2020

Improving Traffic Safety and Efficiency by Adaptive Signal Control Systems Based on Deep Reinforcement Learning

Yaobang Gong
University of Central Florida

IMPROVING TRAFFIC SAFETY AND EFFICIENCY BY ADAPTIVE SIGNAL CONTROL BASED ON DEEP REINFORCEMENT LEARNING

by

YAOBANG GONG
B.S. Central South University, 2016
M.S. University of Central Florida, 2018

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Civil, Environmental and Construction Engineering
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Spring Term
2020

Major Professor: Mohamed Abdel-Aty

ABSTRACT

As one of the most important Active Traffic Management strategies, Adaptive Traffic Signal Control (ATSC) helps improve traffic operation of signalized arterials and urban roads by adjusting the signal timing to accommodate real-time traffic conditions. Recently, with the rapid development of artificial intelligence, many researchers have employed deep reinforcement learning (DRL) algorithms to develop ATSCs. However, most of them are not practice-ready. The reasons are two-fold: first, they are not developed based on real-world traffic dynamics and most of them require the complete information of the entire traffic system. Second, their impact on traffic safety is always a concern by researchers and practitioners but remains unclear. Aiming at making the DRL-based ATSC more implementable, existing traffic detection systems on arterials were reviewed and investigated to provide high-quality data feeds to ATSCs. Specifically, a machine-learning frameworks were developed to improve the quality of and pedestrian and bicyclist's count data. Then, to evaluate the effectiveness of DRL-based ATSC on the real-world traffic dynamics, a decentralized network-level ATSC using multi-agent DRL was developed and evaluated in a simulated real-world network. The evaluation results confirmed that the proposed ATSC outperforms the actuated traffic signals in the field in terms of travel time reduction. To address the potential safety issue of DRL based ATSC, an ATSC algorithm optimizing simultaneously both traffic efficiency and safety was proposed based on multi-objective DRL. The developed ATSC was tested in a simulated real-world intersection and it successfully improved traffic safety without deteriorating efficiency. In conclusion, the proposed ATSCs are capable of effectively controlling real-world traffic and benefiting both traffic efficiency and safety.

ACKNOWLEDGMENTS

First of all, I would like to express my sincere gratitude to my advisor Professor Mohamed Abdel-Aty, for the continuous guidance and support of my master’s study and other research, for his patience, motivation and immense knowledge. I could not have imagined having a better advisor!

Secondly, I would like to appreciate the rest of my respectable committee members: Dr. Samiul Hasan, Dr. Qipeng Zheng and Dr. Qing Cai, for their invaluable comments.

I would like to thank my friends and my colleagues. I would also like to acknowledge Florida Department of Transportation for funding this research and providing data.

Finally, I would like to thank my parents. You gave me bravery to pursue my dream here, ten thousand miles away from home.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS/ABBREVIATIONS

3DQN	Double Dueling Deep Q network
AASE	Average Absolute Speed Error
AI	Artificial Intelligence
API	Application Programming Interface
ATM	Active Traffic Management
ATSC	Adaptive Traffic Signal Control
ATSC-MORL	ATSC Controller developed using a Multi-Objective RL algorithm
ATSC-SORL	ATSC Controller developed using a Single-Objective RL algorithm
ATSPM	Automated Traffic Signal Performance Measures
BDS	Bluetooth Detection System
BLE	Bluetooth Low Energy
BMS	Bluetooth MAC address scanner
BMS-BLE	BLE based Bluetooth MAC address scanner
BMS-O	Ordinary Bluetooth MAC address scanner
CAV	Connected and Automated Vehicles
CNN	Convolutional Neural Network
CV	Connected Vehicle
DDPG	Deep Deterministic Policy Gradient
DNN	Deep Neural Network

DQN	Deep Q Network
DRL	Deep Reinforcement Learning
GPS	Global Positioning System
MAC	Media Access Control
MAD	Median Absolute Deviation
MARL	Multi-Agent Reinforcement Learning
MORL	Multi-Objective Reinforcement Learning
MVDS	Microwave Vehicle Detection System
O/D	Origin-Destination
OUATS	Orlando Urban Area Transportation Study
RHODES	Real-time Hierarchical Optimizing Distributed Effective System
RL	Reinforcement Learning
RSSI	Received Signal Strength Indicator
SCATS	Sydney Coordinated Adaptive Traffic System
SCOOT	Split Cycle Offset Optimization Technique
SEB	Speed Error Bias
SVM	Support Vector Machine
TD-Learning	Temporal Difference Learning
TOD	Time of Day
UCF	University of Central Florida
V2I	Vehicle-to-Infrastructure

CHAPTER 1: INTRODUCTION

1.1 Background

Growing economy and population result in increasing travel demand that often goes beyond the capacity of the current traffic system. This leads to inevitable traffic congestion. Rather than building more roadways that might incur even more demand, a cost-effective approach to address this issue is improving the efficiency of traffic management, such as traffic signal control. Even frequently re-timed, the traditional pre-timed signal controllers, whose timing are determined by historical traffic information, might not be able to handle the dynamic traffic demand. To overcome these limitations, many adaptive traffic signal control systems (ATSC) were developed, such as Split Cycle Offset Optimization Technique (SCOOT), Sydney Coordinated Adaptive Traffic System (SCATS) and Real-time Hierarchical Optimizing Distributed Effective System (RHODES). Those ATSCs use traffic detectors to acquire current traffic flow information, especially the turning movement counts, then feed them into models to forecast near-future traffic flow profiles and finally adjust the signal timing according to the prediction. These systems were seen as successful in multiple deployments around the world over the years.

The aforementioned ATSCs are fully based on “human-crafted” traffic flow features such as traffic volume, queue length, delay or travel time; “human-crafted” traffic flow models; and “human-crafted” signal control elements such as cycle length, offset, splits, and etc. Admittedly, all those “human-crafted” features are valuable human knowledge. Yet the complex, discrete and heterogeneous demand and behavior of the vehicles might be overlooked by the model-based decision making and aggregated input data. Especially with the rapid development of connected vehicle (CV) technology, vehicles could provide their real-time information to signal controllers via vehicle-to-infrastructure (V2I) communication. Therefore, the vehicle-based data will be largely available in the foreseeable future. To fully utilize the high-dimensional-high-complexity information, many researchers proposed ATSCs based on one of the artificial intelligence (AI) algorithms, reinforcement learning (RL), to let the "machines" learn how to control traffic signals ((Samah El-Tantawy et al., 2014). Although, most of the traffic flow models and most of the signal control parameters are no longer needed in such ATSCs, unfortunately, due to the limited computational power of conventional RL algorithms, most of the studies still need to aggregate the discrete traffic information in some degree to generate the input of the RL algorithms.

Recently, due to the rapid development of deep learning, the so-called deep reinforcement learning (DRL) algorithms incorporated with deep neural networks (DNNs) shows its ability to handle high-dimensional-high-complexity-disaggregated input data. Hence, ATSCs based on DRL are able to get rid of "human-crafted" traffic information and high-resolution traffic data such as the position, speed or even the origin and destination of individual vehicles could directly be used. Nevertheless, with the help of better hardware and the improved computational power of DRL, ATSCs based on DRL have the ability to achieve the centralized control of the roadway network that conventional-RL-based ATSCs have never accomplished. The flexibility of the objective definition in DRL algorithm also enlightens the possibility to develop a multi-objective ATSCs, which could pro-actively improve the traffic operation and safety at the same time.

Several exploratory studies in the last three years (L. Li et al., 2016a; Lin et al., 2018; Van Der Pol & Oliehoek, 2016) shows the bright future of ATSCs based on DRL. However, most of the existing DRL-based ATSCs are not practice-ready. The reasons are two-fold. First, most of them did not prove their capability in controlling real-world traffic. They were developed and evaluated in simplified traffic networks with hypothetical traffic demands. Moreover, most of them require the complete information of the entire traffic system, such as the location, and/or the speed, and/or even the origin-destinations of all the vehicles within the network. This is not feasible in practice. Therefore, a DRL-based ATSC that is developed based on real-world traffic dynamics is desired. As the ATSC should use data from traffic detection systems and/or connected and automated vehicles (CAV) as input rather than assuming the complete information is known, necessary knowledge about traffic data is also desired.

Second, the impact of DRL-based ATSCs on traffic safety is a concern by researchers and practitioners but remains unclear (Wei et al., 2019). As the DRL algorithms provide a flexible definition of its objectives, an ideal solution might be developing an ATSC that is able to simultaneously optimize traffic safety and efficiency.

1.2 Research Objectives

In order to fill the research gap mentioned in the previous section, the primary research objective of this dissertation is making DRL-based ATSC more implementable. The detailed objectives are listed as follows:

- Understanding the types of traffic data on arterials that can be utilized as the data feeds of DRL-based ATSC
- Developing algorithms to improve existing traffic data collection systems that could be used as the data feed of ATSC
- Developing a network-level ATSC algorithm based on decentralized Multi-Agent Reinforcement Learning (MARL) based on limited traffic information and real-world traffic dynamics
- Conducting an exploratory study on a multi-objective ATSC aiming at optimizing traffic safety and efficiency simultaneously in real-time

The first objective has been achieved in Chapter 3 by the following sub-tasks:

a) Reviewing traffic data collection systems used by the state-of-the-practice ATSCs or other types of traffic control systems

b) Identifying other data collection systems that could serve as the potential data feeds of ATSCs

The seconds objective has been achieved in Chapter 4 by the following sub-tasks:

c) Assessing the feasibility of using Bluetooth Low Energy (BLE) to obtain traffic counts in terms of the detection rate and range of BLE scanners.

d) Developing an algorithm based on BLE for estimating the counts of pedestrians and bicyclists

The third objective has been achieved in Chapter 5 by the following sub-tasks:

e) Formulating the traffic control problem into a multi-agent deep reinforcement learning setting

f) Developing the microscopic traffic simulation environment based on real-world traffic data

g) Training the DRL-based ATSC algorithm in the simulation and evaluate its performance by the emulated field traffic control system

The fourth objective has been achieved in Chapter 6 by the following sub-tasks:

h) Formulating the traffic control problem into a multi-objective deep reinforcement learning setting

i) Estimate the real-time crash model based on the historical traffic and crash data

j) Developing the microscopic traffic simulation environment based on real-world traffic data

k) Training the Multi-objective-reinforcement-learning-based ATSC algorithm in the simulation and evaluate its performance by the emulated field traffic control system and DRL-based ATSC developed in Chapter 5

l) Analyzing the control policies of the developed safety-oriented ATSC

1.3 Dissertation Structure

The rest of the thesis is organized as follow: Chapter 2 provides a comprehensive review of state-of-art RL algorithms and early studies developing ATSCs based on DRL; Chapter 3 reviews and evaluates traffic data collection systems that can serve as the data feeds of ATSCs;

Chapter 4 elaborates two machine learning frameworks that aim at improving vehicular and non-motorized traffic data; Chapter 5 presents an exploratory study of a network-level signal control algorithm using deep reinforcement learning in a decentralized approach; Chapter 6 introduces a multi-objective ATSC that is able to improve traffic efficiency and safety simultaneously. Finally, Chapter 7 summarizes the overall dissertation and presents the implications of the work.

CHAPTER 2: LITERATURE REVIEW

2.1 Deep Reinforcement Learning

Deep Reinforcement Learning is a family of Reinforcement learning (RL) algorithms incorporated with Deep Neural Networks (DNNs). RL (Sutton & Barto, 2018) is a goal-oriented machine learning algorithm. It learns to achieve a complex *goal* over many discrete steps by interacting with the environment.

In the context of a control problem, in every discrete control step, an RL control *agent* (e.g. signal controller) iteratively observes the *state s* of the environment (e.g. roadway network), takes an *action a* (e.g. directly change the signal phase or set the green duration of the signal phase) accordingly based on its underlying behavior *policy* π, receives a feedback reinforce *reward r* (e.g. waiting time, delay or travel time) for the action taken, which will be accumulated to its long-run *goal* (minimizing delay, decreasing travel time or minimizing stops), from the environment, and transits to the *next state* s' according to the environment dynamics and state *transition probability P.* The RL agent optimizes the *policy*, which is the mapping from the set of the all possible states *S* to the set of all possible actions *A*, by learning from the accumulated discounted long-term *reward*, with a *discount factor* γ, of applying different action sequences. During the learning process, it keeps adjusting its *policy* by maximizing the expectation of the long-term *reward* until it converges to the *optimal policy* π^*. Figure 1 gives an illustration of the setting of the reinforcement learning problem.

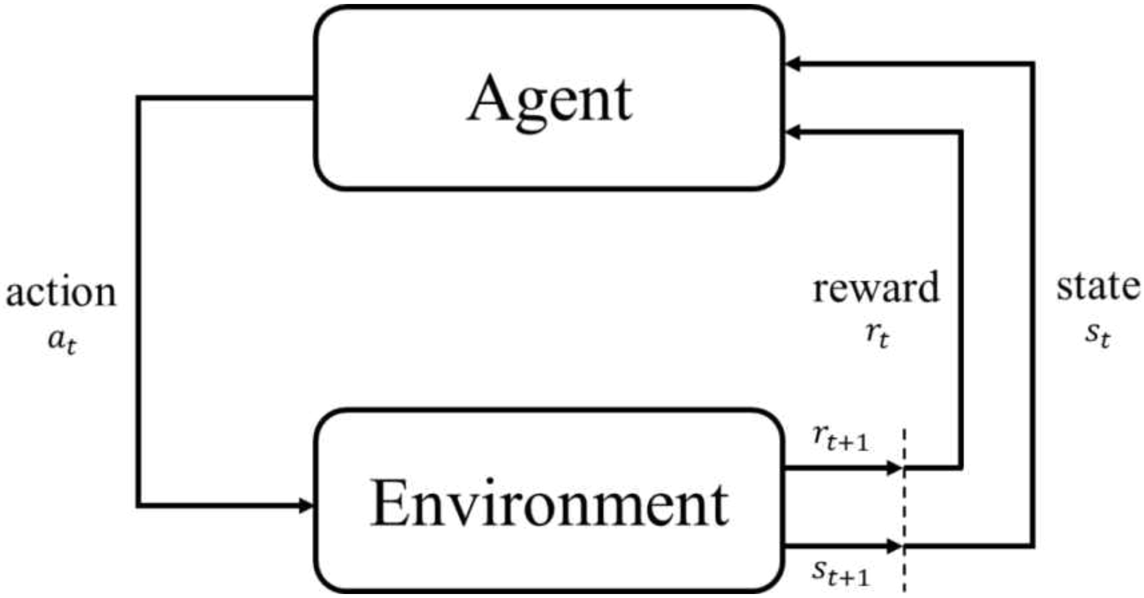


Figure 1 The Illustration of the Reinforcement Learning Problem Setting (Sutton & Barto, 2018)

The *value function* of the RL problem is the estimation of the long-term reward of each state or state-action pair. The state value is the expected long term discounted reward for following *policy π* from *state s*, which is defined as:

$$\boldsymbol{V_\pi(s) = E[R_t|s_t = s]} \tag{1}$$

and it decomposes into the Bellman equation:

$$\boldsymbol{V_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} P(s',r|s,a)[r + \gamma V_\pi(s')]} \tag{2}$$

The Q value, or action value, refers to the expected long-term discounted reward for selecting *action a* and in *state s* and then following the *policy π,* which is defined as:

$$\boldsymbol{Q_\pi(s,a) = E[R_t|s_t = s, a_t = a]} \tag{3}$$

and it also decomposes into the Bellman equation:

$$\boldsymbol{Q_\pi(s,a) = \sum_{s',r} P(s',r|s,a)[r + \gamma \sum_{a'} \pi(a'|s') Q_\pi(s',a')]} \tag{4}$$

If there are multiple objectives needs to be optimized, multi-objective reinforcement learning (MORL) should be used. In the context of MORL, multiple *goals* are optimized simultaneously. In MORL, each objective has its associated reward and value function. Thus, Q-values are expressed as Q vector $\boldsymbol{MQ_\pi(s,a)}$:

$$\boldsymbol{MQ_\pi(s,a) = [Q_\pi^1(s,\mathrm{a}), Q_\pi^2(s,\mathrm{a}), \ldots, Q_\pi^n(s,\mathrm{a})\ ]^T} \tag{5}$$

Intuitively, the optimal Q vector is defined as

$$\boldsymbol{MQ^*(s,a) = \max_p MQ_\pi(s,\mathrm{a})} \tag{6}$$

The “maximum operation” of a vector could have different definitions. Generally, there are two ways for MORLs handling the “maximum operation”: single-policy MORL approach and multi-policy MORL approach (C. Liu et al., 2015). Single policy approaches aim to find the best single policy representing the preferences or the trade-off among the objectives. Several different algorithms are developed to determine and express the preferences or trade-off, such as linear/non-linear weighted sum approach, W-learning, AHP approach, ranking approach, and geometric approach, etc. Multi-policy MORL aims at approximating the Pareto front by a set of policies. The Pareto front is a set of Pareto non-dominated solutions. If any objective of solution could not be improved without sacrificing at least one other objective, the solution is a Pareto non-dominated solution.

The most well-known algorithm family to learn the value functions is temporal difference (TD) learning. TD-learning and its extension TD(λ)-Learning (Sutton, 1988) learn the state value function $V_\pi(s)$, while SARSA (Singh & Sutton, 1996) and Q-Learning (Watkins & Dayan, 1992) learn the action value, or Q value. Among those algorithms, Q-Learning is the most common RL algorithm utilized in Adaptive Signal Control.

Q-learning is an off-policy algorithm, which means its policy being followed (the actioned chosen) is independent with its learning process. In Q-learning, the agent chooses the action $a \in A$ with the highest Q-value (greedy action) based on a matrix called Q-table. The Q-table is a mapping table of all discrete state value $s \in S$ to all discrete action value $a \in A$. At every discrete step, Q-learning improves its policy greedily. The adjusted Q-value is learned by

$$\boldsymbol{Q_\pi(s,a) = \sum_{s',r} P(s',r|s,a)[r + \gamma \sum_{a'} \pi(a'|s')Q_\pi(s',a')]} \quad (7)$$

where Q_k is the new Q-value after the adjustment at learning step *k*; Q_{k-1} is the current Q-value stored in the Q-table; *s, a, r* are current state, action, and reward at step k; *s', a'* are the next state and action; α is the learning rate controlling the adjusting size.

