depart and arrive at locations on the map, independent of TAZs, but still follow the shortest path. Additionally, vehicle IDs in one test trace are different from those in another test trace (i.e., no duplicated IDs exist). This scenario may not be totally realistic but is meaningful to assess the performance of the proposed proactive caching system under such extreme circumstances. For consistency, there are also 200 random trips in each test trace of this scenario. Figure 3e and Figure 3f show an example of this scenario in the two cities.

- **Scenario III - Mixed traffic**:

  In reality, it is very likely that the daily traffic in an urban area is mixed. In other words, it is composed of both commuting traffic and random traffic. The former is the commuters and the latter is generally new and random traffic going through the area. Therefore, the purpose of Scenario III is to simulate this more realistic scenario and is a mixture of Scenario I and II. For simplicity, traffic is mixed with an equal percentage of 50%, which results in two groups of vehicles: 200 commuting vehicles and 200 random vehicles, in each test trace of Scenario III. In addition to the mentioned traffic features in Scenario I and II, this test scenario also differentiates the two vehicle groups by their IDs (i.e., random vehicles do not use IDs ranging from 0 - 199). An example of this scenario can be referred to as the combination of Figure 3c and 3e or Figure 3d and 3f.

2) Traffic simulation

Each of the above scenarios has 200 test traces including departure test traces and return test traces. These 200 test traces are organized in the order of *departure-return-...-departure-return* during simulation for simulating a complete workday in an urban area, though this is not important for Scenario II which simulates completely random traffic.

On the other hand, 200 test traces also aim to simulate 200 workdays and the simulation period in SUMO is between 8 am to 9 am for departure trips and 5 pm to 6 pm for return trips. The vehicles' routes 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 The Following Model is used to set the maximum safe speed in the sense of being able to stop in time to avoid a collision. Other road behaviors apply as well such as lane changing, acceleration/deceleration, intersections, etc. Technical details about these settings can be found in SUMO documentation[1].

3) Network simulation

Discrete event-driven system simulation [7], [13], [28] is a common simulation method to use in wireless networks including vehicular networks. It enables simulation to perform through a series of events. Such discrete events are generated from SUMO test traces as described earlier, which include *departure* and *arrival* of vehicles, *content request*, *handover*, and *finishing of content consumption*. A complete cycle of the simulation is 200 test traces, each of which is technically a workday. As this is an online learning process, the RSUs make predictions as they learn throughout the simulation cycle and become increasingly knowledgeable as the simulation runs. In addition, Table 1 summarizes the important parameters used in traffic simulation and network simulation.

TABLE 1: Simulation Parameters

Parameter	Description	Value
α_{win}	WoLF learning rate when winning	0.05
α_{lose}	WoLF learning rate when losing	0.5
ϵ	ϵ-greedy exploration-exploitation	0.05
N	No. of test traces	200
V_C	No. of Commuting Vehicles	200
V_R	No. of Random Vehicles	200
T_S	SUMO Simulation Time	1 hour
$\mathcal{M}$	No. of RSUs	32 (Las Vegas) 30 (Manchester)
ω	Backhaul Link Rate	$5Gbps$
e	Transmission rate	$50Mbps$
K	Size of content database	30
F_c	Fragment size	$100MB$

B. PERFORMANCE EVALUATION

Five proactive caching systems are studied to evaluate their prediction performance:

- *HCPC Vehicle-Centric System*: The vehicle-centric variant of the hybrid cMAB system. It implements the switching mechanism at the vehicle level. The window size WS chosen for extracting the historical prediction data is 20 in order to obtain sufficient past prediction samples.
- *HCPC RSU-Centric System*: The RSU-centric variant of the hybrid cMAB system. Different from HCPC Vehicle-Centric system, it focuses the switching mechanism at the RSU level. The window size WS chosen for extracting the historical prediction data is 3, because it is sufficient to obtain a similar sample size with $WS = 20$ in the Vehicle-Centric system.
- *Previous-RSU cMAB-based Proactive Caching System*: This is the system that only uses the previous RSU as the context in cMAB. Its superiority has been tested and verified in the work [7]. In this paper, the WoLF variable learning rate is further implemented in order to maintain consistency with the HCPC system.
- *CPT+ based Proactive Caching System*: This system is based on the sequence prediction algorithm Compact Prediction Tree+ (CPT+). Different from the work [13], we have adjusted the algorithm to be used in an online mode. Briefly, an RSU trains its prediction tree model with all the 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). To some extent, CPT+ also makes use of "context".
- *PPM based Proactive Caching System*: This system implements the first-order Prediction by Partial Matching (PPM). It is a broadly used technique for context modeling and prediction as in [16]. Again, we have adjusted this technique to exploit online learning.

Remark: For clarity, the above five systems are referred to and denoted in the following figures as: *HCPC Vehicle-Centric*, *HCPC RSU-Centric*, *PrevRSU-cMAB*, *CPT+* and *PPM*, respectively.