Conventional tabular form Q-learning requires a Q-table to store Q-values for all *(s, a)* pairs. However, when state set *S* (e.g. detailed representation of traffic states) is growing, the Q-table becomes extremely large, which makes the learning process intractable (the curse of dimensionality). As a consequence, function approximation $Q(s,a;\ \theta)$ (where θ is the hyper-parameters of the approximator) of the Q-table was introduced. Function approximation aims to generalize a "function" from examples to estimate the mapping. It is usually a concept of supervised learning. Linear function approximation is a popular choice before the age of deep learning. However, the neural network was utilized dated back a long time ago (Bertsekas & Tsitsiklis, 1996).

Since the Q-values are estimated by a function, the learning process is no longer directly updating Q-value but updating the hype-parameters θ. However, diverge and instability might occur when the neural network approximators are applied, especially for high-dimensional continuous state-action spaces (Tsitsiklis & Roy, 1997). It is because of the highly correlated consecutive states inputs and frequently changing policy due to slight changes in Q values.

Recently, Deep Q Network (DQN), one of the earliest DRL algorithm using deep neural networks (DNNs) as the functional approximator, shows its capability to deal with the large state-action space in RL problems (Mnih et al., 2015; Mousavi et al., 2018). Besides, it tackles the aforementioned instability and diverging issue by experience replay (LIN, 1993) and target network. Experience replay breaks the temporal correlation of states in the consecutive learning process. In every training step, the agent stores its 'experience' (s, a, r, s') into the experience replay memory θ and then randomly samples a minibatch from it to update the θ. In this way, the strong correlation between consecutive states is eliminated.

The target network also helps to mitigate the correlation issue. A target network is a DNN which has the same structure with the primary network (or value network) yet updates less frequently. It is used to generate target-Q-value to update the hype-parameters θ of the primary network:

$$\boldsymbol{J} = \sum_{\boldsymbol{s}} \boldsymbol{P}(\boldsymbol{s})(\boldsymbol{Q}_{\boldsymbol{target}}(\boldsymbol{s}, \boldsymbol{a};\ \boldsymbol{\theta}^{-}) - \boldsymbol{Q}(\boldsymbol{s}, \boldsymbol{a};\ \boldsymbol{\theta})) \tag{8}$$

where $P(s)$ denotes the probability of the state s in the minibatch; Q and Q_{target} are Q-values estimated by primary network and target network respectively. J is used as a loss function to update θ in backpropagation optimization. Every certain steps, the hyper-parameters of target network θ^{-} is updated by the hyper-parameters of primary network θ. As a result, the loss function of the current training step is evaluated by an earlier snapshot, decorrelate the target and primary Q-values/state and increase the stability of the learning algorithm.

Double DQN (van Hasselt et al., 2015) is an improved version of DQN. In standard DQN, the selection of the greedy action and the evaluation of the Q-value are using the same action value (Q-value), which might lead to overoptimistic value estimates. van Hasselt et al. (van Hasselt et al., 2015) proposed to evaluate the greedy action by the online network but to estimate its value by the target network. This is achieved by modifying the calculation of the target Q value:

$$\boldsymbol{Q_{target}(s,a) = r + \gamma Q\left(s', \mathbf{argmax}_{a'}(Q(\,s', a'; \theta)); \theta^{-}\right)} \tag{9}$$

Double DQN is proved to be able to find better policies than standard DQN.

To get a faster converge, Wang et al. (Z. Wang et al., 2015) proposed the dueling DQN. The dueling DQN estimate the state value function and associated advantage function and then confine them to get the Q-value:

$$\boldsymbol{Q(s,a) = V(s;\ \alpha) + [A(s,a;\ \beta) - \frac{1}{|A|}\sum_{a'} A(s,a';\beta)]} \tag{10}$$

where $V(s;\ \alpha)$ is value function which indicates the expected overall rewards of a specific state s; $A(,a;\ \beta)$is the state-dependent advantage of a specific action a over other actions and it is then normalized by the average value of all actions $a' \in A$. The performance of dueling DQN is shown better than the standard DQN especially when there exist several similar-valued actions.

However, one of the most serious problems of DQN algorithms is that it is hard for them to handle the case when the action set *A* (e.g. centralized network signal control) is extremely large or even continues. Since in continuous action spaces, finding the greedy policy requires optimization of *a* at each discrete timestamp. The optimization is too slow with large and unconstrained approximators like DNN (Lillicrap et al., 2015). There is another family of RL algorithms called policy-based methods is able to tackle this issue. Different from the value-based TD learning, policy-based methods optimized the function approximation of *policy* $\pi(a|s;\theta)$ directly. The hyper-parameter θ of the approximation function is updated by gradient ascent on the expected long-term reward. REINFORCE (Williams, 1992) is a popular policy-based algorithm using policy gradient. However, the conventional policy gradient is relatively more stable than Q-learning but less efficient. Thus, the actor-critic algorithm (Sutton & Barto, 2018) takes advantage of both policy-based and value-based algorithm. The "critic" learns the hyper-parameters of action-value function while the "actor" learns the hyper-parameters of policy function. Asynchronous advantage actor-critic (Mnih et al., 2016) is a recently developed popular actor-critic algorithm based on DNN.

Another popular DRL based on actor-critic is Deep Deterministic Policy Gradient (DDPG) (Lillicrap et al., 2015). For the continuous action space, the so-called Deterministic Policy Gradient comes up with a deterministic policy rather than a stochastic policy as conventional actor-critic algorithms, which is more efficient. Besides, with the help actor-critic structure, the DDPG avoids the optimization of *a* at each discrete timestamp. In addition, similar to DQN, DDPQ employs the experience reply and a “soft” updating target similar to the target network in DQN. The tests of the DDPG algorithm on some benchmark environments showed that DDPG can solve problems 20 times faster than DQN.

However, DDPG is still not the perfect solution to a continuous action space problem. There is a potential issue of all actor-critic-like algorithms is that learning rate of the actor and critic needs to be tuned very carefully to avoid inconsistent learning process. There are a few studies aiming at improving of DQN to make it suitable for continuous action space (Gu et al., 2016).

Deep reinforcement learning is a still active research domain. In terms of its application in the ATSC and ramp control, the DQN-like algorithms are suitable for ATSC of an isolated intersection and it could be applied decentralized network control. Whereas, actor-critic-like algorithms could be applied to adaptive ramp metering which might require a continuous ramp metering rate.

2.2 ATSC Based on Conventional Reinforcement Learning

The research use of RL as the back algorithm of ATSC dates back to the mid-1990s. As the best of the authors’ knowledge, Thorpe and Anderson (Thorpe & Anderson, 1996) firstly developed a traffic signal control algorithm using SARSA. Although its state, action and reward formulation are still crude, the evaluation shows that it outperforms the fixed timing plan by 29%

when it is implemented in a simulate 4×4 grid network. During the past two decades, a lot of researchers developed ATSCs based on different kinds of RL algorithms without the help of DNNs, which is classified as ATSC based on "conventional" reinforcement learning. A review of previous studies is given below.

2.2.1 Problem Formulation

Although, there are several studies developed and tested their ATSCs on an isolated intersection (Abdulhai et al., 2003; Arel et al., 2010; S El-Tantawy & Abdulhai, 2010; Shoufeng et al., 2008), surprisingly, the majority of the previous works developed ATSC for a roadway network. In addition, all the proposed ATSCs for network control are formulated as distributed multi-agent systems, which means each agent (signal controller) is autonomous and responsible for controlling only a signal intersection. The centralized system, which one signal controller controls the whole roadway network, is not preferred as it is computational extremely expensive to deal with such a large state-action space. Therefore, the Multi-agent Reinforcement Learning (MARL), which consists of a number of RL agents, is employed. In the context of MARL, each agent observes a partial state, makes independent actions to maximum its local rewards but learns and acts in the same environment.

However, the coordination between the autonomous agents is always a problem especially sometimes the coordination between signal controllers is needed. Some studies tried to coordinate the neighboring agents by sharing information such as observed state (Balaji et al., 2010; S El-Tantawy & Abdulhai, 2010; Houli et al., 2010; Wiering, 2000; Wu et al., 2012) and reward or Q-value (Salkham et al., 2008; Vidhate & Kulkarni, 2017). Others (Monireh Abdoos et al., 2014; Tianshu Chu et al., 2016; Jin & Ma, 2018; Kuyer et al., 2008; Xu et al., 2013) define a hierarchical learning structure that the lower level is the MARL and the upper level governs the communications between agents.

2.2.2 State Representation

The definition of state *s* will highly influence how the traffic control agent observes the environment. Traffic flow parameters are widely used in ATSCs by Conventional RL, such as: flow or vehicle counts (Balaji et al., 2010; Camponogara & Kraus, 2003; de Oliveira et al., 2006; Salkham et al., 2008; Thorpe & Anderson, 1996; Wiering, 2000), occupancy (de Oliveira et al., 2006; Jin & Ma, 2015; Salkham et al., 2008; Thorpe & Anderson, 1996), and speed (Kuyer et al., 2008). Current traffic signal status like current phase and/or the elapsed green time are also considered by some research (Aslani et al., 2017; S El-Tantawy & Abdulhai, 2012; Jin & Ma, 2015; LA & Bhatnagar, 2011; Lemos et al., 2018; Richter et al., 2007). Some measures of effectiveness (MOEs) are also employed such as queue length (M Abdoos et al., 2011; Abdulhai et al., 2003; Balaji et al., 2010; S El-Tantawy & Abdulhai, 2012, 2010; LA & Bhatnagar, 2011; Lemos et al., 2018; Vidhate & Kulkarni, 2017), the number of queued (waiting) vehicles and/or its waiting time (Aslani et al., 2017; Bazzan et al., 2010; Heinen et al., 2011; Vidhate & Kulkarni, 2017) and delay (Arel et al., 2010; S El-Tantawy & Abdulhai, 2010; Shoufeng et al., 2008). Interestingly, the early works tend to depend more on traffic flow parameters while relatively recent works tend to utilize more MOE related parameters which directly influence the long-run goal or are even part of it. This implies the change from less to more model-based. It is expected the performance of the model should be better since in extreme cases, learning from scratch is definitely more difficult than learning a deterministic function. However, the benefits of the model-free system will highly possibly decrease.

2.2.3 Action Definition

Action defines how the agent controls the signal. There are three major kinds of action definitions: directly change a specific (group of) signal indication (Abdulhai et al., 2003; Arel et al., 2010; Camponogara & Kraus, 2003; S El-Tantawy & Abdulhai, 2012, 2010; Houli et al., 2010; Khamis & Gomaa, 2014; Kuyer et al., 2008; LA & Bhatnagar, 2011; Lemos et al., 2018; Richter et al., 2007; Shoufeng et al., 2008; Thorpe & Anderson, 1996; Vidhate & Kulkarni, 2017; Wiering, 2000), find out the splits of phases for next (few) cycles (Monireh Abdoos et al., 2014; Aslani et al., 2017; Balaji et al., 2010; Salkham et al., 2008) and select a particular pre-defined signal timing plan from several candidates (Bazzan et al., 2010; de Oliveira et al., 2006; Heinen et al., 2011).

Directly changing the indication is the most flexible among all three. It makes the agent be able to respond to the traffic dynamic instantly (the decision step lasts typically one second). However, since it is acyclic, sometimes traffic of one specific direction might wait for the green light for a relatively long time. In addition, the acyclic phase structure will make the coordination between adjacent intersections difficult.

Setting the splits is less flexible since the phase sequence typically is fixed to form a cycle and the decision step lasts at least the length of the cycle. It is similar to forecast the traffic dynamics of the next cycle; therefore it might be more challenging due to the stochastic nature of traffic. However, it makes the coordination relatively easier as long as the coordinated signals have similar cycle lengths and appropriate offsets.

Selecting a particular signal timing plan is the strictest one, and in certain cases, it could not adapt to the traffic dynamics.

2.2.4 Reward Definition

The long-run goal of a signal controller is either to minimize the delay and/or the number of stops or maximize the throughput (Actually they are MOEs of signalized intersections). (Note that the “goal” will be used to evaluate the performance of the control algorithms). Therefore, the reward is typically the discretized goal which is the total delay or the waiting time of the queued vehicles during a certain time period (actually it is the penalty) (Abdulhai et al., 2003; Arel et al., 2010; Balaji et al., 2010; Camponogara & Kraus, 2003; S El-Tantawy & Abdulhai, 2012, 2010; Jin & Ma, 2015; Salkham et al., 2008; Shoufeng et al., 2008; Thorpe & Anderson, 1996; Wiering, 2000), stopped vehicles (Bazzan et al., 2010) or the discharged vehicles (Richter et al., 2007). And Khamis and Gomaa used all three MOEs to form a multi-objective reinforcement learning framework (Khamis & Gomaa, 2014). Some studies use queue length directly (Monireh Abdoos et al., 2014; de Oliveira et al., 2006; Heinen et al., 2011; Houli et al., 2010; LA & Bhatnagar, 2011; Vidhate & Kulkarni, 2017). While the number of vehicles discharged from the queue within a short time period is a good approximation of the delay, whether the queue could be to accumulate to a clear long-run goal is questionable.

2.2.5 Backend RL Algorithms

The majority of the previous research employed value-based RL algorithms such as Q-Learning (M Abdoos et al., 2011; Monireh Abdoos et al., 2014; Abdulhai et al., 2003; Arel et al., 2010; Balaji et al., 2010; Bazzan et al., 2010; Camponogara & Kraus, 2003; Tianshu Chu et al., 2016; de Oliveira et al., 2006; S El-Tantawy & Abdulhai, 2010; Samah El-Tantawy et al., 2014; Heinen et al., 2011; Jin & Ma, 2015, 2018; Khamis & Gomaa, 2014; Kuyer et al., 2008; LA & Bhatnagar, 2011; Lemos et al., 2018; Salkham et al., 2008; Shoufeng et al., 2008; Vidhate & Kulkarni, 2017; Wiering, 2000) and SARSA (Jin & Ma, 2015; Thorpe & Anderson, 1996), while only a few research used Actor-Critic (Aslani et al., 2017; Richter et al., 2007) which is the combination of value-based and policy-based RLs. A possible reason is that Actor-Critic requires a lot more computational resources than only value-based RLs. Therefore, as long as Q-Learning or SARSA have promising performance over the benchmarks, it is not necessary for researchers to use complicated actor-critic algorithms.

Since some studies use a large or even continuous state representation, for example, queue length, there might exists the curse of dimensionality. Some of those studies tackle the issue by function approximation like simple linear function (T Chu et al., 2016; LA & Bhatnagar, 2011; Vidhate & Kulkarni, 2017), tile coding (Monireh Abdoos et al., 2014; Aslani et al., 2017), K nearest neighbors (KNN) (Jin & Ma, 2018) and neural networks (Heinen et al., 2011). Some of those utilize model-based RL (Houli et al., 2010; Kuyer et al., 2008; Wiering, 2000) that Q-value is a pre-determined model of state and action based on human knowledge that is similar to RL with function approximation in some aspects. Others shrink the state space by fuzzy logic, for instance, categorize the continuous values into some levels (M Abdoos et al., 2011; Monireh

Abdoos et al., 2014; Arel et al., 2010; Balaji et al., 2010; Bazzan et al., 2010; S El-Tantawy & Abdulhai, 2010; Khamis & Gomaa, 2014). Admittedly, both function approximation and fuzzy logic lead to information loss. In order to reduce the loss as much as possible, fuzzy logic should be avoided and function approximation should be as complicated as possible.

2.2.6 Simulation

RL agents need training before the implementation like all other ML agents. None of the proposed RL-based ATSCs were trained in the real-world. It is obvious that no one could afford the cost of a non-mature signal control system. As a result, different kinds of simulation testbed are used by researchers, ranging from simple traffic simulators developed by the researchers themselves (Abdulhai et al., 2003; Camponogara & Kraus, 2003; Richter et al., 2007; Thorpe & Anderson, 1996; Wiering, 2000), open-source traffic simulators (Bazzan et al., 2010; de Oliveira et al., 2006; Heinen et al., 2011; Jin & Ma, 2015, 2018; Kuyer et al., 2008; LA & Bhatnagar, 2011; Lemos et al., 2018; Vidhate & Kulkarni, 2017), such as GLD (Camurri et al., 2006), ITSUMO (da Silva et al., 2006) and SUMO (Krajzewicz et al., 2012), to commercial traffic simulators (S El-Tantawy & Abdulhai, 2012, 2010; Xu et al., 2013) such as PARAMICS (Smith et al., 1995) and AIMSUN (Barcelo, 2018). Although open-source traffic simulators provide more flexibility for researchers to customize, the validity of the trained algorithms depends highly on how well the simulator could replicate the vehicle’s behavior in the real world.

2.2.7 Benchmark

Early exploratory studies might use random signals (Camponogara & Kraus, 2003). Almost all other previous studies use fix-timing signal control either hypothetical ones or those implemented in the real world as one of the benchmarks. Some study imitates commercial ATSC systems such as SCATS (Richter et al., 2007; Salkham et al., 2008), GLIDE (Balaji et al., 2010) SCOOT (S El-Tantawy & Abdulhai, 2012) as the benchmark since the technical details of aforementioned systems are prosperity. All the studies conclude that their proposed RL-based ATSCs outperform their different benchmarks.

2.2.8 Summary

Over the past twenty years, a number of researchers developed ATSCs based on conventional RL. The evaluation results show that the proposed ATSCs outperform the fix-timing signal control in terms of various MOEs.

However, the ATSCs based on conventional RL could be improved in several aspects. First of all, as mentioned in the previous sections, due to the curse of dimensionality, simple linear function approximation or fuzzy logic are employed to deduce the dimension of state space even though the information collected is very limited. Take a very simple isolated intersection as an example. Assume an RL signal control agent controls a large intersection with four approaches. Each approach has three through lanes, one left-turning pocket lane and one right turning pocket lane. Two detectors are placed at the beginning and the end of each lane and only the actuation of the detector was collected as the state, which means there are no continuous states. The states of the intersection and its adjacent two intersections are collected to ensure the coordination. Thus, the binary state values of a total of 120 detectors are used as the states. Therefore, there are totally $2^{120} \approx 1.3 \times 10^{36}$ different states. A Q-table is definitely impossible in this case. Take another example. If four levels of queue length (High, Medium, Low, No Queue) of all lanes are used, the state space ($4^{5\times4\times3} \approx 1.3 \times 10^{36}$) is also as large as the previous case. At the end of the day, the functional approximation is proved to be necessary for any realistic representation of the intersection. Furthermore, as the rapid development of connected vehicle technology, vehicle-based data will be largely available. Thus, the complicated traffic states could be represented in a more detailed but nonlinear way. In the era of CAV, a linear function might not be suitable. A non-linear function like neural network make more sense but it is extremely hard to train (Tsitsiklis & Roy, 1997). As a result, both new function form and new training algorithms should be investigated.