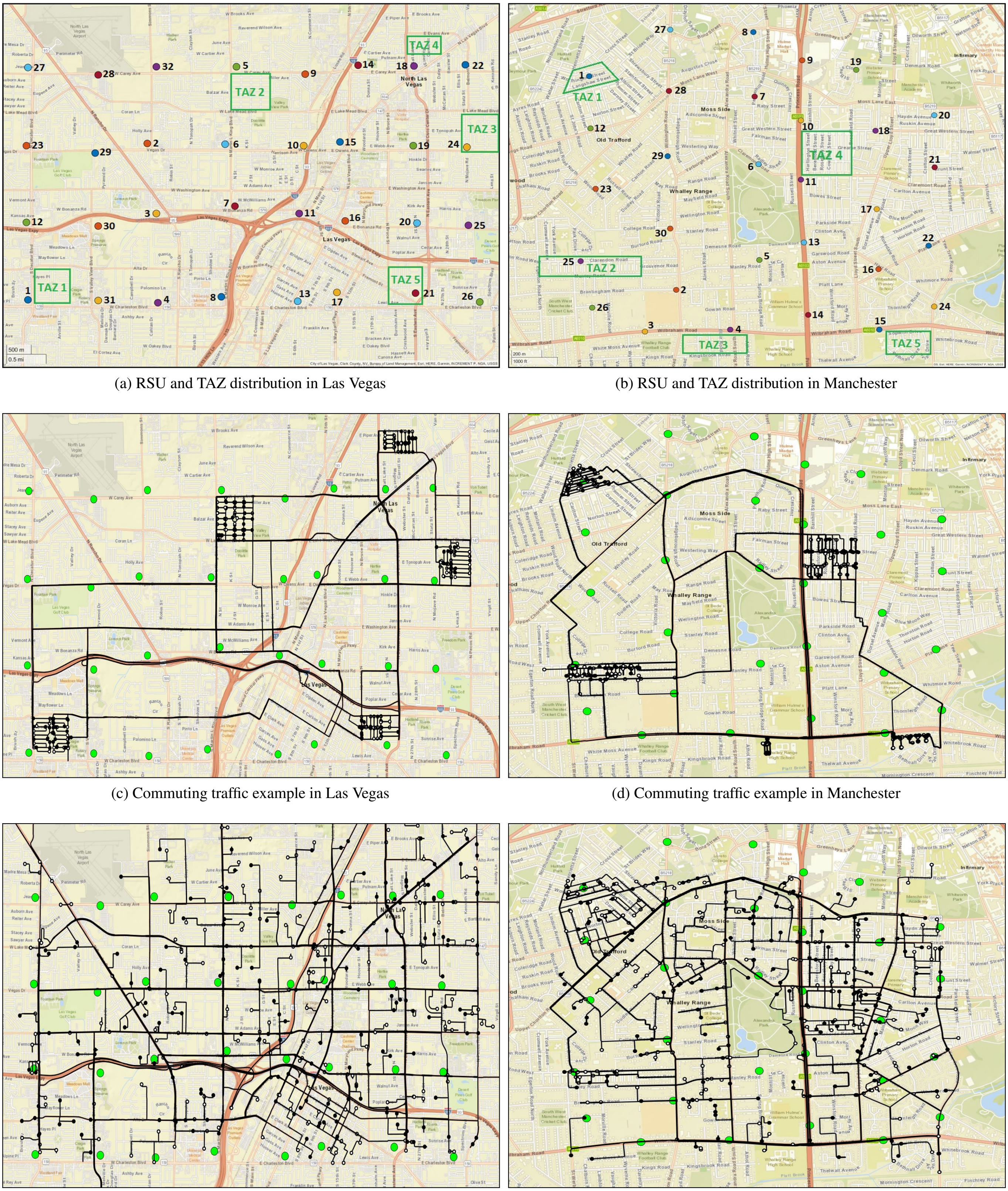
1) Evaluation metrics

The performance of proactive caching system is assessed with *cache hit ratio*. For these systems, cache hit ratio completely depends on how accurately a learning RSU can predict or select the correct next RSU. In other words, a selected action is considered correct if and only if it matches the actual RSU that a vehicle transits to. Therefore, we define the following metrics for system evaluation:

- *Cumulative Prediction Accuracy with Sliding Window*: Denoting the total number of predictions as $Q_i^{prediction}$ and correct ones as $Q_i^{correct}$ of particular test trace $i \in N$. A fixed sliding window sw is applied to the cumulative accuracy. Thus, prediction accuracy PA_n up till test trace $n \in N$ is defined as:

$$PA_n = \begin{cases} \frac{\sum_{i=1}^{n} Q_i^{correct}}{\sum_{i=1}^{n} Q_i^{prediction}}, & n \le sw \\ \frac{\sum_{i=n-sw+1}^{n} Q_i^{correct}}{\sum_{i=n-sw+1}^{n} Q_i^{prediction}}, & n > sw \end{cases}$$

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



(a) RSU and TAZ distribution in Las Vegas

(b) RSU and TAZ distribution in Manchester

(c) Commuting traffic example in Las Vegas

(d) Commuting traffic example in Manchester

(e) Random traffic example in Las Vegas

(f) Random traffic example in Manchester

FIGURE 3: Vehicle routes of different test scenarios in two urban areas: for clarity, the positions of RSUs in (c) - (f) are shown in green dots. Their labels can be mapped to (a) and (b), respectively.

2) Simulation results

We treat Las Vegas as our primary city for simulation. Therefore, all three scenarios have been tested with the traffic data of Las Vegas. As the purpose of using Manchester city is to show the generalization of the proposed system to different road layouts, only the most detailed Scenario III is included to achieve this. In the following, we demonstrate and analyze these results on a scenario basis.