Secondly, while there are several centralized successful model-based ATSC, for example, SCOOT (Bretherton, 1990), centralized model-free RL-based ATSC was seldom developed. Despite the benefits of the distributed multiple-agent system, intuitively, the coordination between intersections is never a problem a centralized system as the goal is naturally enhancing the performance of a roadway network as a whole. However, the curse of dimensionality would be even serious. On the one hand, the state space will increase exponentially. On the other hand, since the number of intersections controlled by the controller might be relatively large, the action space will also increase exponentially. While the large state space issue could be tackled by appropriate function approximation, dealing with large discrete action space in the context of signal control still need more research effort.

Thirdly, as discussed in the “Reward Definition” section, most of the previous works only focus on improving the efficiency of the traffic network. However, other important aspects such as safety and emission are overlooked. As a consequence, such factors should be considered as one of the multiple objectives. In order to implement such multi-objective system, the safety and environmental objectives should be carefully discretized into discrete rewards.

2.3 ATSC Based on Deep Reinforcement Learning

As discussed in section 2.2, the ATSCs based on Conventional RL need to be improved to response more complicated traffic dynamics, benefit the network performance by enhancing the coordination between the intersections, consider more factors as the goal of the control agent and adapt to the new era of connected and autonomous vehicles. The first step is developing a better approximator to account for high-dimensional state space. Recently, thanks to the rapid development of DRL which is able to handle the high-dimensional state space, several researchers started to investigate how the DRL could improve the performance of ATSC.

Li, Lv and Wang (L. Li et al., 2016b) firstly conducted an exploratory study on ATSC Based on DRL for an isolated intersection. A four-layer deep stacked autoencoders was employed as its function approximation. Although the state representation used in the proposed DRL control agent is still the queue length rather than more detailed representation of traffic dynamics, the evaluation results show that the proposed control algorithm could reduce the average delay of a simulated intersection by 14% than that with a conventional RL algorithm incorporating with a shallow neural network. Despite their great contribution as the pioneer, their algorithms were trained and evaluated on a simplified simulated intersection. The intersection only has two phases and only allows through movements, which means the agent only needs to conduct a simple binary decision. And the signal might not be realistic enough since it does not configure the yellow and all-red clearance time.

Genders and Razavi (Genders & Razavi, 2016) firstly proposed a new state structure to represent the detailed complex traffic dynamics in replace of aggregated traffic flow parameters. Each lane approaching the intersection is segmented into several small cells and the present and the average speed of vehicles on the cell are used to represent the traffic dynamics. This approach is similar to the "virtual loop detectors" widely used in the computer-vision-based vehicle detectors. Several other studies (der Pol & Oliehoek, 2016; Gao et al., 2017; Liang et al., 2018; Mousavi et al., 2017; Muresan et al., 2018) also utilized.

Although most of the recent studies (Gao et al., 2017; Genders & Razavi, 2016; L. Li et al., 2016b; Liang et al., 2018; Mousavi et al., 2017; Muresan et al., 2018) focus on controlling the isolated intersections, van der Pol and Oliehoek (Van Der Pol & Oliehoek, 2016) firstly implements the distributed MARL using DRL signal control agent on very small grid networks with two to four intersections. The coordination between the DRL agents is achieved by max-plus coordination and transfer planning. The evaluation shows the proposed algorithm is superior to those with a shallow neural network. However, the contribution of the study is limited by its unrealistic traffic simulation. The proposed algorithm only trained and evaluated in an oversimplified network with only one lane per approach and no turning movement.

With the help of DNN, Tan (Tan, n.d.) firstly tried to develop a centralized signal control algorithm for a 3×3 grid network. Unfortunately, the algorithm is not evaluated by any MOEs, therefore, its validity is questionable. Lin et al. also proposed a centralized ATSC (Lin et al., 2018) based on DRL also for a 3×3 grid network. To tackle the possible large action space issue, the authors predefined a four-phase signal structure with a fixed sequence and the agent is only able to choose to either maintain the current phase or change to the next one. Thus, the action space collapse to only 512 actions which is manageable with scarifying the flexibility to skip the phases without any demand. As a result, it is expected that the evaluation results clearly show that the proposed algorithm outperforms the simple actuated signal control only for the saturated condition.

In terms of DRL algorithms employed, the majority of the previous studies utilized DQN like algorithms (Gao et al., 2017; Genders & Razavi, 2016; L. Li et al., 2016b; Liang et al., 2018; Mousavi et al., 2017; Muresan et al., 2018; Van Der Pol & Oliehoek, 2016) and others used actor-critic-like algorithms such as DDPG (Casas, 2017; Tan, n.d.) and Advantage Actor-Critic (A2C) (Lin et al., 2018). The possible reason for the popularity of DQL is that it is computationally less expensive than actor-critic-like algorithms. However, it is not able to deal with large or continuous action space for centralized systems. Therefore, for all the studies aiming at developing a centralized system, actor-critic-like algorithms and necessary hardware are required.

It is great to see the computational power and bright future of ATSC based on DRLs, the current studies also show several limitations.

Firstly, ATSCs based on DRL are intensively validated to be effective for isolated intersections, its effectiveness of network-wide signal control, especially for arterial networks, is not proven. Van der Pol and Oliehoek's work (Van Der Pol & Oliehoek, 2016) is merely effective for up to four intersections allowing only through movements. Casas (Casas, 2017) tested a DRL algorithm for a network-level signal control for a real roadway network in Barcelona. But unfortunately, his/her algorithm does not converge.

Secondly, the centralized DRL controller still needs to be further investigated. As discussed in previous contexts, there is still no effective systems even for an urban gird network. The headache large action space is a potential issue. More advanced DRL algorithms and more reasonable agent design are expected.

Thirdly, as the simulation is inevitable and crucial, the traffic simulation scenarios used by the previous studies are unsatisfactory. Some of them used in early studies are far away from any real-world applications. The intersection geometric design is not considered. The urban grid network is designed with a constant number of lanes or even with a constant length. Some of them do not have turning lanes or even prohibit the turning movement. Others with turning lane do not consider the length of the turning storage bays. The traffic demand is hypothetical, and the route choice behavior is simplified. The only exception is Casas's work (Casas, 2017) using a real-work network, but again his/her algorithm does not converge. The benchmark signal control system is an arbitrary fixed-timing signal, signal controlled by a conventional RL algorithm incorporating with a shallow neural network which is never be applied in the field, or even random signals. (The only exception is the study conducted by Muresan et al. (Muresan et al., 2018) which used a calibrated timing by Synchro based on hypothetical traffic demand.) No real-world signal timing plans were used as the benchmark like the study of El-Tantawy and Abdulhai (S El-Tantawy & Abdulhai, 2012). As a consequence, the effectiveness of those algorithm on the real-world application is hard to evaluate.

Last but not the least, the ATSCs have to be acceptably safe to be implemented in the field. Existing studies neither evaluated the safety effects of their proposed ATSCs nor developed ATSCs that consider safety effects explicitly. Therefore, the impact of DRL-based ATSCs on traffic safety is still concerned yet unclear.

2.4 RL-based ATSC with multiple objectives

Some studies have proposed RL-based ATSC with multiple optimization objectives. Based on the way to achieve the multi-objective optimization, they could be classified into three different types.

The first type is an ATSC which is able to switch its objective dynamically. Houli et al. (2010) developed an ATSC with three different backend single-objective RL algorithms with different goals. But only one algorithm is activated according to the traffic condition. When the current traffic condition is free flow, the goal of the activated algorithm is minimizing the number of stops. When it is under medium traffic condition, the goal turns to minimize the overall waiting time. When there exists congestion, the goal is switched to minimize queue length to avoid queue spillover. This type of ATSC could be regarded as an ensemble of rule-based AI and RL. However, it might be not suitable if the objectives are required to be optimized simultaneously.

The second type of algorithms creates a synthetic reward to represent multiple objectives simultaneously. The most straightforward approach is using the simple/weighted average of multiple rewards. Each reward is associated with a goal. Khamis and Gomaa (2014) proposed an ATSC with 7 different objectives. Five of them indicate different aspects of traffic efficiency, one represents the fuel consumption and the last one is claimed to be "safety reward". The so-called safety reward is not any safety measure but essentially the average speed of the vehicles. The potential issue of this type of algorithm is that its convergence is not thematically proved since the synthesizing reward is no longer the decomposition of any policy goals. It should be noted that the reward of several single-objective ATSCs (Muresan et al., 2018; Van Der Pol & Oliehoek, 2016; Vidhate & Kulkarni, 2017) could also be the average of several components.

The third type of algorithm is multi-objective RL (MORL). Although similar to the second type, MORL manipulates the *value function* (e.g. Q functions) rather than the rewards. In such algorithms, different *value functions* are learned independently using different rewards, while the second type of algorithms learns a signal *value function* by the single synthetic reward. This is beneficial to the convergence of the algorithm especially when the objectives are irrelevant. For a multi-objective ATSC using the aforementioned algorithm, the control agent could either choose the action based on the weighted average of multiple *value functions* or use one or more of them as the thresholds. The study conducted by Jin and Ma (2015) utilized the later method to assign priority to arterials.

2.5 ATSC Considering Traffic Safety

Improving traffic safety is one of the most important objectives for transportation agencies (Lee et al., 2019; P. Li et al., 2020; M. H. Rahman et al., 2019; M. S. Rahman et al., 2019a, 2019b; Yue et al., 2018, 2019, 2020; Zhang et al., 2020). There have been several studies on optimizing signal timing considering traffic safety (Stevanovic et al., 2013, 2015; Zhu et al., 2019). Almost all of them employed surrogate safety measures as the safety indicator and all these studies focused on fixed-timing controllers. The burden of extending such fixed-timing controllers to safety-oriented ATSC is that the most modern ATSCs are designed to be pro-active/predictive. In other words, the ATSC needs to know how the signal timings affect the future safety condition. Therefore, to use the surrogate safety measures (e.g. traffic conflicts) as the safety indicator, a prediction model of the future surrogate safety measures needs to be developed.

To the best of the authors' knowledge, only one published study (Sabra et al., 2013) developed a non-parametrical traffic conflict prediction model and proposed a safety-oriented ATSC accordingly. The study developed a four-stage algorithm tuning the cycle length, splits, offsets, and left-turn phase sequence sequentially. At each stage, the "predicted" number of traffic conflicts is used to evaluate the signal timing tuned by the control algorithm. If the "predicted" traffic conflict of new signal timing is greater than that of the current one, the controller keeps using the current signal timing. Otherwise, the new signal timing is applied. The proposed algorithm is tested in a simulated real-life arterial corridor and a simulated real-life grid network. For the arterial case, although the proposed algorithm reduces the number of traffic conflicts compared with a coordinated actuated signal optimized by the authors, it does increase the number of traffic conflicts compared with the existing field controllers. For the grid network, the algorithm increases the number of traffic conflicts compared with a coordinated actuated signal optimized by the authors, and no testing results of the existing field operation are provided. Therefore, the ability of the proposed algorithms in improving traffic safety is not conclusive.

2.6 Summary

During the past two decades, substantial studies have developed RL-based ATSCs. Most of them perform better than the fixed-timing signals. Due to the curse of dimensionality, conventional RL-base ones utilize fuzzy logic to simplify the control problem. It might lead to its failure for complicated traffic conditions. And they might not be able to directly employ the heterogeneous data from CAVs as their input data. DRL-based ATSCs successfully resolve the issues by employing non-linear neural networks as its functional approximator. However, the existing DRL-based ATSCs did not prove their capability in controlling the real-world traffic at the network level. Besides, their impact on traffic safety remain unclear.

CHAPTER 3: TRAFFIC DATA COLLECTION SYSTEMS ON ARTERIALS

3.1 Introduction

The capability of a traffic signal control system in handling dynamic traffic demands depends on how well it perceives the traffic demands. The fixed-timed signal does not capture the traffic dynamics and the actuated signal control only captures the presence of the traffic. Therefore, they could not well respond to dynamic traffic demands. Although knowing the real-time traffic condition is crucial, most of the existing DRL-based ATSCs go to another extreme as they assume the traffic information of every individual vehicle is known. This unrealistic assumption prevents the implantation of DRL-based ATSCs in the field. Thus, to make the DRL-based ATSCs more implementable, they should utilize traffic data collection systems to perceive the real-time traffic condition. Therefore, existing traffic data collection systems and those will be available in the near future that are also to serve as the data feed of ATSCs should be understood.

3.2 Counts Data from Infrastructure-based Detectors

Traditional actuated traffic signal control systems require traffic detectors to inform the controller that there is a user requests service. Inductive loop detectors are widely used to detect the vehicles' presence, and it could also be replaced by video detectors or microwave detectors (Arroyo et al., 2015). Inductive loop detectors detect the magnetic presence of a vehicle while video-based and microwave detectors are able to create "virtual loops" to detect the presence of a vehicle. If the vehicles' presence is recorded and aggregated, vehicular traffic counts data are available from those detectors designed for the signal actuation.

One of the most famous data systems utilizing the actuation data is Automated Traffic Signal Performance Measures (ATSPM). ATSPM is employed to measure the actual performance of the traffic signal control systems and provide information for re-timing. In fact, ATSPM data consist of high-resolution monitoring logs of traffic signals which is not limited to the vehicle's actuation. The logs have a simple data structure with only three attributes: timestamp, event type and event parameter (for signifying detector numbers and phases). The events recorded in the log include active phase event (phase on, phase green, etc.), active pedestrian event, detector event (detector on/off), barrier/ring events, phase control events, overlap events, preemption events, coordination events, and cabinet/system events. ATSPM event log data provide rich information about the traffic condition and traffic signal performance on arterials. The actual signal phase sequence and phase timing are able to be estimated from the active phase events, the volume per lane per phase is able to be estimated from the detector events, and other signal performance measures such as queue length, v/c ratio and etc. The ATSPM could provide high-quality data inputs for ATSC, however, the deployment of such systems is really limited across the nation.

State-of-the-practice ATSCs use a combination of detection layouts to estimate the current traffic states, which is later used to adjust traffic control in a network (Board & National Academies of Sciences and Medicine, 2010). Most of the ATSCs require additional detectors beyond those used in the conventional actuated signal control systems to provide good traffic measures for its adaptive control logic. For example, they employ advance detectors to estimate queue length.

In term of the detection technology, most of the state-of-the-practice ATSCs do not require using a specific detection technology as long as they could provide the required data. According to an international survey (Board & National Academies of Sciences and Medicine, 2010), more than 90% of implemented ATSCs utilize inductive loop detectors and no more than half of the implemented ATSCs use video detection systems. In fact, the majority of the state-of-the-practice ATSCs assume the conventional loop detectors as the defect detection technology. The major reason is to fully utilize the detectors of existing conventional actuated signal controllers to reduce the deployment cost. However, there does exist one ATSC, which are implemented widely across the nation, use video detection as its default detection technology. It fully utilizes the flexibility of the video detection system to acquire the lane-based traffic counts, the maximum waiting time and the queue length of each phase.

3.3 Bluetooth Detection System (BDS)

In recent years, traffic agencies have begun to implement Bluetooth detection systems (BDS) to collect travel time and origin-destination data on arterials in real time (Chung et al., 2020). BDS employed Bluetooth technology to detect the vehicles carried with a Bluetooth device (e.g. smartphones and hand-free devices). The advantages of BDS are its relatively low-installation and maintenance cost (Singer et al., 2013) and acceptable data quality (Bhaskar & Chung, 2013; Haghani et al., 2010).

BDS estimates origin-destination on urban arterials and freeways(Friesen & McLeod, 2015) by tracking the trajectory of detected devices. A number of studies have also investigated the feasibility of BDS to estimate the number of travelers. However, the most critical issue is that BDS tends to either underestimate or overestimate the number of travelers. First, as pointed out by several studies (Abedi et al., 2015; Bullock et al., 2010; Malinovskiy et al., 2012), the detection rate of such systems is relatively low. Because the Bluetooth MAC address scanner (BMS) using ordinary Bluetooth protocol could only detect the "discoverable" devices. Typically, newer Bluetooth devices are in the discoverable mode only for the initial connection to enhance security and protect privacy. Thus, although we could assume that the penetration rate of the Bluetooth device is high enough because of the popularity of smartphones, most of them are not always discoverable, which reduces the number of travelers that could be detected by BMS. Second, the number of detected MAC addresses might be greater than the number of detected travelers. The duplicate detection issue occurs when a traveler carries two or more devices. It might lead to overestimating the actual counts. The issue would be more severe for estimating vehicular counts, as there could be multiple passengers in a vehicle, especially when there exist many transit buses. In addition, in order to save budget, ensure the power supply and increase the detection rate, for arterials, Bluetooth detectors are installed within or close to the signal cabinet near the intersection. And the detectors are configured to be able to detect all the "discoverable" vehicles within the intersection. Thus, a single vehicle could be detected by a specific detector more than once within a short period of time due to the large detection range (typically 300 to 400 feet). This leads to a location ambiguity and increases the error of travel time estimation (Araghi et al., 2013; José et al., 2015). Moreover, in an extreme case, when the detection ranges of two detectors overlap, a single Bluetooth device could be seen simultaneously by both detectors. The “ping-pong” effect might lead to detecting a traveler at the adjacent intersections, especially in the urban area where intersections are spaced closely. Third, as pointed out earlier, theoretically, it takes up to 10.24s for a Bluetooth device to be “discovered” by the Bluetooth detectors (Kasten & Langheinrich, 2001). If the speed of a vehicle is high enough, it might not be detected.

3.4 Vehicle-Based Traffic Data

Probe vehicles are one of the most effective methods for collecting traffic data. State-of-the-practice probe-vehicle based data collection systems use the position and speed reported by the global positioning system (GPS)-equipped probe vehicles. Private sector traffic service companies such as INRIX, HERE and TomTom take advantage of probe vehicle tracking technologies to provide speed and travel time data of the road network including most of the arterials. The major advantage of the vehicle-based data from the private sectors is that the vendors are able to provide more extensive geographic data coverage than any kind of infrastructure-based traffic data. To benefit from this, one recent study (Sharma et al., 2019) employs the vehicle-based travel time (speed) data from the private sectors to develop a rule-based ATSC. The proposed ATSC first forecast the near-future speed using a single exponential smoothing technique. Then a multivariable linear regression model is developed by utilizing forecasted link-level speed, signal control variables, and link length as predictors to estimate traffic flow parameters. A simple signal timing algorithm uses the forecasted traffic flow parameters to generate an optimized signal timing. The study indicates that the widely available travel time data could be used as a valuable input of ATSCs. However, there is often a concern that the exact data source and processing algorithms are always proprietary (Elefteriadou et al., 2014).

Connected vehicles (CV) are another source of vehicle-based data. Several studies (Feng et al., 2018; Goodall et al., 2013; Joyoung et al., 2013) have employed CV's trajectory data as the ATSC's input. However, the proposed ATSCs requires a relatively high market penetration rate. The least market penetration rate of CV required by these ATSCs is around 10% (Feng et al., 2018). Until March 2020, there is no ATSC based on the CV is implemented in the field.

3.5 Summary

Existing traffic data collection systems in the field include traffic counts from inductive loop detectors installed for signal actuation, travel time and origin-destination data from BDS and segment-based speed data from the private sectors. However, the infrastructure-based systems, either the inductive loop detectors or BDS, is limited in terms of geographical coverage. Thus, the widely available segment-based speed data from the private sectors could be served as valuable data feeds of ATSCs.

Most of state-of-the-practice ATSCs requires customized detection layout for their own control logic. Especially some vendors depend on the video detection systems. The rapid development of CV brings the opportunity for ATSCs to use the vehicle's trajectory as its input data. However, there is still no ATSC based on the CV is implemented in the field until March 2020.

CHAPTER 4: ESTIMATING NON-MOTORIZED TRAFFIC COUNTS FROM BDS USING BLUETOOTH LOW ENERGY

4.1 Introduction

Quantifying the traffic counts is important for the ATSCs. Some Recent studies have looked at Media Access Control (MAC) address scanning sensors such as Bluetooth and Wi-Fi scanners. Such kinds of sensors detect and track travelers indirectly by communicating with the Bluetooth/Wi-Fi devices carried by them through radio waves. However, as mentioned in the previous chapters, the biggest challenge of using Bluetooth/Wi-Fi sensors for estimating counts is that they are only able to detect a portion of the travelers. The detection rate of the Bluetooth sensors is only 0.6%-6.8% (Abedi et al., 2015; Bullock et al., 2010; Malinovskiy et al., 2012), while that of the Wi-Fi sensors varies from 9% to 97% for different environments (Abedi et al., 2015; Du et al., 2017; Kurkcu & Ozbay, 2017; Ohlms et al., 2019; Schauer et al., 2014). Thus, in order to estimate the accurate counts, extrapolation of the raw data based on empirical detection rate is inevitable, which might introduce errors.

To resolve the limitations of the current MAC scanning based counting systems, recently available Bluetooth Low Energy (BLE) scanners are advocated. This chapter developed a system estimating the count of pedestrians and bicyclists of the intersection. BLE is an upgraded version of Bluetooth, which is able to detect more devices than the ordinary one (Gomez et al., 2012). Since to the best of the authors' knowledge, there is no published study evaluating the applications of BLE technology in transportation. This study starts with the feasibility assessment of BLE to obtain pedestrians and bicyclists counts in terms of detection rate, detection range, and data quality. Then, a travel mode classification algorithm based on machine learning is proposed to help to estimate accurate counts. The system is implemented in a test site to evaluate the performance and prove the viability of the approach.

4.2 Assessing the Feasibility of BLE to Obtain Pedestrians and Bicyclists' Counts

As mentioned in the last section, compared with ordinary Bluetooth MAC address scanner (BMS-O), BLE based Bluetooth MAC address scanner (BMS-BLE) has a higher detection rate, which might be preferable for estimating counts of pedestrians and bicyclists. As there is no published study assessing the BMS-BLE used in BDS, the author conducted several pre-experiments to assess its feasibility. The first pre-experiment aims to assess the detection rate of BMS-BLE. The objective of the second pre-experiment is to verify whether the detection range is large enough to cover the intersection while avoiding the "ping-pong" effect.

Eight BMS-BLEs of a commercialized BDS were used. They were installed at eight signalized intersections within and near the campus of the University of Central Florida (UCF), U.S. Three of them were deployed along a major arterial at the western edge of campus where there are existing BMS-Os. These detectors were employed to compare the characteristics between BMS-BLE and BMS-O. Five of them were deployed along the major ring road of the campus. These detectors were used to detect the non-motorized travelers as there are a lot of pedestrians and bicyclists crossing those intersections.

First, the detection rate, in terms of the number of detected unique devices of BMS-BLE was compared with that of BMS-O were conducted. One-week data (from 07/01/2019 to

07/07/2019) were used for the comparison. Figure 2 illustrates the comparison results. It clearly shows that all three BMS-BLEs detect more devices than BMS-Os. The results confirm that the detection rate of BMS-BLE is higher than that of BMS-O.

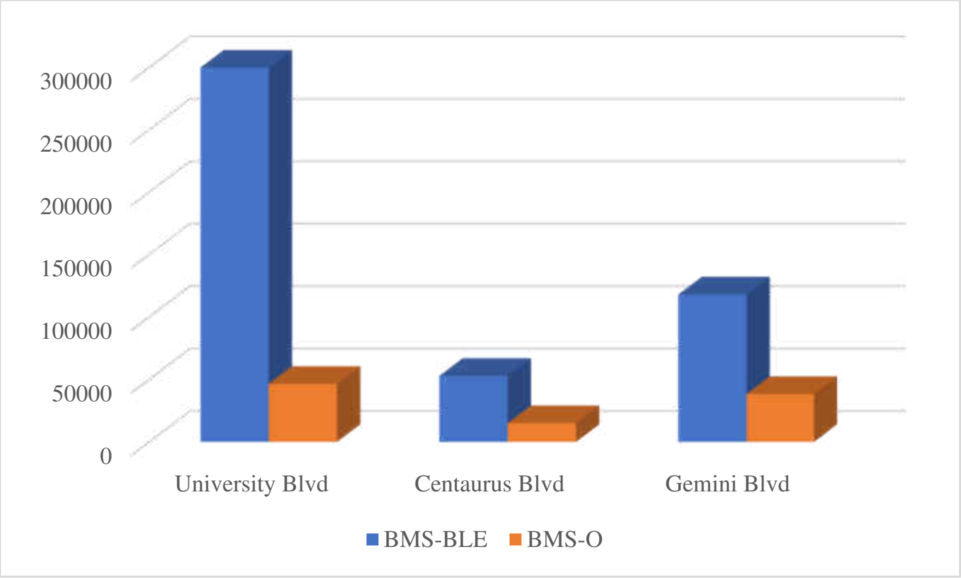


Figure 2 Comparison of unique devices detected from 07/01/2019 to 07/07/2019

Then, to verify the actual detection rate of BMS-BLE, one intersection of the ring road with heavy volumes of both vehicles and pedestrians/bicyclists was selected as the study site. Around 3-h videos were recorded by an action camera on two weekdays. The camera was mounted high enough to cover the whole intersection. The number of pedestrians/bicyclists and vehicles were counted manually. The corresponding BMS-BLE records were also collected and the counts extracted from the video footage serve as the benchmark data.

Table 1 shows the number of unique devices captured by BMS-BLE and the benchmarking traffic counts. The number of detected unique devices by BMS-BLE is greater than the benchmark counts. Although such a high detection rate (over 100%) was never seen in any published studies assessing the detection rate of BMS-O, it is reasonable because there might be duplicate detections of a single traveler. Moreover, the benchmark count of motorized traffic is the number of vehicles rather than the number of travelers. Since there were several university shuttles passing at the intersection during the time the video recorded, it is expected that the number of motorized travelers is greater than the number of vehicles. In addition, some fixed devices might also be included as the data had not been pre-processed. Although it is not possible to find out the actual detection rate of travelers, it is reasonable to conclude that BMS-BLE is able to provide sufficient detections to estimate the counts of pedestrians/bicyclists. However, the raw data from the BME-BLE is subject to cleaning.

Table 1 Detection Rate Assessment (Raw Data)

Source	**Benchmark**			**BMS-BLE**	**Detection Rate**
	Vehicle	Pedestrian/Bicyclist	Total		
Counts	3930	2060	5990	9685	161.69%

In the second pre-experiment, volunteers were asked to drive along the roadways with Bluetooth devices, and then the locations where they were detected by the BMS-BLE were investigated. A preliminary test was conducted to investigate what kinds of Bluetooth devices could be detected by the BMS-BLE. Several different kinds of devices were tested at the intersection where both BMS-O and BMS-BLE are installed. Bluetooth headphones could be only detected by BMS-BLE. However, BMS-BLE is able to detect both devices compatible with only Bluetooth ordinary and devices compatible with BLE, such as smartphones and laptops. A possible reason is that the vendor of the BDS would like to ensure the detection rate.

The ring road in the campus is selected as the study site since the BMSs are closely located. And two kinds of devices were carried by the volunteers: “discoverable” smartphones representing the Bluetooth ordinary devices, and Bluetooth wireless headphones representing the BLE devices. The trajectories of the volunteers were recorded using smartphone GPS tracking applications. These applications provide detailed trajectory points in one-second granularity. After GPS trajectory points were collected, they were matched with Bluetooth detections from five BMS-BLEs by MAC addresses of the devices and recorded timestamps.

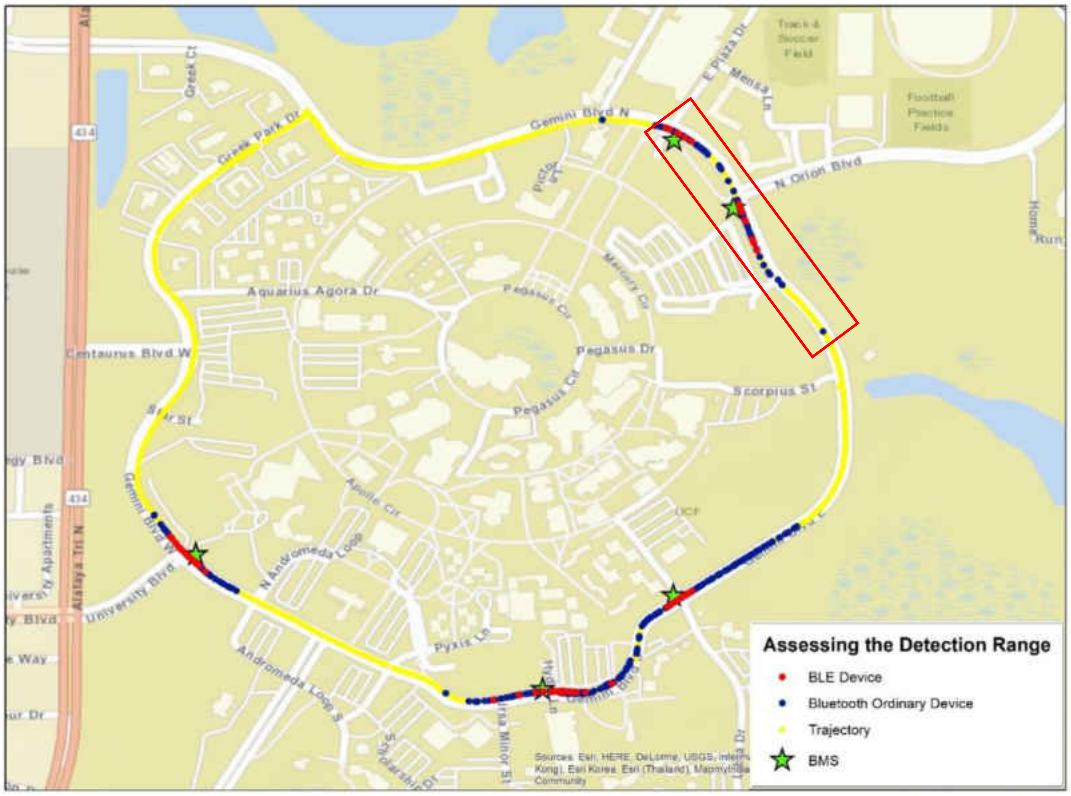


Figure 3 Comparison of unique devices detected from 07/01/2019 to 07/07/2019

Figure 3 is the plot of the volunteers’ trajectory points and those matched with the Bluetooth data. At first glance, the detection range of BMS-BLE is large enough to cause the "ping-pong" effect. For example, the Bluetooth Ordinary Devices were almost continuously detected when the volunteers passed E Plaza Dr and N Orion Blvd (Red rectangle in Figure 5). However, as BMS-BLE could detect both BLE and Bluetooth Ordinary devices, if we assume that the BMS-BLE consists of two "virtual" modules: the BLE module detecting only BLE devices while the Bluetooth Ordinary Module detects only Bluetooth Ordinary devices. The detection range of the "virtual" BLE module is much less than the "virtual" Bluetooth Ordinary module and it is small enough to avoid the "ping-pong" effect while covering the whole intersection. Moreover, as the previous studies have found the detection rate of BMS-O does not exceed 10% (see Table 3), according to the high detection rate showed in pre-experiment one, most of the devices should have been detected by "virtual" BLE module. Therefore, compared to the BMS-O, the BMS-BLE is able to reduce the error caused by the "ping-pong" effect while still ensuring the coverage of the intersection.

4.3 Methodology

Data from BMS provides the number of devices detected during a specific period time. However, it never identifies whether a specific device is carried by a pedestrian, a driver or a police officer directing the traffic at the intersection. In order to estimate the counts of pedestrians and bicyclists, devices carried by them have to be identified. The identification process consists of two stages. First, stationary devices are filtered out from the moving devices. Then, the devices carried by the motorized travelers are filtered out. The leftovers are the non-motorized ones. This process is accomplished with the help of a machine-learning algorithm: one-class support vector machine.

The raw data from BMS typically consists of only three parameters: the MAC address of the detected device (usually only last six digit is kept due to privacy), the timestamp when the device was detected, and the Received Signal Strength Indicator (RSSI) of the detected device. Given the same device and similar environment, the RSSI is related to the distance between BMS and device. However, none of these three parameters can be directly used for classification algorithms. Fortunately, because of the large detection range of BMS, a device can be detected multiple times as long as it stays in the detection range. Therefore, there might be several records in raw data from BMS of a single "visit" by a specific device to an intersection. As RSSI could serve as a relative location indicator, these records could provide valuable information to infer the behavior of devices, such as the travel time, time mean speed, etc.

In this study, we use the term "RSSI-trajectory" estimated from the multiple detection records of a device to infer its behaviors. The trajectory of a traveler is its spatial location as a function of time. Similar to the actual trajectory, the "RSSI-trajectory" is the RSSI of a device as a function of time (Figure 4). The black line shows the theoretical "RSSI-trajectory". The five red dots are the detection records obtained from the BMS, which is a discrete sample of the "RSSI-trajectory". The parameter duration D, which is defined as the elapsed time between a device entering and exiting the detection range of a specific BMS, is used to classify whether a device is moving or not. The value of D of a permanent stationary device, such as a device installed at the intersection is theoretically close to positive infinity. The value of D of a "relative" stationary device, such as the device carried by a police officer directing the traffic is sufficiently large. On the contrary, D value of a moving device, even if it is stopped by a red light, should be smaller than a threshold value D_{th}. Therefore, the classification is done as such:

$$device\ is \begin{cases} stationary & when\ D > D_{th} \\ moving & when\ D \le D_{th} \end{cases}) \tag{11}$$

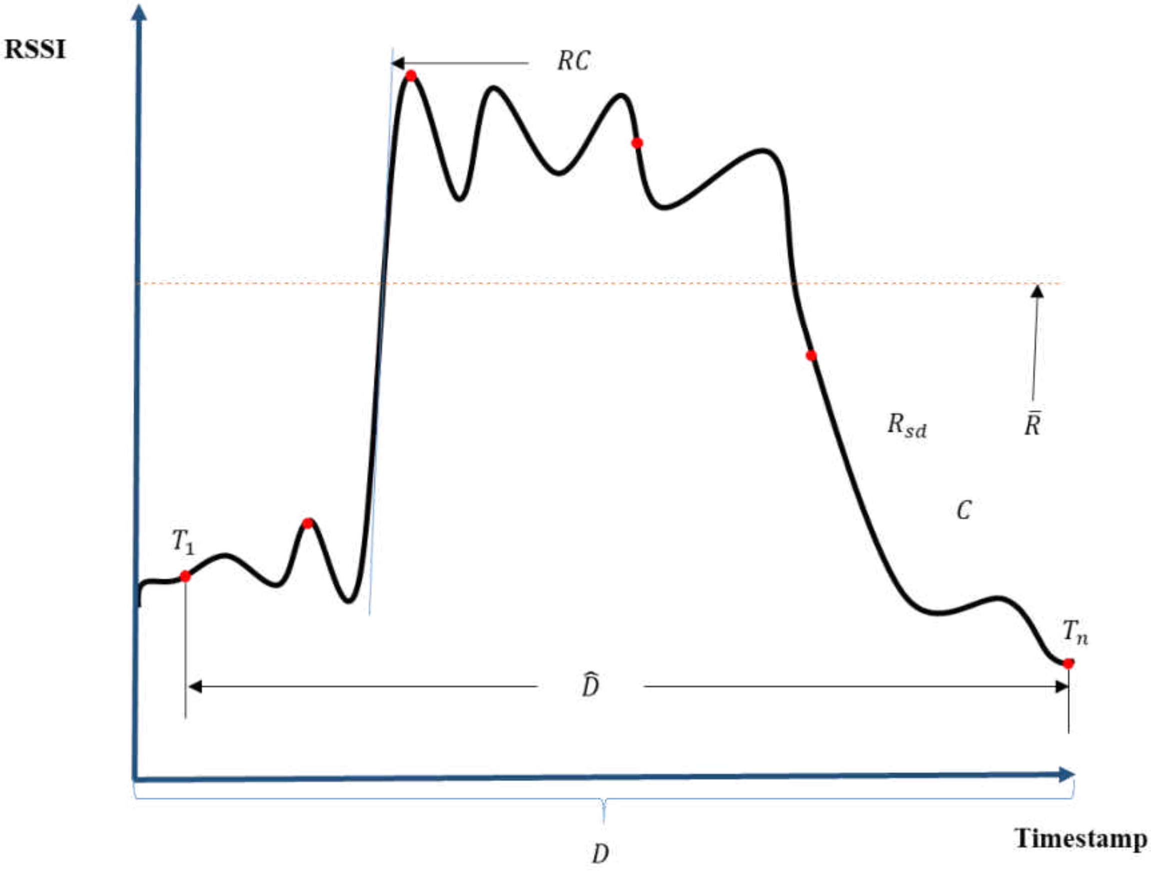


Figure 4 Plot of the Trajectories and the Matched Trajectory Points

$\boldsymbol{D_{th}}$ is defined as the longest time a traveler could spend at the intersection:

$$D_{th} = \frac{d_{max}}{v_{min}} + t_{maxwait} \tag{12}$$

where $\boldsymbol{d_{max}}$ is the longest distance a road user could travel at the intersection. If only motorized travelers are considered, the $\boldsymbol{d_{max}}$ is the detection range; but if pedestrians and bicyclists are also considered, the $\boldsymbol{d_{max}}$ should be the maximum value between detection range and longest possible route of the pedestrian or bicyclist. The length of the route needs to be determined based on the geometry of the specific intersection. $\boldsymbol{v_{min}}$ is the minimum speed of the travelers. $\boldsymbol{t_{maxwait}}$ is the maximum waiting time of the travelers.