A) Scenario I - Commuting traffic

Figure 4 demonstrates the prediction performance of the five proactive caching systems under Commuting traffic scenario in Las Vegas. As the traffic pattern of this scenario focuses on purely commuting traffic, their routes should be predictable. The accuracy of the two HCPC systems that reach nearly 95% after convergence further validates this. The lost 5% accuracy results from ϵ-greedy exploration algorithm where 0.05 is adopted. The significant superiority of HCPC systems benefits from the switching mechanism which guarantees the best accurate action to be taken. It is obvious that the prediction accuracy of both HCPC systems does not show a clear difference and again, this is due to 1) the nature of the commuting traffic pattern in this scenario and 2) the introduction of vehicle ID in the dual-context cMAB algorithm. After a certain period of learning (approximately 20 test traces as depicted in Figure 4), overall the RSUs in both HCPC Vehicle-Centric and HCPC RSU-Centric tend to finalize their decisions with the prediction of dual-context cMAB.

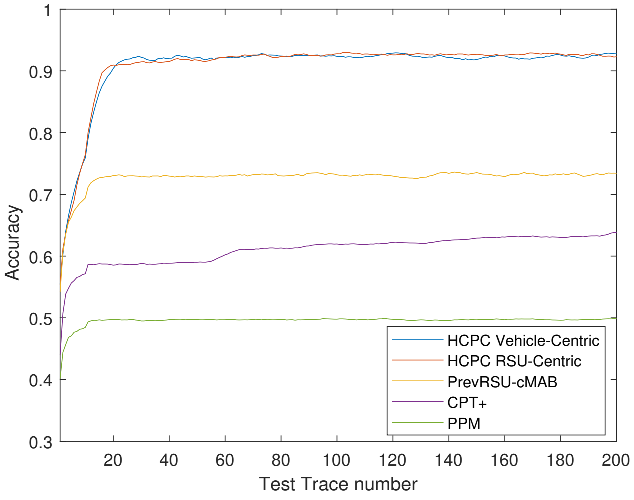


FIGURE 4: Cumulative prediction accuracy of the five proactive caching systems in Las Vegas - Commuting Traffic Scenario

They outperform the PrevRSU-cMAB system by 20% and nearly 30% over the CPT+ system despite the fact that it is experiencing a slow-growing trend as the CPT+ model gets increasingly mature with more data being used to establish its model. With this trend, we could infer CPT+ may reach a similar level of performance as HCPC systems perhaps after 1000 more test traces. Nevertheless, this is also its limitation in terms of adaptability and flexibility. The first-order PPM system performs the worst because essentially it is the same to the baseline *Probability-based Proactive Caching System* investigated in [7] and therefore cannot break the intrinsic limit of a certain scenario.

B) Scenario II - Random traffic

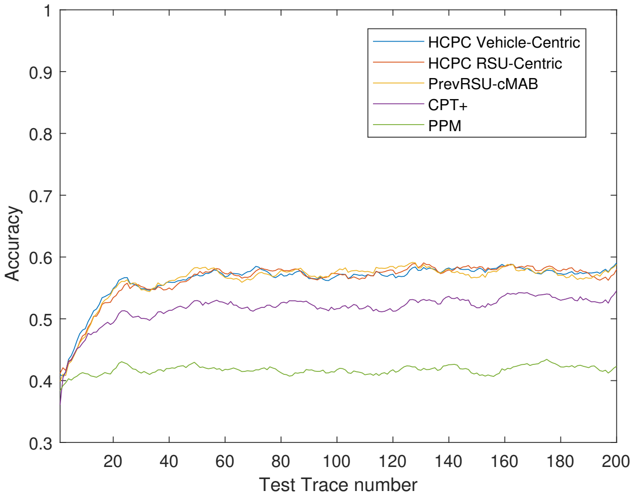


FIGURE 5: Cumulative prediction accuracy of the five proactive caching systems in Las Vegas - Random Traffic Scenario

The performance of the systems under extreme Random traffic in Las Vegas depicted in Figure 5 shows obvious degradation, especially for cMAB-based systems. Recall the traffic pattern in this is extremely random including both randomnesses in routes and vehicle IDs. Due to this nature, the HCPC systems always finalize their predictions with single-context cMAB because the accuracy of dual-context cMAB is constantly outperformed by single-context cMAB. This makes both systems identical to the PrevRSU-cMAB system that uses previous RSU only as context. Despite this, they still outperform CPT+ and PPM-based ones. Such randomness in this scenario is also reflected in the oscillations of the result curves, unlike a much more smooth curve as in the purely commuting scenario.

C) Scenario III - Mixed traffic

Prediction performance of the proactive caching systems in Las Vegas and Manchester under the mixed scenario is shown in Figure 6 and Figure 7 respectively. HCPC Vehicle-Centric system outperforms the other four systems and shows a similar performance of nearly 80% accuracy in both cities. Therefore, the proposed HCPC Vehicle-Centric system can be generalized and applicable in various urban areas.