However, in practice, it is infeasible to obtain D as the discrete data from BMS do not always capture the exact timestamps when the device entered and exited the detection range. Thus, in this study, D is approximated by $\widehat{D}$:

$$\widehat{D} = T_n - T_1 \tag{13}$$

The multiple detections recording a single visit of a device were identified and grouped. $\boldsymbol{T_1}$ is the timestamp when the device was first seen by the BMS, which is the first record in the group. $\boldsymbol{T_n}$ is the timestamp when the device was last seen, which is the last record in the group. Thus, the classification process becomes:

$$device\ is \begin{cases} stationary & when\ \widehat{D} > D_{th} \\ moving & when\ \widehat{D} \leq D_{th} \end{cases} \tag{14}$$

To accurately estimate the counts of pedestrians and bicyclists from mixed traffic, algorithms are needed to classify the travel mode. The naïve approach to classify the travel mode is using its speed:

$$device\ is\ carried\ by \begin{cases} vehicle & when\ s_o \geq s_v \\ bicyclist/pedestrain & when\ s_o < s_v \end{cases} \tag{15}$$

where s_o is the operating speed of the device and s_v is the preset threshold representing the minimal speed of the vehicle. Determining the value of s_v is not trivial as the speed of vehicles are comparable to the speed of the bicycles when the roadway is congested (see the red area of Figure 5). Therefore, the performance of the naïve approach is often not satisfactory.

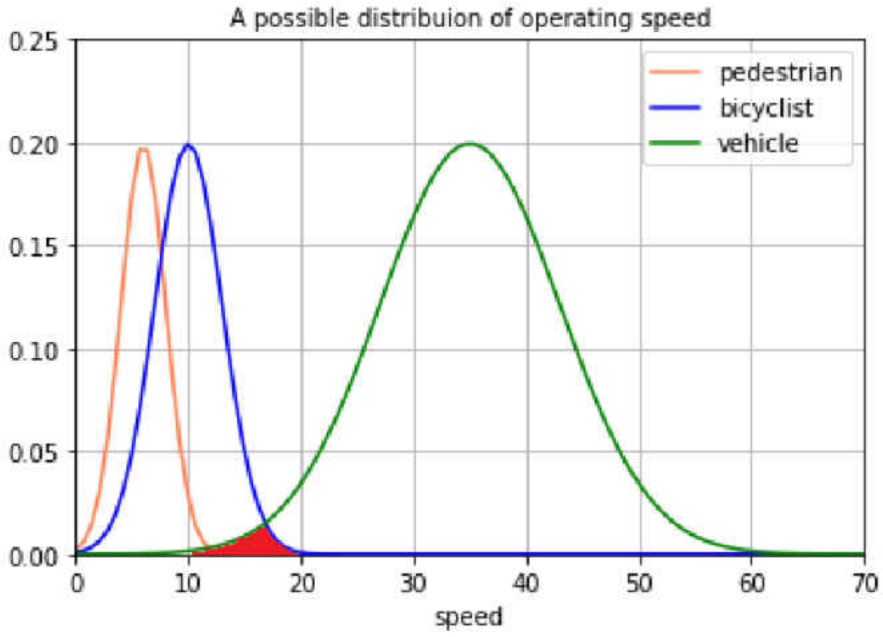


Figure 5 Illustration of the speed-readings that are hard to be classified

Most of the existing studies depend on the speed/travel time (Araghi et al., 2016; Bathaee et al., 2018; S. Liu et al., 2014; Yang & Wu, 2018). However, MAC scanning based systems could only obtain the travel time of those who are tracked by at least two sensors. For urban areas with a higher number of pedestrians and bicyclists, some of them detected by a sensor at the intersection are “lost” since they have reached their trip destination. Thus, the speed of those “lost” ones cannot be obtained from the BDS. Therefore, the aforementioned classification method, which requires knowing the speed of every device, could not be used in this study.

Fortunately, there is plenty of information that could be extracted from the "RSSI-trajectory". If it is possible to obtain a pre-labeled sample of the travelers and their "RSSI-trajectory", a model could be estimated to classify the travel mode of other travelers.

The time of a device traveling from a BMS to another BMS and space mean speed could be calculated by matching the MAC address:

$$s_{(d,ab)} = \frac{T_b - T_a}{L_{ab}} \tag{16}$$

where $\boldsymbol{T_a}$ is the timestamp when a device is seen at the BMS a; T_b is the timestamp when a device is seen at the another BMS b; $\boldsymbol{L_{ab}}$ is the length of the roadway segment connecting a and b; and $s_{(d,ab)}$ is the space mean speed of the device d traveling from a to b. It should be noted that in practice, the BMS is often installed close to the intersection. Thus, the travel time sometimes includes the waiting time and starting lost time of a traveler at one or both intersections, which eventually leads to the estimated space mean speed to be lower than the operating speed.

Then the naïve classification approach could be applied to filter out the non-motorized travelers if s_v is set appropriately. There is a trade-off in selecting the s_v. The higher the s_v is, the higher the probability the motorized travelers are labeled correctly. But if the s_v is set to be too high, the identified sample could not capture the vehicles that stopped at the intersection waiting for the green light. Careful selection of s_v should be conducted. Similarly, a sample of non-motorized travelers could also be obtained if the s_v is set low enough.

It should be noted that the naïve classification is only for obtaining a sample of vehicles or pedestrians/bicyclists. Because first of all, s_v is set to a value that could allow the travelers whose speed could not serve as a reliable classifier (red area of Figure 7) to be excluded. Secondly, the speed of some travelers is not available as stated earlier. Furthermore, as pointed out by the pre-experiment three, in this particular study, it is only feasible to get the sample of vehicles but not the sample of pedestrians and bicyclists.

Traditional classification models, no matter whether it is the straightforward logistic regression or the sophisticated deep neural network, are trained by the data labeled with all classes. While in this study, only data from one class, motorized travelers, are available. Therefore, the model should be trained purely by the one-class data and be able to determine whether a specific data point belongs to the class or not. As a result, one of the machine-learning algorithms, one-class support vector machine (SVM) is used.

The objective of one-class SVM (Bernhard Schölkopf et al., 2001) is approximating the distribution of data belonging to the specific class. It aims to find a function such that most of the data points live in the "small" region where the function is one, and minus one elsewhere. Although the labeled sample of motorized travelers is used in this study to train the one-class SVM, it could also be trained by the labeled sample of pedestrians/bicyclists as long as it is available. The performance of the model is evaluated by a mixed sample of motorized and non-motorized travelers.

As mentioned earlier, the information provided by BMS indicating the behavior of a detected device is summarized as its "RSSI-trajectory". Since RSSI is a location indicator, variables describing the "RSSI-trajectory" might indicate the speed, acceleration, waiting time at the intersection, etc. As the continuous "RSSI-trajectory" is not available from BMS data (see Figure 6), discrete detections of a single device serve as the approximation and all the variables are calculated using the detections. Table 2 shows the definitions of the variables. Figure 6 also shows the graphical illustration of the variables. Among all six parameters, C and $\widehat{D}$ serve as the indicator of the travel time within the detection range. R_{sd} and $\bar{R}$ are statistics of RSSI. RC approximates the derivative of RSSI value by time, which is an indicator of point speed. Thus, $\overline{RC}$ indicates the time-mean speed. Theoretically, the detection with maximum RSSI represents the closest point of a device to the detector. Thus, t_m is an indicator of the waiting time, especially for vehicles.

Table 2 Variables describing the RSSI-trajectory

Variable	Description
C	Number of detections (trajectory points) of the devices
$\widehat{D}$	Time difference between the last and first detection
R_{sd}	Standard deviation of RSSI value among the detections
$\bar{R}$	Mean RSSI value among the detections
$\overline{RC}$	The mean change rate of RSSI. The change rate of the RSSI is approximated by the difference of RSSI value between two consecutive detections.
t_m	Time difference between the first detection and the detection with Maximum RSSI

4.4 Validation

The proposed model is validated by the same video footage used in the pre-experiment one. Because of the setting of the camera, the 3-h video was segmented to 30 video clips. The average length of the clips is around 6 minutes. The detection data from BMS-BLE at the same time period were also collected. Then the stationary data filter was applied to the raw data. In this case study, the D_{th} is set to be 180 seconds. The first-stage filtering results show that there were 9,685 unique devices that were detected by 118,061 times. Out of the 9,685 devices, 8,332 (86.03%) were identified as moving objects, which results in the actual detection rate lower to 139.10%. A possible reason for such a high number of stationary objects might due to an event at the arena close to the intersection when the data were collected. The video clips showed that many people are queued to obtain the tickets for the event within the detection range of the BMS-BLE.

In this case study, due to the limited sample size, a ten-fold cross-validation is used to evaluate the performance of the one-class SVM. The filtered BMS-BLE detection data were also divided into 30 slices corresponding to the 30 video clips (Figure 6). Then they were grouped into ten folds where each fold contains three consecutive slices. For every one-class SVM model, nine folds are used to obtain the training data and the last fold is used to test the model.

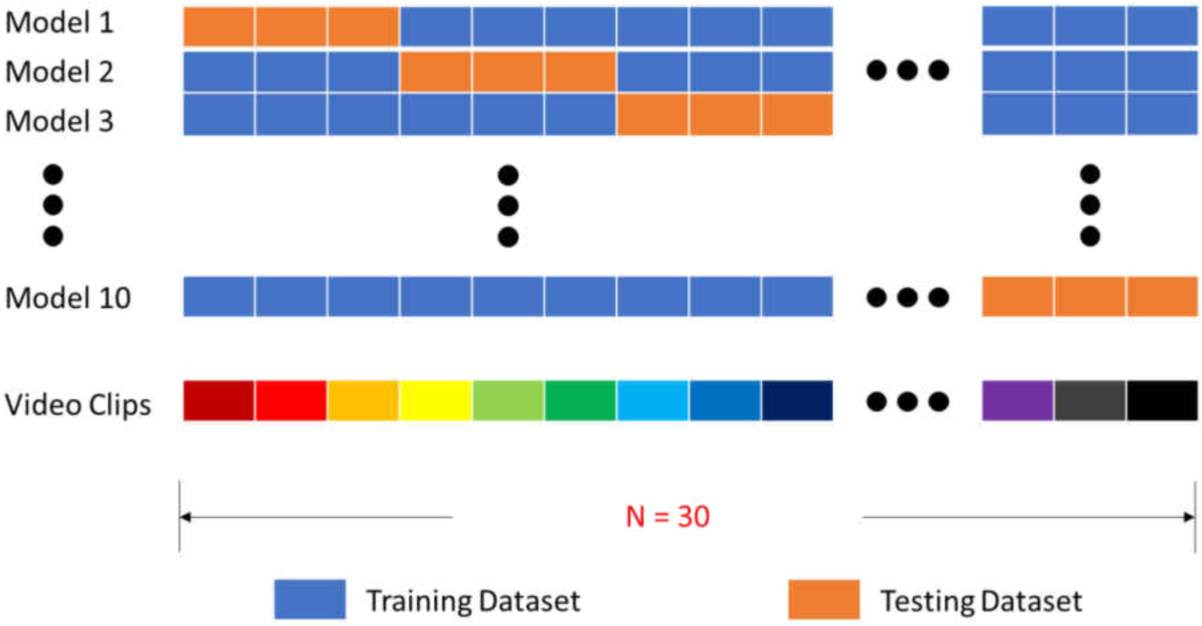


Figure 6 Illustration of the ten-fold cross-validation used in this study

To obtain the sample of motorized travelers, data from the upstream and downstream BMS-BLEs of the same time period were collected. Records with the same MAC address were matched and space mean speeds were calculated by formula (6). s_v was set to be 13 mph, which is the average operating speed of the bicycle measured in study by the Federal Highway Administration, U.S (Hummer et al., 2006). As stated in the previous section, space mean speed is typically lower than the operating speed, such setting is able to ensure that most of the identified devices are carried by motorized travelers.

The one-class SVM is deployed using scikit-learn (Pedregosa et al., 2012) package in Python 3 environment. The implementation of one-class SVM in scikit-learn requires the choice of a kernel and a scalar parameter to define the frontier. In this case study, the KBF function is chosen as the kernel because it is the default in the scikit-learn implementation. In addition, the regulation parameter v was set as 0.05 in order to capture the characteristics of most training data while preventing the over-fitting.

Table 3 shows the results of the cross-validation. There are several evaluation metrics presented. The accuracy in estimating the number of pedestrians and bicyclists is used as the major performance metric. It is defined as the number of predicted non-motorized travelers divided by the ground truth from the video. The miss-identifying rate is defined as the false negatives divided by the number of labeled motorized travelers. The definition of the confusion matrix in table 3 is slightly different from the two-class classification problem. Because in the test datasets, only a portion of motorized travelers was correctly identified. The travel mode of the remaining portion is unknown and there are no labeled pedestrians and bicyclists. Therefore, the true positive (TP) means that a known motorized traveler is classified as a motorized traveler.

The false negative (FN) means that a known motorized traveler is misclassified as a non-motorized traveler. The true negative (TN) means that an unknown mode traveler is classified as a non-motorized traveler; and the false positive (FP) means that an unknown mode traveler is classified as a non-motorized traveler.

Table 3 Ten-fold validation results.

Model	Ground Truth		Confusion Matrix				Performance Metrics	
	Motor	Non-Motor	TP	FP	FN	TN	Miss Rate	Accuracy
1	448	355	315	705	6	70	7.9%	90.4%
2	235	89	85	249	0	44	0.0%	95.5%
3	261	123	125	275	9	68	11.7%	108.9%
4	268	131	115	320	6	65	8.5%	92.4%
5	317	190	172	468	4	50	7.4%	92.6%
6	418	195	197	528	3	70	4.1%	102.6%
7	556	195	208	656	4	73	5.2%	108.7%
8	608	296	271	836	2	68	2.9%	92.2%
9	491	254	255	826	2	70	2.8%	101.2%
10	328	232	215	836	5	56	8.2%	94.8%
Average		Miss Rate	Absolute Percentage Error		Accuracy			
					Non-Motor		Motor	Total
10-fold		5.9%	6.35%		97.9%		160.1%	138.2%

On average, the accuracy of the model in estimating pedestrians and bicyclists counts is 97.9% while the miss rate is 5.9%. The average absolute percentage error in estimating pedestrians and bicyclists counts is 6.35% and the absolute percentage error does not exceed 10% for every validation fold. Moreover, the number of pedestrians and bicyclists counts per fold varies from 89 to 296, which confirms that the model works well for different traffic conditions. Thus, the proposed model is able to estimate the counts of pedestrians and bicyclists with relatively low error.

The cross-validation results show that the predicted counts are close to the actual observed counts. However, as the proposed model is a classification model, one might still question whether the model successfully identifies a pedestrian/bicyclist or just coincidently generates plausible counts. Although there are no labeled pedestrians/bicyclists for verification, statistics of variables describing the RSSI trajectories by class might provide some insights of the travel behavior that they exhibit.

Table 4 shows the mean and standard deviation of the variables. First, the predicted pedestrians/bicyclists exhibit significantly larger C and $\widehat{D}$ than that predicted for the motorized travelers. It is reasonable since the pedestrian/bicyclist should have longer travel time. Second, the variation of $\widehat{D}$ and t_m is greater than that for motorized travelers (in terms of coefficient of variation). Such high variation might be due to the bi-modal vehicular traffic flow pattern (Gong, Abdel-Aty, & Park, 2019) at signalized intersections. The travel time of a vehicle forced to stop and waiting for the green phase is longer than that traveling through the intersection without stopping. Third, given that the mean RSSIs are similar for both classes, the variation of RSSI of the predicted motorized traveler is higher. It might be due to the sharp change of RSSI caused by the acceleration of stopped vehicles waiting for the green signal. A similar phenomenon is also found for $\overline{RC}$. The variation of $\overline{RC}$ is higher for motorized travelers. Therefore, according to the statistics of the variables, the predicted pedestrian/bicyclist exhibits known characteristics of pedestrian/bicyclist while those predicted as motorized travelers do have similar behavior of motorized travelers.

Table 4 Statistics of variables describing the RSSI trajectories by class

Variable	Statistics	Predicted Pedestrian/Bicyclist	Predicted Motorized Travelers
C	Mean	14.35	4.39
	S.D.	19.38	5.74
$\widehat{D}$	Mean	89.91	15.55
	S.D.	54.25	22.11
R_{sd}	Mean	4.12	2.63
	S.D.	3.48	3.17
$\bar{R}$	Mean	-72.83	-73.42
	S.D.	8.85	6.64
$\overline{RC}$	Mean	1.13	0.96
	S.D.	1.15	1.37
t_m	Mean	43.97	7.99
	S.D.	44.52	16.30

Another interesting finding is that the predicted number of motorized travelers is 1.61 times the average number of vehicles. According to the 2017 National Household Travel Survey (Federal Highway Administration, 2017), the average vehicle occupancy is 1.67. Although this study does not focus on estimating the counts of vehicular traffic as the extrapolation using occupancy is inevitable, such finding supports the validity of the model.

Admittedly, this work has several limitations. First, the fundamental assumption of this work is that on average each person carries one BLE/Bluetooth Ordinary detectable device. It is valid for university campuses used as the case study and most of the urban environment of developed countries because of the popularity of smartphones and wearable devices[1]. However, for the area where the penetration rate of smartphones and/or wearable devices is not sufficiently high, the counts estimated by the proposed model should be adjusted by the penetration rate.

[1] The penetration rate of smartphones in the United States in 2019 is projected to be 79.1% according to a global mobile market survey (Newzoo, 2019).

Second, Similar to other machine-learning-based models, the accuracy of simply transferring the model developed to another site is not guaranteed.

4.5 Summary and Conclusion

This Chapter developed a system estimating counts of pedestrians and bicyclists at intersections based upon a commercialized Bluetooth Low Energy traffic detection system. Since there are not any published studies assessing the performance of BLE based traffic detection systems, as part of the contributions, two evaluation experiments were conducted. The experiments confirmed that the detection rate of BLE based traffic detection system is sufficiently high for traffic counts studies and its detection zone is able to cover the whole intersection while reducing overestimation caused by the large detection range in comparison with other MAC address scanning sensors. Thus, in general, it is feasible to employ BLE based traffic detection systems for traffic counts' studies.

A two-step framework is then proposed for identifying the pedestrians and bicyclists from stationary objects and motorized travellers. First of all, the stationary objects are filtered out depending on the length of time observed by the detectors. Then, a one-class support vector machine algorithm is employed to classify the pedestrians and bicyclists from the motorized travellers. The algorithm is trained by a local sample of historical labelled motorized travelers or pedestrians/bicyclists. This ensures the ability of the proposed framework to be applied in real-time.

The system was implemented at an intersection on the university campus to validate the accuracy. Manual count from the three-hour video was used as the ground truth. The one-class support vector machine algorithm was trained by a sample of motorized travellers. The results of a ten-fold validation showed a promising performance. The average absolute percentage error in estimating counts of pedestrians and bicyclists is 6.35%. Thus, the proposed system could reasonably identify the pedestrians and bicyclists from the mixed-traffic environment.

This study also concluded that compared to the traditional Bluetooth and Wi-Fi, BLE is more suitable for estimating the counts of pedestrians and bicyclists. It avoided the error incurred by the extrapolation of the raw detections due to its high detection rate.

The proposed system is compatible with any other BLE based traffic detection systems as long as they provide RSSI values. However, practitioners should be cautious to apply this system to an environment with low BLE device penetration rate or directly transfer the model developed for another area. For the future work, the labelled bicyclists or pedestrian’s data would be collected, and the proposed system would be extended to have the ability to classify bicyclists versus pedestrians.

CHAPTER 5: A DECENTRAILIZED NETWORK LEVEL ADAPTIVE SIGANL CONTROL ALGORITHM BY DEEP REINFORCEMENT LEARNING

5.1 Introduction

As discussed in the previous chapter, the current studies of ATSCs based on DRLs have several potential issues such as 1) the effectiveness of network-wide signal control, especially for arterial networks, is not proven; 2) the traffic simulation scenarios and the benchmark used by the previous studies are unsatisfactory for the evaluation of the effectiveness in real-world application; 3) most of the existing DRL-based ATSCs assumes the complete information of traffic network is known which is not realistic in the field. To fill the gap, this study proposes a network-level decentralized adaptive signal control algorithm with limited known traffic information using one of the famous DRL methods, double dueling deep Q network (3DQN), in a MARL framework. It allows the coordination among adjacent intersections by sharing the information. The proposed algorithm was trained in a simulated suburban traffic corridor with the real-world roadway design. An AM peak traffic scenario is calibrated by the real-world demand and traffic data. Finally, it was evaluated by real-world coordinated actuated signal system whose configuration is provided by the local jurisdiction.

5.2 Decentralized Adaptive Signal Control Algorithm Based on 3DQN

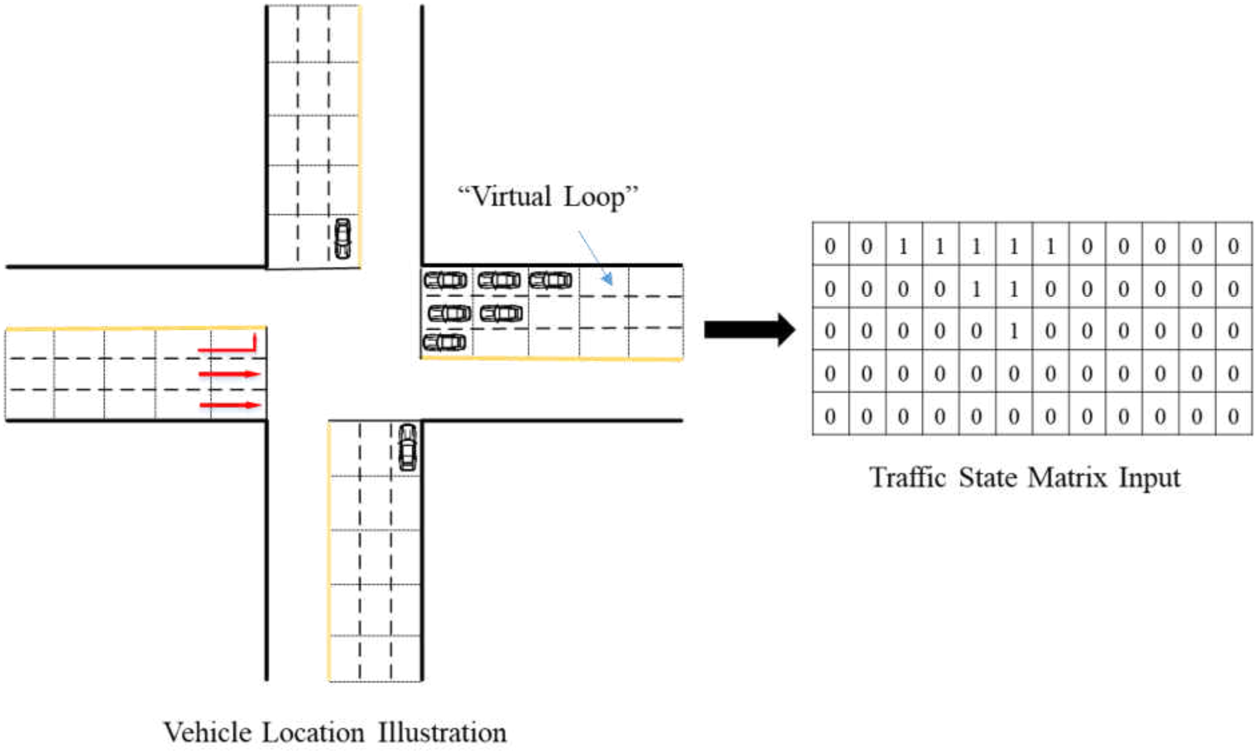
In this study, a decentralized signal control problem is formulated into the standard MARL setting: each individual signal controller acts as an RL agent; the agent observes the condition of the intersections as the state; the agent directly selects the appropriate phase every step as its action and the discrete signal performance metrics act as the reward (actually is the penalty) of the state-action pair; the long-run goal of the system is reducing the cumulative delay. The signal control agent coordinate with agents controlling its upstream and downstream intersections by sharing their state vectors with each other. A convolutional neural network (CNN) was used to build the 3DQN of the algorithm. The details of the algorithm are elaborated in this section.

5.2.1 State Representation

The state matrices are collected at every control step. In order for the controller to coordinate with other controllers, not only the condition of the intersection it controlled but also those of the upstream and downstream intersections are considered (3 intersections in total). Two aspects of the intersection condition are collected as the state: the current traffic state which the controller should "adaptive to" and the current signal phase.

The proposed algorithm uses the detailed location of every vehicle occupying the roadway within a certain distance from the stop line to represent the traffic state. It is important not to use the locations of ALL vehicles within the network. Since in the field, traffic cameras used in many ATSCs are only able to capture vehicles close to the intersection. The widely used "virtual loop" (see Figure 7) concept in video detection is applied as well. The length of the virtual loop detectors in this study is 15 feet and the maximum number of loop detectors for each lane is 20 (due to the length of turning storage bays, the number of "virtual loop" could be less than 20).

Then the algorithm converts the “virtual loop” actuations to a traffic state matrix. The traffic state matrices of all three intersections are stacked into one big matrix as an input of 3DQN.



0	0	1	1	1	1	1	0	0	0	0	0
0	0	0	0	1	1	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Traffic State Matrix Input

Vehicle Location Illustration

Figure 7 Traffic State Representation

The current activated signal phase is also recorded as a part of the *state*. The signal phase is defined as the combination of two or more non-conflicting vehicular movements. Figure 8 shows a typical eight-phase schema of a four-way intersection. An “interphase” refers to the clearance time including yellow time and all-red clearance. The current signal phase is coded as a vector with a length of n+1, where n is the number of phases of the signal. The last digit indicates whether the phase is interphase or not. Interphases are added between two phases as the clearance time including yellow time and all-red clearance. Note that the behavior of vehicles under yellow time has two phases depending on the “red percentage” defined. It indicates the percentage of yellow time the vehicles will consider as red light (Figure 9). Therefore, the representation of the current signal phase is coded as a vector with a length of n+1, where n is the number of phases of the signal. For example, when phase 1 is activated as green or the signal is not at the "red percentage" of the yellow time behind phase 1, the first element of the vector is set to 1 and all others are set to 0; when all-red phase was activated or the signal is at the "red percentage" of the yellow time, the last element is set to 1 and all others are set to 0. The traffic state matrices of all three intersections are also stacked into one big vector.

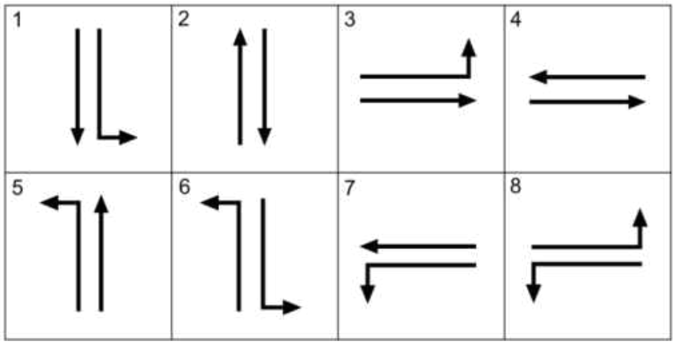


Figure 8 Eight phases for four-way intersections

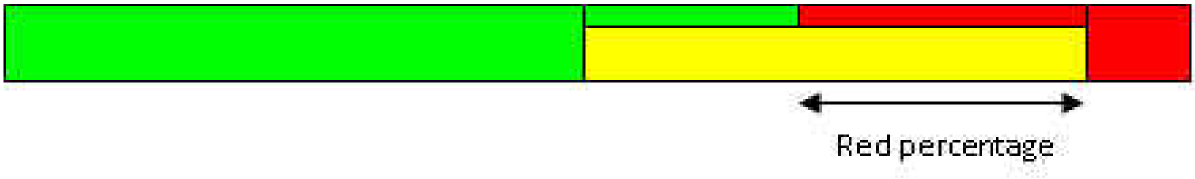


Figure 9 "Red Percentage" of the Yellow Time

5.2.2 Action Definition

The action set of the algorithm is all legal phases of the particular signal. When the signal is not in the "interphase", the algorithm selects an appropriate phase and send the phase to the controller. If a phase changing occurs, the controller will activate the "interphase" to clear the intersection. In addition, a 3-second minimum duration of each phase is enforced in consideration of the reaction time. Note that the sequence of the phases would be flexible, and it is not likely for the phases to form a traditional signal "cycle".

5.2.3 Reward Definition

For an RL problem, the reward should accumulate to the ultimate goal of the agent with a temporal discount. As for the adaptive signal controller, the goals could be minimizing the travel time of a vehicle, or the total delay, which theoretically is defined as the difference between the actual and expected travel times. However, it is unfeasible to get the travel time or total delay every discrete control step (e.g. every one second) since those measures are only available when the vehicle reaches its destination (e.g. every several minutes). Therefore, the cumulative waiting time of the vehicles in the queue (the dominant part of total delay), which could be monitored every control step, rather than the total delay was utilized as the goal indicator. And the reward is defined as the difference between the current and previous waiting times of all vehicles:

$$r_t = -(W_{t+1} - W_t) \tag{17}$$

where W_{t+1}, W_t are the waiting time of step $t+1$ and t. When the vehicle is queued, the agent will be penalized; when the vehicle waiting in the queue is discharged, the agent will be rewarded.

However, getting the cumulative waiting time of every vehicle within a relatively large road network is computationally expensive. Hence, the reward is estimated for each step directly:

$$r_t \ = wn_t^d - n_t^q t_s \tag{18}$$

where n_t^q is the number of the queued vehicles at step t; t_s is the step length; n_t^d is the number of the vehicles discharged during step t; w is an average waiting time of discharged vehicles.

5.2.4 Deep Q Network Structure

As mentioned in the last section, DQN uses DNNs as the functional approximator. The input of the DNN is the state and the output of it is the action. To extract valuable information from the big traffic state matrix, convolutional neural network (CNN), which is widely used in many pattern recognition problems such as image-processing, is used in this algorithm.

CNN is a deep neural network composed by a sequence of three kind of layers: convolutional layer, pooling layer and fully-connected layer. Convolutional layer acts a “window” “scanning” across the big input matrix and extracts information of a local region; pooling layer performs a down sampling operation and fully-connected layer is a regular neural network layer which is typically one of the last few layers. Compared to the regular neural network, the CNN takes advantage of local spatial coherence in the input which allows it to have much fewer parameters, and therefore overcomes the overfitting issue and saves computational resources.

The structure of CNN used in the proposed algorithm is shown in Figure 10. The traffic state matrix is filtered into a vector by two convolutional layers and two pooling layers. Then it is combined with the signal phase state vector as the input of the fully-connected layer. The output vector of the fully connected layer is used to estimate state value and the advantage of all actions. Finally, they are added up to get the Q-value. The Leaky Rectifier Nonlinearity Units (Leaky ReLU) (Maas et al., 2013) was applied as the activation function. Noticed that the size of layers varies for different intersections due to the different input sizes.

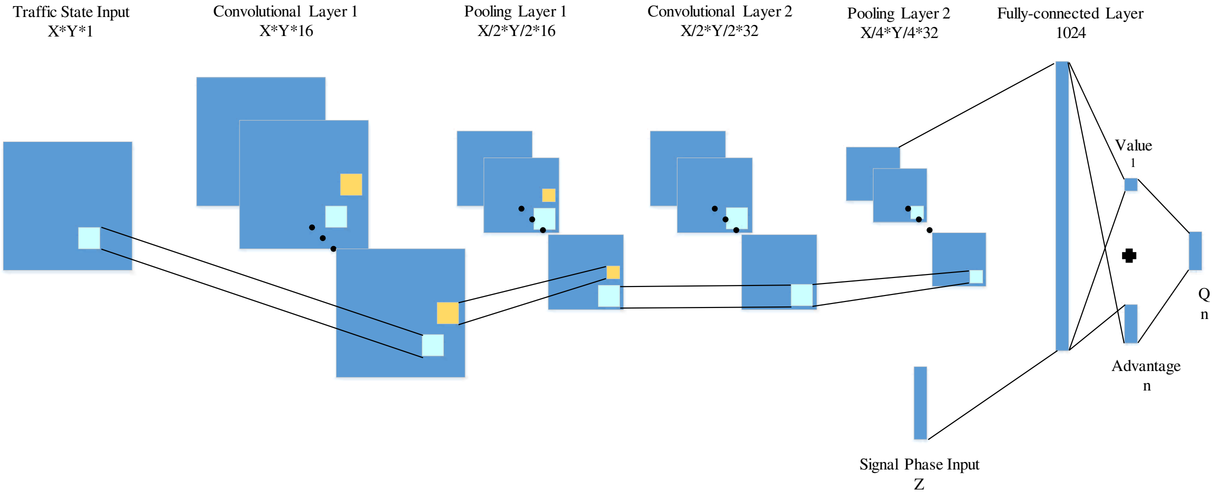


Figure 10 Structure of the CNN used in the Algorithm

5.2.5 Supervised Pre-Training

In order to speed up the training process, the parameters of DNN θ are initialized by a supervised learning process instead of random initialization and exploration. During the pre-training, the algorithm is forced to record the policy of a reasonable fixed-timing traffic signal, then updates θ based on learned policy. The main difference between the pre-training and training are: (1) the agent observes the action taken by the fixed-timing traffic signal rather than execute its own; (2) the reward is set to a constant value in order to update the θ.