Compared to Commuting traffic and Random traffic

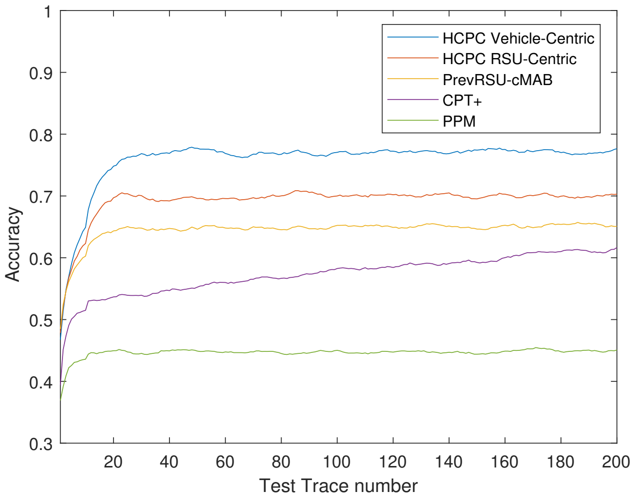


FIGURE 6: Cumulative prediction accuracy of the five proactive caching systems in Las Vegas - Mixed Traffic Scenario

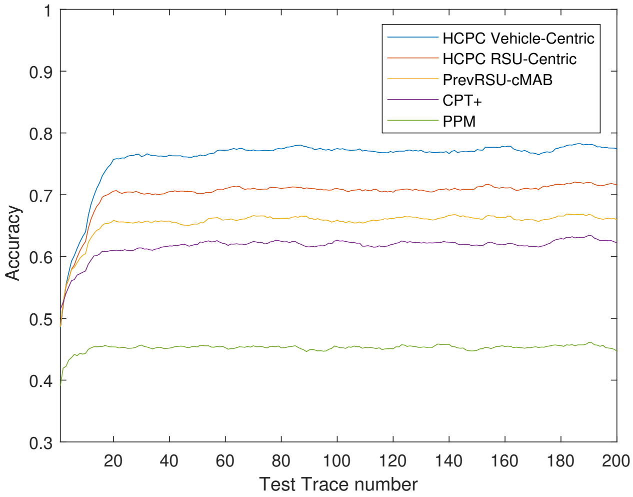


FIGURE 7: Cumulative prediction accuracy of the five proactive caching systems in Manchester - Mixed Traffic Scenario

in Scenario I and II, its accuracy falls in between. One reason for this is because of the co-existence of both commuting traffic and random traffic. On the other hand, it is in this relatively more realistic scenario that the proposed HCPC Vehicle-Centric system shows its superiority over its counterpart HCPC RSU-Centric system that has 70% of overall prediction accuracy. Thanks to its vehicle-centric feature, the most possible prediction is always made for an individual vehicular user (most likely a commuter vehicle) independent from other users. However, an RSU in the HCPC RSU-Centric system may make a less accurate prediction for a vehicle due to its RSU-centric feature. For instance, a vehicular user may benefit if the RSU finalizes its prediction for this user with dual-context cMAB but for historical reasons, the RSU still believes the prediction of single-context cMAB can benefit most of the users connecting to it. This is when inaccurate predictions are made. In contrast, the HCPC Vehicle-Centric system avoids such situations by guaranteeing that the finalized prediction is vehicle-specific. To further validate this argument, Figure 8 demonstrates the prediction accuracy of all the commuting vehicles in the two HCPC systems in Las Vegas and Manchester. For Las Vegas, the cumulative accuracy of these vehicles in the HCPC Vehicle-Centric system is the same as in the pure commuting scenario and is not affected by the random traffic, but they experience degradation in the HCPC RSU-Centric system. Although not shown, this is also a valid argument in the purely commuting traffic in Manchester.

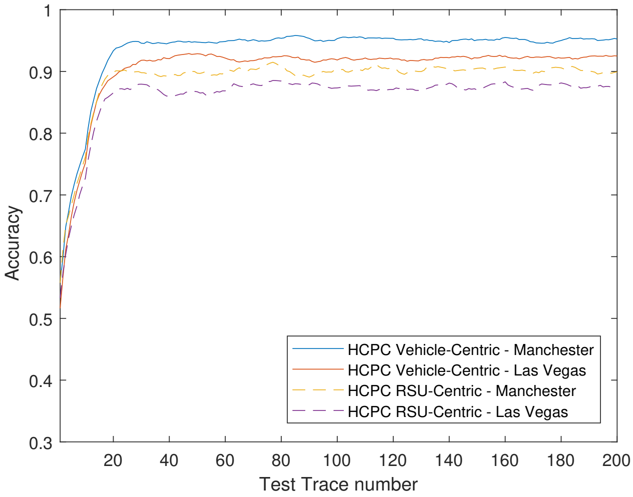


FIGURE 8: Prediction accuracy of the Vehicle-Centric and RSU-Centric hybrid systems showing only commuting vehicles in two cities - Mixed Traffic Scenario

VI. EXTENDED STUDY ON AN ALTERNATIVE COMMUTING TRAFFIC SCENARIO

This section aims to provide insights into situations that may impact the accuracy of dual-context cMAB, through analysis of individual vehicles and RSUs in a special commuting traffic scenario which is an intermediate between Scenario I and Scenario II in Las Vegas in Section V. In fact, it is identical to the scenario in [7], except that [7] did not consider return trips of vehicles. In addition to showing the general prediction performance of the proactive caching systems, there will be a comprehensive comparison to the point-to-point commuting traffic scenario in Section V. By analyzing the unfavorable factors that limit the performance of dual-context cMAB, this section also aims to conclude the common limitations of MAB-based algorithms.

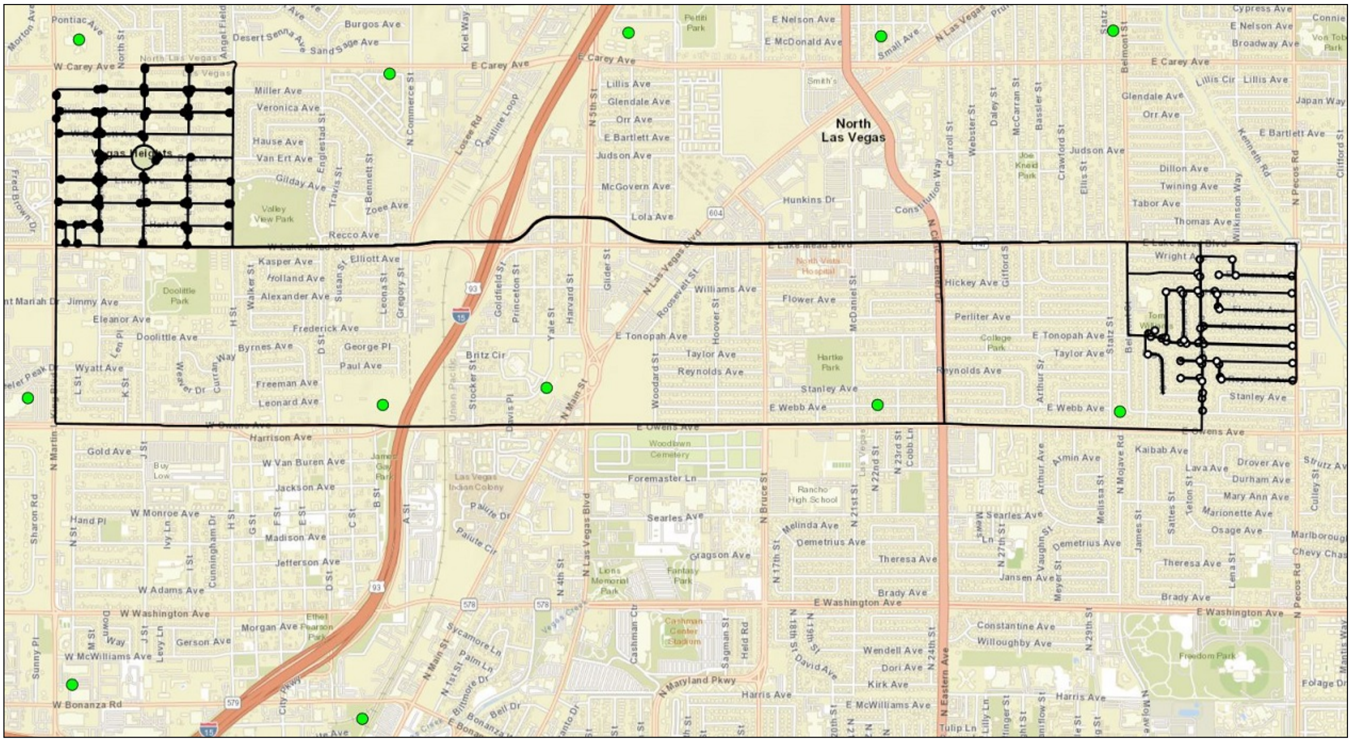


FIGURE 9: An illustration of a vehicle's departure routes in Scenario IV - Commuting traffic with random Origin-Destination (OD). The figure shows all the 100 departure routes (overlapped) of a vehicle, where all hollow circles indicate the starting points and all solid circles indicate the ending points. All the starting or ending points are located in their own TAZ, which means that the vehicle follows its daily routine from one TAZ to another but varies in location.

The following is a detailed description of this scenario:

- **Scenario IV - Commuting traffic with random Origin-Destination (OD)** This is a special variant of Scenario I in Section V. The only difference is that commuters in this scenario do not follow a fixed point-to-point daily routine. Instead, they may depart and arrive at random locations within the departing and arriving TAZs. Therefore, it is still called a commuting scenario and may exist in reality where people do not own fixed parking places and park anywhere nearby. Figure 9 shows a concrete example of this scenario.

As depicted in Figure 10, while both HCPC systems still outperform other proactive caching systems, they experience a degradation in accuracy compared to Scenario I - Commuting traffic. This is mainly because of the randomness in origins and destinations within TAZs. To provide more insight into this, RSU 10 is selected for further analysis. Note that only its performance in the Vehicle-Centric system is analyzed here. As shown in Figure 3(a), RSU 10 has four actions: $\{6, 9, 11, 15\}$, and it is very close to TAZ 2. However, its overall prediction accuracy in Scenario I - Commuting traffic and Scenario IV - Commuting traffic with random OD shows a disparity in Figure 11(a). RSU 10 only predicts around 75% accurately in Scenario IV in contrast to 95% accuracy in Scenario I.