5.2.6 Overall Algorithm

The pseudocode of the proposed algorithm is summarized in Algorithm 1. Note that during training, the number of steps to update the target network (frozen period) is increasing by episodes to ensure the convergence. The algorithm is coded by Python programming language using deep learning modules Tensorflow (Abadi et al., 2016)

Algorithm 1 3DQN for Decentralized Adaptive Signal Control

```
Input: discount factor $\gamma$, greedy $\epsilon_e$, replay memory size $M$, minibatch size $B$,
number of episodes $N$, number of the steps in an episode $T$, starting number of steps to
update target network $N_{us}$; target network updating increment $N_{ui}$
Initialize primary network with pre-trained weights $\theta$;
Initialize target network with weights $\theta^- = \theta$;
for episode = 1 to $N$ do :
    Initialize state $s$ as with the starting scenario at the intersection;
    Initialize state $a$;
    Initialize the reply memory $m$ as empty;
    Initialize number of steps to update target network $N_u$: $N_u \leftarrow N_{us}$;;
    for step = 1 to $T$ do
        The agent choose an action $a$ based on $\epsilon$ greedy;
        The agent observe the current signal phase $p$;
        if $p$ is an interphase then
            Assign action $a'$ of a lastest step with non-interphase to $a$: $a \leftarrow a'$;
        else if $a == p$ then
            Keep signal unchanged;
        else
            Determine and activate the appropriate interphase;
        end if
        The agent observes the reward $r$ and current state $s'$;
        Store the experience $(s, a, r, s')$ into $m$;
        Assign $s'$ to $s$: $s \leftarrow s'$;
        if $|m| > B$ then
            Randomly sample $B$ experiences from $m$;
            Calculate the loss $J$ by formula (3) (4);
            Update $\theta$ by $\nabla J$ using Adam (Kingma & Ba, 2014) back propagation algorithm;
            if step % $N_u == 0$ then
                $\theta^- = \theta$;
            end if
        end if
    end for
    Update the number of steps to update target network $N_u$: $N_u = N_u + N_{ui}$;
end for
```

5.3. Experiment and Evaluation

The proposed algorithm was implemented in a simulated traffic network using a commercial traffic simulator Aimsun Next 8.2.3. The simulator provides a Python application programming interface (API) to get access to the coded ATSC algorithm. The algorithm gets the information from the simulated traffic and implements its control command to the simulated signal controllers. The simulated RL signal control agents were trained for certain episodes. After the training, its performance is evaluated by the real-world signal timings.

5.3.1 Simulation Set Up

To better examine the performance of the algorithm, the simulation scenario was built based on a real-world suburban traffic corridor in Seminole County, Florida. There is one limited-access freeway (SR 417) and several parallel or connected arterials (SR 426, Red Bug Lake Road, Slavia Road, West Chapman Road, and Dean Road) within the corridor. Figure 11 shows the location of the corridor and Figure 12 illustrates the high-level abstract of the corridor including nodes IDs (signalized intersections, Origin/Destination centroids and ramp connections), connectivity between nodes and the length in miles of the links. The authors did their best effort to match the geometric design of simulation corridor with that in the real-world, although some inevitable minor modifications were done to fit in the available demand and traffic data.

Along with the corridor, there are eight signals in total under control of the proposed algorithm. The detailed information about the number of lanes of each approach, whether there are dedicated left-turning or right-turning lanes and the phase configuration is provided in Table 5. Note that there are four signals among them which are connected with the on/off-ramp of the freeway, and therefore interact with the freeway traffic flow. All eight intersections are using coordinated actuated signal controllers and their timings are provided by Seminole County. These actuated signal controllers are used as the benchmark to evaluate the performance of the proposed algorithm. In addition, the length of the "interphases" used in the algorithm for each signal was the same as the summation of yellow and red clearance time of the real-world signal timing.

Table 5 Information Regarding Signals under Control of the Algorithm

Intersection ID	Connected with Ramp	Lanes of Approaches	Number of Phases	Legal Movement of Phases	Aberrations
1	Yes	SB: 1*LR+1*R WB: 1*L+2*T EB: 3*T	3	ST (SL+SR) WL+WT ET+WT	Approaches: NB: Northbound SB: Southbound WB: Westbound EB: Eastbound Movement of a lane: T: Trough L: Left turning R: Right turning LR: Shared left-right turning Legal Movement of a Phase: First digit: Approach Second digit: Turning Movement
2	Yes	NB: 2*L+2*R WB: 3*T EB: 1*L+2*T	3	NT (NL+NR) ET+WT EL+ET	
3	No	NB: 2*L+1*R WB: 2*L+2*T EB: 2*T+1*R	3	NT (NL+NR) WL+WT ET+WT	
4	No	NB: 2*T+1*R SB: 2*L+2*T WB: 2*L+1*R	3	NT+ST SL+ST WT(WL+WR)	
5	No	NB: 1*L+2*T SB: 2*T EB: 1*L+1*R	3	NL+NT ST+SL ET(EL+ER)	
6	No	NB: 2*L+2*T+1*R SB: 2*L+2*T+1*R WB: 2*L+2*T+1*R EB: 2*L+2*T+1*R	4	NL+NT SL+ST WL+WT EL+ET	
7	Yes	NB: 2*L+1*R WB: 3*T EB: 1*L+3*T	3	NT (NL+NR) WT+WL WT+ET	
8	Yes	SB: 2*L+1*R WB: 3*T EB: 3*T	2	ST (SL+SR) WT+ET	

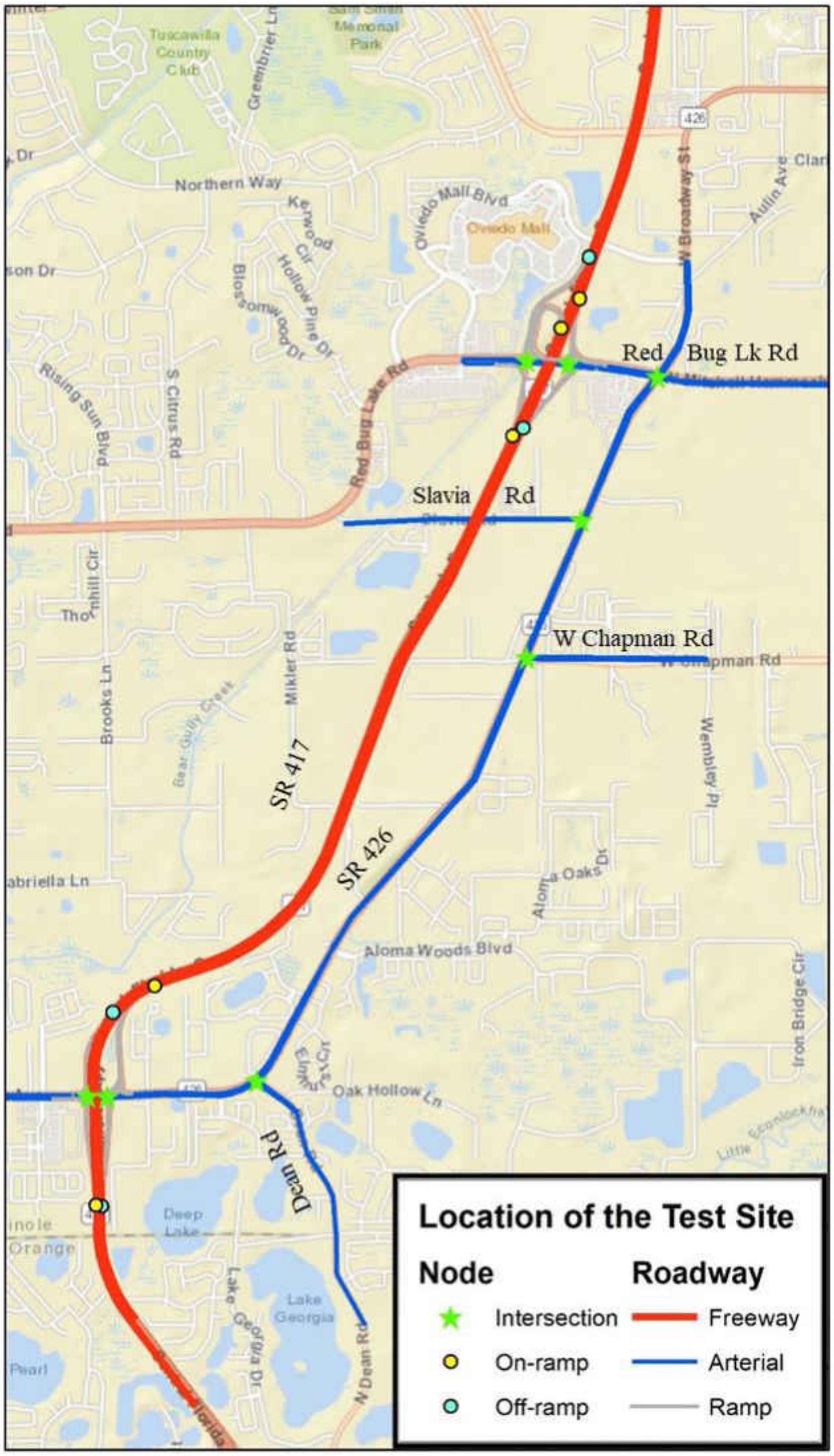


Figure 11 Location of the Simulation Test Site

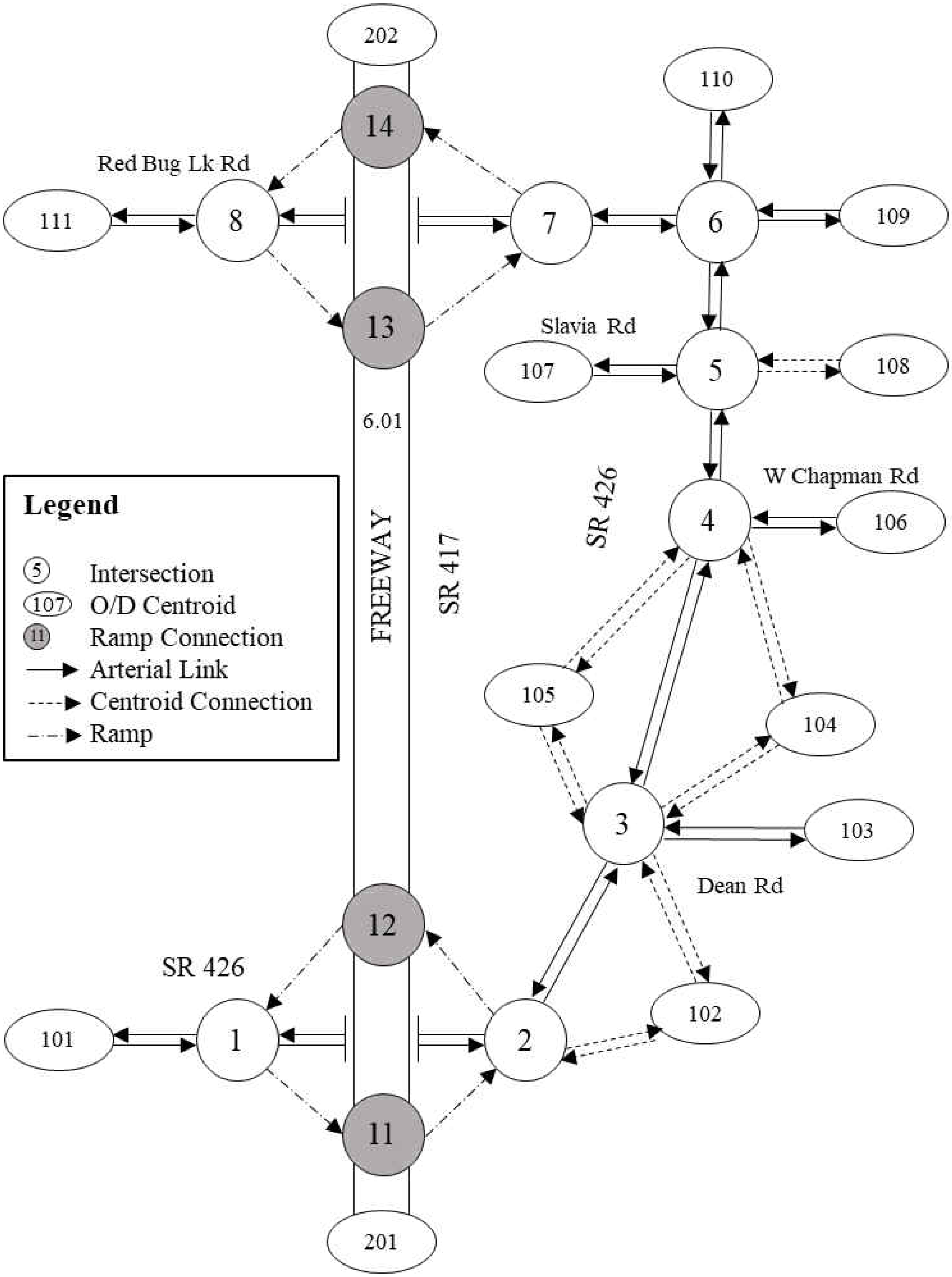


Figure 12 High-Level Abstract of the Test Corridor

In this study, the demand data of the AM peak hours (7:00-9:00) were used as the input to simulate the recurrent congestion situation. The daily Origin-Destination (O/D) matrices were extracted from Orlando Urban Area Transportation Study (OUATS) with the base year 2009, which is the most recent regional planning model available when the authors conducted the study. The 15-minute aggregated real traffic count data of January 18th, 2018 were utilized to estimate peak-hour O/D matrix from the raw daily O/D matrices. The real dataset was extracted from Microwave Vehicle Detection System where (MVDS) and Automated Traffic Signal Performance Measures (ATSPM) for the freeway and arterials, respectively. The two-hour aggregated matrix is provided in Table 7 for the readers' reference. Static origin-destination and departure adjustment were applied to convert the two-hour O/D matrix to eight 15-minutes time-dependent matrices based on the aforementioned real dataset. The simulation scenario is further calibrated by the traffic counts from the real dataset. One of the most important calibration metrics, the root mean square error between simulated counts and the real counts, is 4.7 vehicles per hour, which means the scenario is well calibrated.

5.3.2 Training

The algorithm is trained in episodes. One episode is one simulation replication with the aforementioned two-hour AM congested traffic. The goal of the algorithm is to minimize the total delay of the episode. The length of training and control step is one second. Therefore, there are 7200 training steps in one episode. The simulation replications were warmed up by 15-min real-world demand (6:45-7:00). To reduce the overfitting issue, even though the traffic demand of every episode is the same, its random state was set differently.

The signal timing used in the pre-training was generated by the maximum green time of the real-world actuated signal timing. The initial and fined-tuned input hyper-parameters of the algorithm are shown in Table 7

Table 6 Two-hour Aggregated O/D Matrix

O\D	101	102	103	104	105	106	107	108	109	110	111	201	202	Total
101	0.00	0.00	0.00	149.98	157.33	226.87	0.00	84.43	90.20	45.75	23.93	839.48	297.32	1915.28
102	0.00	0.00	4.18	2.19	4.25	15.01	0.00	22.63	1.87	1.49	1.32	0.00	24.27	77.21
103	0.00	1.94	0.00	1.05	2.11	0.00	0.00	11.75	0.00	0.00	0.00	0.00	0.00	16.84
104	121.92	1.01	9.91	0.00	0.72	5.81	16.21	6.26	12.12	24.46	0.00	122.68	11.10	332.21
105	131.80	2.07	2.19	0.76	0.00	0.00	101.12	14.41	0.00	23.25	0.00	239.74	92.57	607.91
106	241.04	20.30	10.48	7.42	0.00	0.00	219.88	0.00	0.00	8.75	0.00	0.00	129.16	637.03
107	0.00	0.00	0.00	16.07	64.06	127.57	0.00	133.32	0.00	0.00	0.00	0.00	0.00	341.01
108	139.11	3.09	4.33	1.79	3.93	0.00	53.76	0.00	0.00	6.11	0.00	324.96	39.59	576.67
109	184.33	82.21	58.60	24.22	0.00	0.00	0.00	0.00	0.00	20.96	658.66	468.06	444.42	1941.45
110	38.31	37.93	43.73	34.32	30.96	7.77	0.00	20.77	47.63	0.00	97.76	104.18	81.88	545.26
111	15.76	0.00	0.00	0.00	0.00	0.00	0.00	0.00	193.43	68.33	0.00	71.30	238.01	586.83
201	516.35	42.32	0.00	37.07	53.08	0.00	0.00	141.52	146.68	112.16	101.13	0.00	1782.95	2933.27
202	291.61	0.00	0.00	29.38	71.51	91.16	0.00	115.26	188.36	68.13	274.68	1534.16	0.00	2664.24
Total	1680.23	190.87	133.43	304.25	387.95	474.20	390.97	550.34	680.27	379.39	1157.47	3704.56	3141.26	13175.20

Table 7 Parameters Used in Algorithm

Parameter	Initial Value	Fine-tuned Value
Discount factor γ	0.99	0.999
Ending greedy ϵ_e	0.1	0.01
Greedy decrement ϵ' (if applicable)	0.001	0.0001
Replay memory size M	2000	20000
Minibatch size B	64	64
Number of episodes N	20	20
Number of the steps in an episode T	7200	7200
Starting number of steps to update target network N_{us}	200	200
Target network updating increment N_{ui}	0	200
Learning rate α	0.001	0.0001
Leaky ReLU β	0.01	0.01

Figure 13 visualizes the average waiting time per vehicle per mile for each training episode. The training episode 0 indicates the value of the pre-training. The curve shows that the average waiting time dropped dramatically after the first training episode and it was continuously dropping until the 9th episode. During the last 10 episodes, the average waiting time fluctuates in a very small range which indicates the optimal policy was learned.

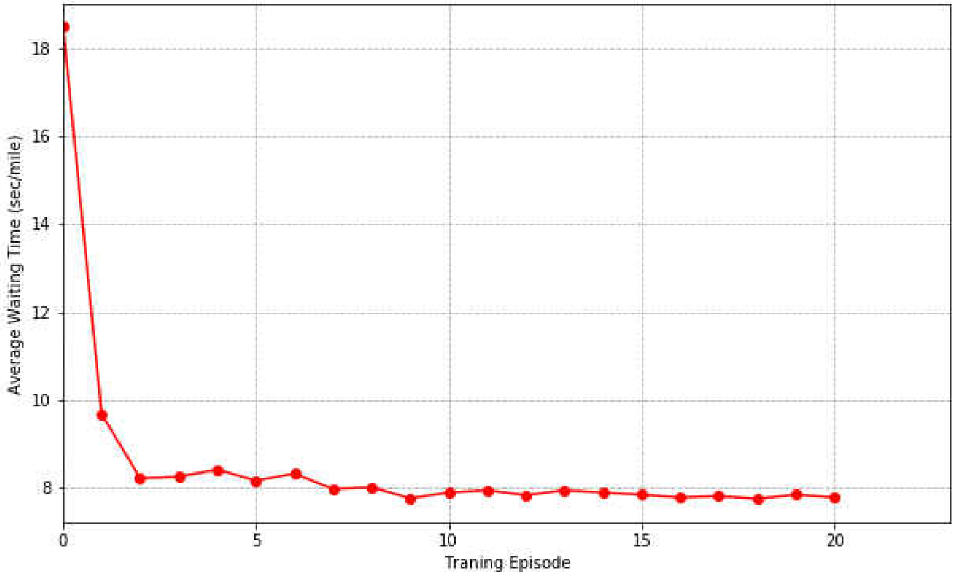


Figure 13 Location of the Simulation Test Site

5.3.3 Training Evaluation Results and Discussion

To evaluate the performance of the proposed ATSC algorithm, both the well-trained algorithm and the real-world benchmark signal control were implemented in a simulation replication with the same random state. Three kinds of performance measures were observed in 5-min aggregation intervals: average travel time per vehicle of all the vehicles traveling in the road network; average total delay (the difference between the actual and expected travel times) per vehicle per mile of the whole network. Figure 16 shows the performance measures by simulation time.