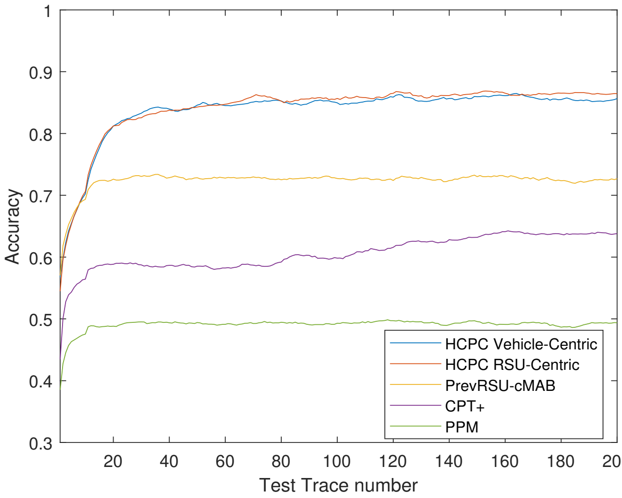
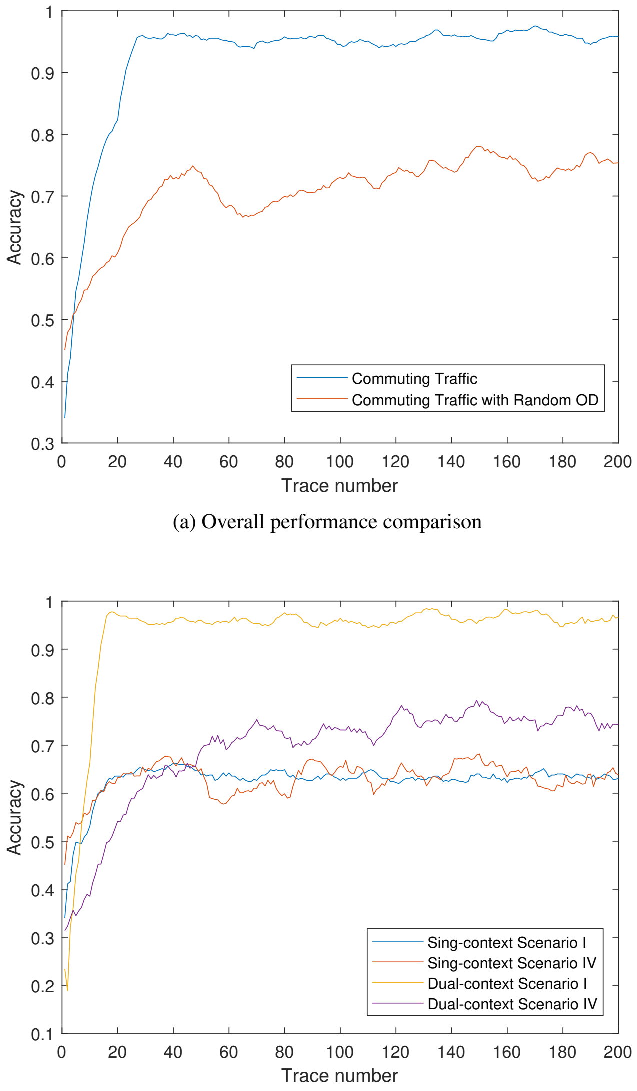


FIGURE 10: Overall prediction accuracy comparison in Scenario IV - Commuting traffic with random Origin-Destination (OD)

Figure 11(b) further disaggregates its overall performance into the separate performance of the two underlying cMAB algorithms. It is obvious that in both scenarios, dual-context cMAB dominates the performance at some point during the simulation, though this happens much later in Scenario IV



(a) Overall performance comparison

(b) Performance of the two underlying algorithms in two scenarios

FIGURE 11: Prediction performance comparison of RSU 10 in two commuting traffic scenarios in Las Vegas: Scenario I - Commuting Traffic vs Scenario IV - Commuting traffic with random OD.

than in Scenario I. Despite the notable oscillations of single-context cMAB in random OD scenario, the performance difference of single-context does not seem to be significant (both around 62%). Given the final overall accuracy, the gain brought by dual-context cMAB is considerable.

However, for some vehicles that RSU 10 predicts for in Scenario IV, dual-context cMAB does not work accurately and is even outperformed by single-context cMAB. Therefore, the problem now becomes what causes such a remarkable degradation of dual-context cMAB in the two scenarios. Take vehicle 90 as an example and consider the last 30 test traces, i.e., from trace 171 to 200. The followings are some observations based on the analysis of the data of vehicle 90:

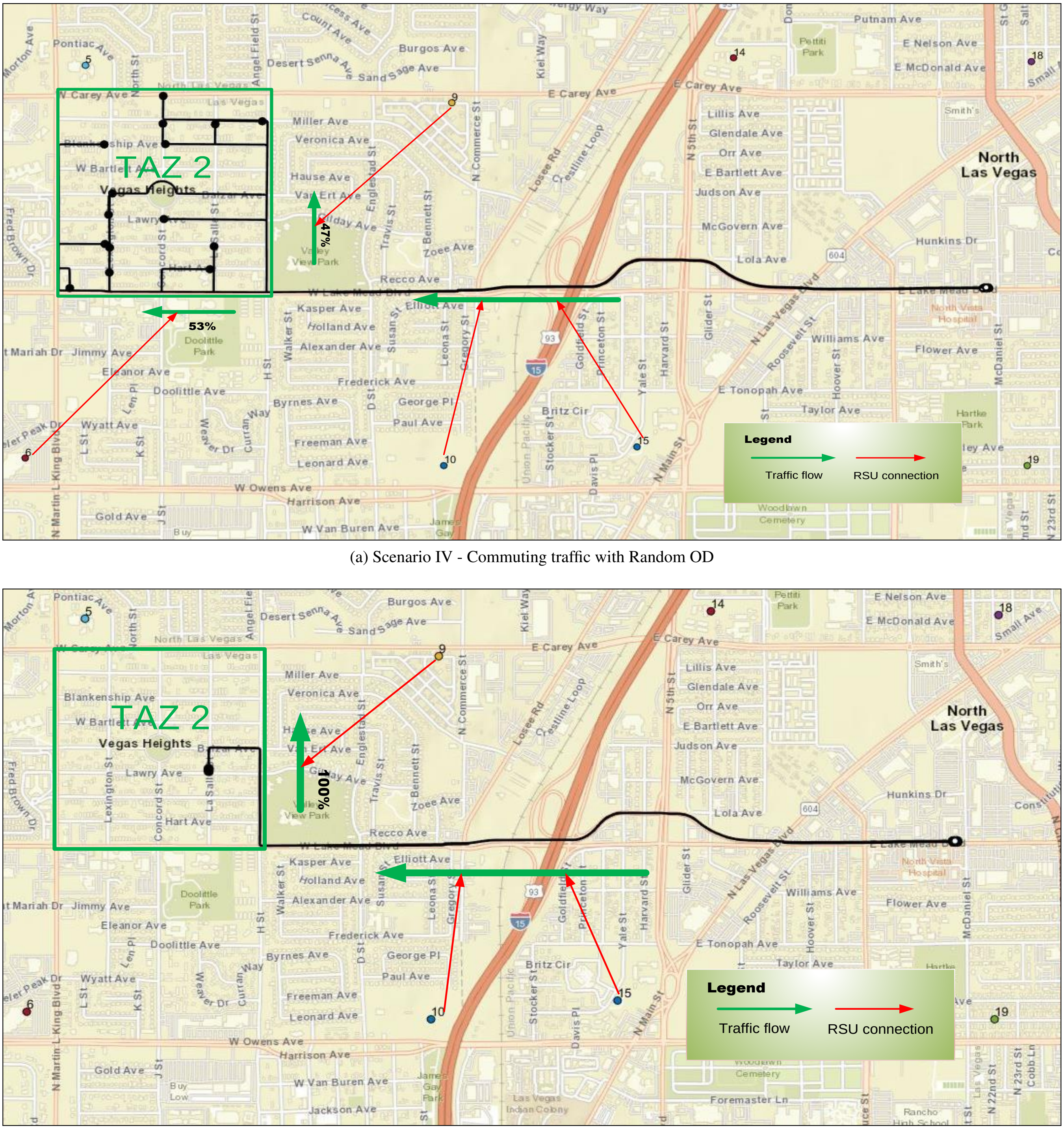
- Prediction accuracy of the last 30 traces is 50%
- Dual-context combinations, (Vehicle ID, Previous RSU), used by RSU 10 to make a prediction for vehicle 90 are: (90, 6), (90, 9), and (90, 15)
- Basically, all the wrong predictions happened in vehicle 90's departure trips, from TAZ 3 to TAZ 2 (referred to Figure 3(a)), under context (90, 15)
- The prediction accuracy under context (90, 15) is only 6.67%

As mentioned earlier, the main contributor to this inaccuracy is the randomness in the arrival TAZ, TAZ 2 in this case. Figure 12 illustrates some partial departure routes of vehicle 90 before it arrives TAZ 2. As shown in Figure 12a, vehicle 90 connects to RSU 6 or RSU 9 after RSU 10 because its destination is somewhere in TAZ 2. The proportions of such transitions to RSU 6 and RSU 9 in the last 30 test traces are 53% vs 47%, respectively. Consequently, the Q-values of context (90, 15) of RSU 10 end up converging to $\langle -0.9980, -0.9965, -0.9980, -0.9980\rangle$. This means that RSU 10 believes that no convincing action exists and it is very easy to make inaccurate predictions with Q-values like these. In contrast, such a situation is rare in Scenario I as shown in Figure 12b, because it simulates point-to-point traffic and such randomness in TAZs is minimized. As a result, Q-values of $\langle -0.5000, 0.9927, -0.5000, -0.500\rangle$ of context (90, 15) is achieved at the end of the simulation, which means that the second action i.e., RSU 9 is a convincing action to take to achieve accurate prediction.

To sum up, the above situation where Q-values are all negative or even close to -1 may happen in any MAB-based algorithms including dual-context cMAB, single-context cMAB as well as non-contextual MAB as studied in [7]. Every dimension of context introduced is to help reduce the uncertainty of the agent RSU about its actions. Therefore, to resolve the above dilemma, the agent RSU may need further information on top of dual context, e.g., the lane in which the vehicle is currently positioned. This will be discussed in future works.

VII. CONCLUSION

This paper addressed the problem of proactive caching at the next RSU with a *Hybrid cMAB Proactive Caching System* that exploits two parallel underlying cMAB-based prediction algorithms: *Dual-context cMAB* and *Single-context cMAB*. The system allows RSUs to adaptively finalize their predictions between two algorithms. The hybrid system is further developed into two variants, *Vehicle-Centric* System and *RSU-Centric* System, and their prediction performance is evaluated by comparing with three other systems, namely Previous-RSU cMAB, CPT+, and PPM, under three realistic-like traffic scenarios in two urban areas of Las Vegas, USA and Manchester, UK. Simulation results have shown the excellent performance of the proposed hybrid proactive caching system. It has reached approximately 93% prediction accuracy under the Commuting traffic scenario and the Hybrid



(a) Scenario IV - Commuting traffic with Random OD

(b) Scenario I - Commuting Traffic (point to point)

FIGURE 12: Partial Departure routes of vehicle 90 in Las Vegas

Vehicle-Centric System, in particular, still reaches nearly 80% accuracy in the Mixed traffic scenario while keeping the excellent prediction performance for commuting vehicles the same as in the Commuting traffic scenario. In addition, an extended study was conducted to provide discussion on and insight into the potential limitation on the performance of MAB learning systems. The results of the two cities demonstrate its superiority over the other three proactive caching systems, as well as its adaptability and applicability to different test scenarios and road layouts.

REFERENCES

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

[2] P. Dai, K. Liu, X. Wu, Y. Liao, V. C. S. Lee, and S. H. Son, "Bandwidth efficiency and service adaptiveness oriented data dissemination in heterogeneous vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 6585–6598, 2018.

[3] Z. Su, Y. Hui, T. H. Luan, Q. Liu, and R. Xing, *The Next Generation Vehicular Networks, Modeling, Algorithm and Applications*. Springer, 2021.

[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 of Things Journal*, vol. 6, no. 3, pp. 4512–4520, 2018.

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

[6] 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 Global Communications Conf. (GLOBECOM)*, Dec. 2018, pp. 1–6.

[7] Q. Wang and D. Grace, "Proactive edge caching in vehicular networks: An online bandit learning approach," *IEEE Access*, vol. 10, pp. 131 246–131 263, 2022.

[8] A. Mahajan and D. Teneketzis, "Multi-armed bandit problems," in *Foundations and applications of sensor management*. Springer, 2008, pp. 121–151. [Online]. Available: https://doi.org/10.1007/978-0-387-49819-5_6

[9] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[10] N. Morozs, T. Clarke, and D. Grace, "Distributed heuristically accelerated q-learning for robust cognitive spectrum management in lte cellular systems," *IEEE Transactions on Mobile Computing*, vol. 15, no. 4, pp. 817–825, 2016.

[11] N. B. Hassine, P. Minet, D. Marinca, and D. Barth, "Popularity prediction–based caching in content delivery networks," *Annals of Telecommunications*, vol. 74, no. 5, pp. 351–364, 2019.

[12] 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. 27 699–27 719, 2019.

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

[14] 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: Springer International Publishing, 2015, pp. 625–636.

[15] 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 Journal on Selected Areas in Communications*, vol. 36, no. 8, pp. 1786–1801, 2018.

[16] 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 Transactions on Vehicular Technology*, vol. 67, no. 6, pp. 5435–5444, June 2018.

[17] C. J. C. H. Watkins, "Learning from delayed rewards," 1989.

[18] D. Bouneffouf and I. Rish, "A survey on practical applications of multi-armed and contextual bandits," 2019.

[19] 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 Transactions on Vehicular Technology*, vol. 69, no. 7, pp. 7821–7834, 2020.

[20] Y. Miao, Y. Hao, M. Chen, H. Gharavi, and K. Hwang, "Intelligent task caching in edge cloud via bandit learning," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 1, pp. 625–637, 2020.

[21] X. Xu, M. Tao, and C. Shen, "Collaborative multi-agent multi-armed bandit learning for small-cell caching," *IEEE Transactions on Wireless Communications*, vol. 19, no. 4, pp. 2570–2585, 2020.

[22] C. Claus and C. Boutilier, "The dynamics of reinforcement learning in cooperative multiagent systems," *AAAI/IAAI*, vol. 1998, no. 746-752, p. 2, 1998.

[23] D. Russo, B. V. Roy, A. Kazerouni, and I. Osband, "A tutorial on thompson sampling," *CoRR*, vol. abs/1707.02038, 2017. [Online]. Available: http://arxiv.org/abs/1707.02038

[24] M. Bennis and D. Niyato, "A q-learning based approach to interference avoidance in self-organized femtocell networks," in *2010 IEEE Globecom Workshops*. IEEE, 2010, pp. 706–710.

[25] A. H. Ko, R. Sabourin, and F. Gagnon, "Performance of distributed multi-agent multi-state reinforcement spectrum management using different exploration schemes," *Expert systems with applications*, vol. 40, no. 10, pp. 4115–4126, 2013.

[26] M. Bowling and M. Veloso, "Multiagent learning using a variable learning rate," *Artificial Intelligence*, vol. 136, no. 2, pp. 215–250, 2002.

[27] 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 *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018. [Online]. Available: <https://elib.dlr.de/124092/

[28] G. A. Wainer and P. J. Mosterman, *Discrete-event modeling and simulation: theory and applications*. CRC press, 2018.

QIAO WANG (S'20) received his PhD from University of York in 2022, with the subject being 'Mobility-prediction based proactive caching in vehicular networks', and his Master's degree from Tampere University of Technology (now Tampere University) in 2014. Since 2015, he has been working as an LTE System Developer with Ericsson. He is currently working as a postdoctoral researcher in the iTwins lab at University of York and his research interests include intelligent open RAN, machine learning and federated learning in O-RAN, and vehicular networks.

DAVID GRACE (S'95-A'99-M'00-SM'13) received his PhD from University of York in 1999, with the subject of his thesis being 'Distributed Dynamic Channel Assignment for the Wireless Environment'. Since 1994 he has been a member of the Department of Electronic Engineering at York, where he is now Professor (Research), Head of Communication Technologies Research Group, and Director of the Centre for High Altitude Platform Applications. 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 currently a lead investigator on H2020 MCSA SPOTLIGHT, UK Government funded MANY, dealing with 5G trials in rural areas, and HiQ investigating Quantum Key Distribution from high altitude platforms. He was technical lead on the 14-partner FP6 CAPANINA project that dealt with broadband communications from high altitude platforms. He is an author of over 280 papers, and author/editor of 2 books. He is the former chair of IEEE Technical Committee on Cognitive Networks for the period 2013/4. He is a founding member of the IEEE Technical Committee on Green Communications and Computing. From 2014-8 he was a non-executive director of Stratospheric Platforms Ltd. In 2000, he jointly founded SkyLARC Technologies Ltd, and was one of its directors.