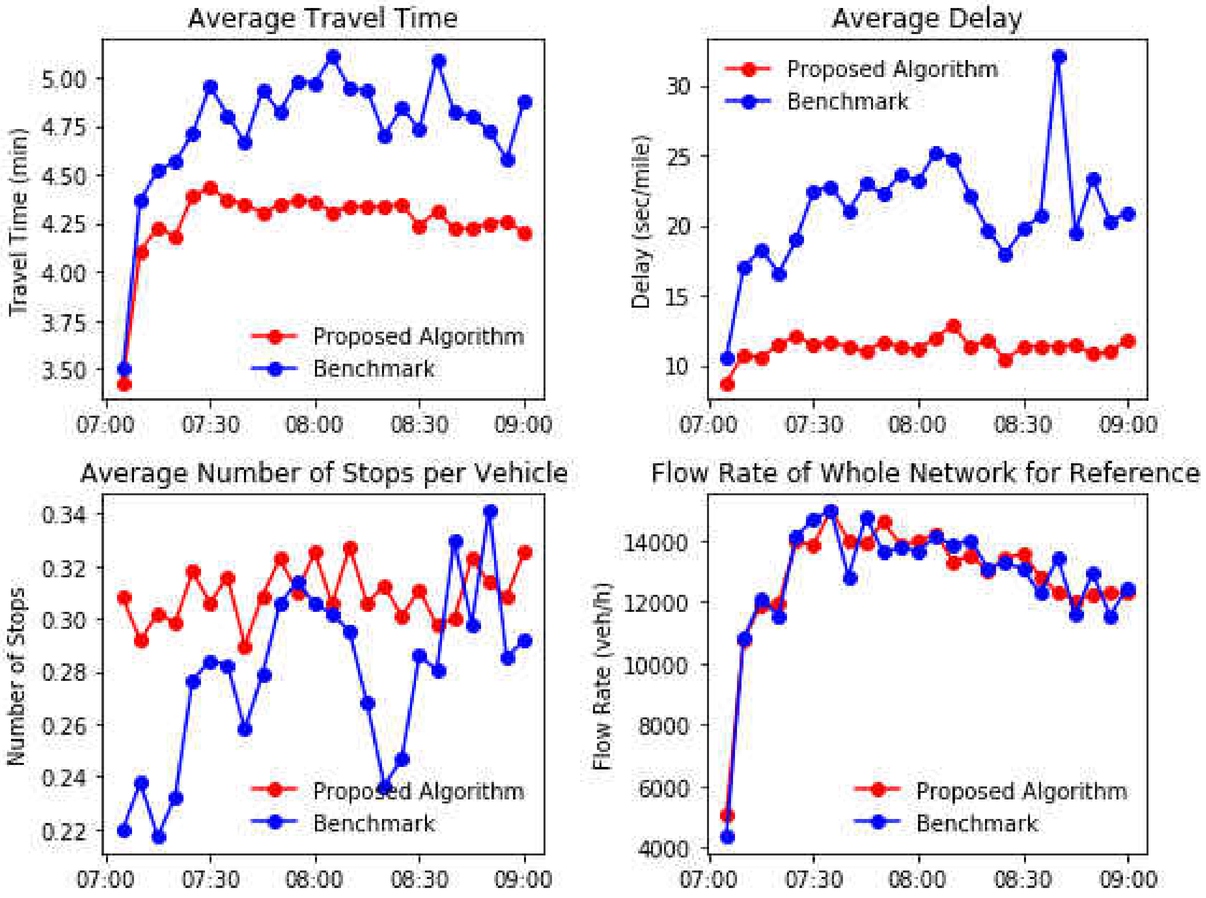


Figure 14 Performance of Proposed ATSC algorithm and Benchmark

During the whole simulation episode, the average travel time and average delay of the simulated network controlled by the proposed algorithm are less than those of the network controlled by the benchmark signal. On average, the proposed algorithm reduced 10.27% (4.26 minutes versus 4.75 minutes) of travel time and 46.46% (11.27 second/mile versus 21.06 second/mile) of the total delay. According to Figure 14, at the beginning of the simulation, there is no significant difference of performance between the two signal controls. However, as the time goes by, on the one hand, the performance of the benchmark is getting worse than that of the proposed algorithm. On the other hand, the performance of the benchmark fluctuates a lot while that of the proposed algorithm remains stable. One possible reason is that at the beginning of the AM peak period the traffic volume remains at a low level and then it fluctuates a lot (see flow rate in Figure 14). This confirms that the algorithm well adapts to the changing traffic demand.

However, given that the number of the stops remains at a low level, the proposed ATSC algorithm tends to increase the number of stops by 11.29% on average (0.31 versus 0.28). Interestingly, compared with the benchmark, the number of the stops remains at a relatively stable level when the network is controlled by the proposed algorithm. In previous studies, only Li et al. (L. Li et al., 2016b) evaluated the specific performance metrics but it was compared with the signal control based on conventional RL algorithm incorporating with a shallow neural network. Thus, it is not appropriate to use the result as the guidance of the field application. According to the careful examination of the algorithm during the simulation process, the length of the phases allowing through movement of major approaches are typically shorter than the benchmark. This ensures the fair travelling rights between major and minor approaches as well as through and turning movements by reducing the waiting time of vehicles queueing the minor approaches or left-turning bay. However, the side effect is that the vehicles travelling through the major approaches are forced to stop to yield the right of way. Although the result is always reasonable since reducing the number of stops is never the goal the algorithm. This trade-off should be taken into account by practitioners.

5.4. Conclusion

This chapter proposes a decentralized adaptive signal control algorithm, in the context of network-level control, using one of the famous deep reinforcement learning methods, double dueling deep Q network. In the algorithm, for every individual intersection, a reinforcement learning agent controls the signals based on high-resolution real-time traffic data. It captures detailed locations of vehicles as the input, extracts relevant information by convolutional neural networks, and selects appropriate signal phases every second to reduce vehicles' waiting time. To achieve better network performance, control agents are coordinated with each other by sharing the information regarding the traffic and signal state.

The proposed algorithm was trained and evaluated in a simulated real-world suburban traffic corridor with eight signalized intersections in Seminole County, Florida. Travel demand of AM peak period was used in the simulation for the algorithm to reduce the recurrent congestion. The performance of the well-trained algorithm was compared with the real-world coordinated actuated signals of the corridor provided by the local jurisdiction. The evaluation results showed that the algorithm adapts well to the changing traffic demand. And it reduces 10.27% of network travel time and 46.46% of network delay compared with the real-world benchmark.

CHAPTER 6: SATETY-ORIENTED ADAPTIVE TRAFFIC SIGNAL CONTROL WITH MULTI-OBJECTIVE REINFORCEMENT LEARNING

6.1 Introduction

As one of the most important Active Traffic Management (ATM) strategies, Adaptive Traffic Signal Control (ATSC) helps improve traffic efficiency of signalized arterials and urban roads by adjusting the signal timing in response to the dynamic traffic demand. However, the safety effects of state-of-practice ATSCs are not consistent. Some studies show that the installation of ATSCs reduces the number of crashes (Jiaqi et al., 2016; Khattak et al., 2018). While another study concludes that the crash frequency before and after the implementation of ATSCs is not significantly different (Fink et al., 2016). A study on traffic conflicts, which is one of the surrogate safety measures, even found that there is a considerable increase in both frequency and severity of conflicts following the installation of the ATSC (Tageldin et al., 2014). The mixed evidence raises the concern of ATSC's safety impact.

This study advocates designing an ATSC that is able to ensure or improve traffic safety, i.e., a safety-oriented ATSC. Several recent studies have found that signal timing is related to the crash occurrence at signalized arterials and intersections (Yuan et al., 2018, 2019b; Yuan & Abdel-Aty, 2018). By applying the models developed by the aforementioned studies, the safety-oriented ATSC is able to reduce the crash occurrence by dynamically optimizing its signal timing in response to different traffic conditions. Moreover, the proposed safety-oriented ATSC could also serve as a strategy of pro-active traffic safety management to improve traffic safety of arterials and urban roads like other ATM strategies (e.g. Ramp Metering and Variable Speed Limits) do for freeways (Abdel-Aty et al., 2006; L. Wang et al., 2017; Yu & Abdel-Aty, 2014).

Although the safety-oriented ATSC aims at optimizing traffic safety, it should not be detriment to efficiency. Therefore, we proposed an ATSC system that utilizes a multi-objective framework to simultaneously optimize traffic efficiency and safety. A real-time crash risk model (Abdel-Aty et al., 2004) is applied to generate the indicator of near-future crash likelihood. The multi-objective reinforcement learning algorithm is used for optimization. The proposed algorithm was tested in a simulated real-world isolated intersection. Its performance in terms of delay and crash risk reduction was compared with a replicated field controller and an ordinary ATSC optimizing only traffic efficiency. A discussion about the control policy of the proposed safety-oriented ATSC and the potential impact of different signal configuration is also provided.

6.2 Background: Real-Time Crash Risk Models

The first step of optimizing the signal timing for traffic safety is understanding how they are correlated. In the pro-active perspective, the impact of a specific signal timing on the future crash potential needs to be quantified. Recently, researchers (Yuan et al., 2018, 2019b; Yuan & Abdel-Aty, 2018) have proposed real-time crash risk models to examine the relationships between future crash potential and traffic conditions including signal timing. The basic assumption underlying real-time crash risk models is that there exist certain conditions that are relatively more "crash-prone" than the others, which could be called "crash precursors". For example, conditions that are just before the crash occurrence would be regarded as "crash condition". By comparing the characteristics of "crash conditions" with "non-crash conditions", crash precursors could be identified. Like other binary classification models, the output of real-time risk models indicates the forecasted crash potential. It could be the probability of the crash or the odds of crash versus non-crash. Similar to the efficiency measures like delay, the forecasted crash potential could be directly employed by “predictive/pro-active” controllers to assess the future safety effect of a candidate signal timing in real-time.

6.3 Algorithm

In this study, the control problem is formulated into the MORL setting: the signal controller acts as an RL *agent*; it observes the traffic condition of the intersection and the current signal status as the *state*; it directly selects the appropriate phase as its *action*; the waiting time of vehicles acts as the efficiency *reward* while the risk score derived from the real-time crash risk model acts as the safety *reward*; and the *goals* of the agents are reducing the delay (efficiency) and future crash potential (safety). Weighted sum approach is selected to develop a single policy MORL and one of the famous deep reinforcement learning algorithms Double Dueling Deep Q Network (3DQN) is utilized as the backend learning algorithm. The details of the algorithm are elaborated on in the subsections below.

6.3.1 State

The state used in this chapter is the same as the ones used in Chapter 5.

6.3.2 Action

The *action* of the agent is selecting the appropriate signal phase at each time interval based on the current *policy*. If a phase changing occurs, the controller will activate the interphase to clear the intersection.

Several other rules are applied to restrict the arbitrary selection of actions to ensure traffic safety and overcome some fundamental limitations of RL-based ATSC:

1) *Ensure minimum green time* (g_{min}): the minimum green time concept is used in actuated signal control to satisfy the driver's expectation (Arroyo et al., 2015). If a minimum green time is set too low (or even omitted) and violates the driver's expectation, there exists a risk of increased rear-end crashes. Therefore, the controller is configured not to allow the change phase if the minimum green time is not satisfied. The values of minimum green times are set to be the same as the ones used in the field to avoid double investigation. However, if there is no existing signal control, the values should be set based on the local traffic signal timing manual.

2) *Default phase* (p_0): If there is no vehicle at the intersection, theoretically the RL-based ATSC randomly selects a phase to activate during the learning stage. This is detrimental to both traffic efficiency and safety. Therefore, a default phase that represents the major approach through movements is set to avoid the random phase changes.

3) *Maximum allowed waiting time* ($t_{maxwaiting}$)*:* The benefit of setting a maximum allowed waiting time is ensuring fair travel rights. Consider an extreme case. There is only one vehicle waiting on the minor approach to turn left, while there are one hundred vehicles that are waiting on the major approach going through. As one of the objectives of ATSC is reducing the TOTAL delay, the controller would favor clearing the major approach, which results in

excessively long waiting time for the vehicle on the minor approach. Therefore, a maximum allowed waiting time is configured to prevent the occurrence of such a situation.

6.3.3 Reward

Two rewards are designed for traffic efficiency and safety. As for the efficiency, the goal is minimizing the travel time of a vehicle, or the delay, which theoretically is the difference between the actual and expected travel time. However, it is not feasible to obtain the travel time or delay as they are only available when the vehicle reaches its destination. Thus, the cumulative waiting time of the queued vehicles is used as the goal indicator. The reward representing the traffic efficiency is defined as the difference of the current and previous cumulative waiting time of all vehicles.

As for safety, a risk score is utilized to indicate the relative risk level. The score is calculated using a real time crash risk model calibrated based on the local historical crashes and traffic data. As the score is site-specific, its calculation process varies for different kinds of intersections, different locations, and different interests of the users. Other surrogate safety measures that imply the future crash risk could also be used as the "risk score".

In this study, the reward for safety is defined as the adjusted risk score by a baseline:

$$r_{ts} = \begin{cases} riskscore_t - riskscore_{base} & when\ riskscore\ is\ generated \\ 0\ \ otherwise \end{cases} \tag{19}$$

where $\boldsymbol{riskscore_t}$ is the risk score at timestamp *t* and $\boldsymbol{riskscore_{base}}$ is a baseline risk score calculated during the pre-training. It should be noted that the risk score might not be generated at every control step. In this case, the risk reward acts as a "delay" reward.

The reward could be interpreted as an “advantage”: when the reward is positive, the safety performance is better than the baseline, which means that the agent will be rewarded; otherwise, the agent will be penalized. Defining the reward as an advantage term accelerates the learning process as it helps the agent to find the direction.

It should be also noted that the rewards for traffic efficiency and safety are used to direct the training process. Once the control agent is well-trained, such rewards are no longer needed during the operation.

6.3.4 Weighted Sum Approach for Single Policy MORL

Weighted sum approach (Karlsson, 1997), one of the single policy MORL algorithms, is selected due to its computational efficiency since the multi-policy algorithms are computational intractable for the ATSC problems. It computes a linearly weighted sum of Q-values for all the objectives to obtain a synthetic Q function:

$$SQ(s,a) = \sum_{i=1}^{N} w_i Q_i(s,a) \tag{20}$$

where $SQ(s,a)$ is the synthetic Q value; $Q_i(s,a)$ is the Q value of the *i*th objective; w_i is the weight, it implies the relative importance of the specific *i*th objective. The weights could be pre-configured by the algorithm developers or be determined by the users.

In this study, the synthetic Q value is the weighted sum of the normalized Q values:

$$Q(s,a) = w_e \frac{Q_e(s,a) - Q_{e,min}(s,a)}{Q_{e,max}(s,a) - Q_{e,min}(s,a)} + w_s \frac{Q_S(s,a) - Q_{s,min}(s,a)}{Q_{s,max}(s,a) - Q_{s,min}(s,a)} \tag{21}$$

where $Q(s,a)$ is the synthetic Q value used to evaluate the actions; $Q_e(s,a)$ and $Q_s(s,a)$ is the Q value of traffic efficiency and traffic safety, respectively. Two Q values are normalized to the same magnitude by the min-max method since the rewards and Q values of the two objectives have a huge difference in terms of the magnitude. $Q_{i,min}(s,a)$ and $Q_{s,min}(s,a),\ i \in [e,s]$ are the minimum and maximum values of the two Q values estimated from the pre-learning. w_e and w_s are the weights.

6.3.5 Backend Learning Algorithm

The proposed ATSC utilizes Double Dueling Deep Q Network (3DQN), one of the advanced Q-value-based deep learning algorithms, as its backend algorithm. 3DQN uses DNNs as its functional approximator. In this study, the convolutional neural network (CNN), one of the DNNs widely used in pattern recognition, is employed to construct the functional approximator. The structure of CNN used in the proposed algorithm is similar to the one used in the Chapter 5. CNN takes the state as the input and outputs the Q values of all actions. It should be noted that the functional approximators of two Q values have the exact same structure but are trained separately.

6.3.6 Pre-Training

According to the design of the algorithm, there are two pre-requests for learning: first, getting the baseline risk score to derive the safety reward; second, getting the estimated range of the Q values to obtain normalized Q-values. Therefore, a two-phase pre-training is designed.

The objective of the first phase of the pre-training is to find out the baseline risk score. Since the risk score is a relative measure and site-specific, it is impossible to find a “best” risk score. Thus, the hourly-average risk score of a benchmark scenario, which is used to evaluate the proposed algorithm, is utilized. It means that the control agent is “directed” to perform better than the benchmark signal controller does.

To estimate the range of Q-values, at the second phase of the pre-training, the control agent is asked to learn the exact policy of the benchmark signal controller or another reference controller if necessary. During the pre-learning, the agent observes the action taken by the benchmark controller rather than taking actions based upon its own policy and update the hyper-parameters of the functional approximator. The minimum and maximum of Q values generated by functional approximator during the learning course are recorded to get the estimated range of the Q values. It should be noted that as the baseline controller is not exactly the same as the optimal controller, the range of Q values of those two controllers might have a subtle difference. Therefore, there exists a dilemma that while it is impossible to know the range of Q values of the optimal policy before learning, without knowing the range of the Q values, it is impossible for the MORL agent to learn the optimal policy. Thus, a compromise could be achieved by using the reference controller if the baseline controller is not close enough to the optimal controller.

6.3.7 Overall Algorithm

The flow chart of the proposed algorithm is presented in Figure 15. The algorithm is coded by Python programming language using deep learning package Tensorflow (Abadi et al., 2016).

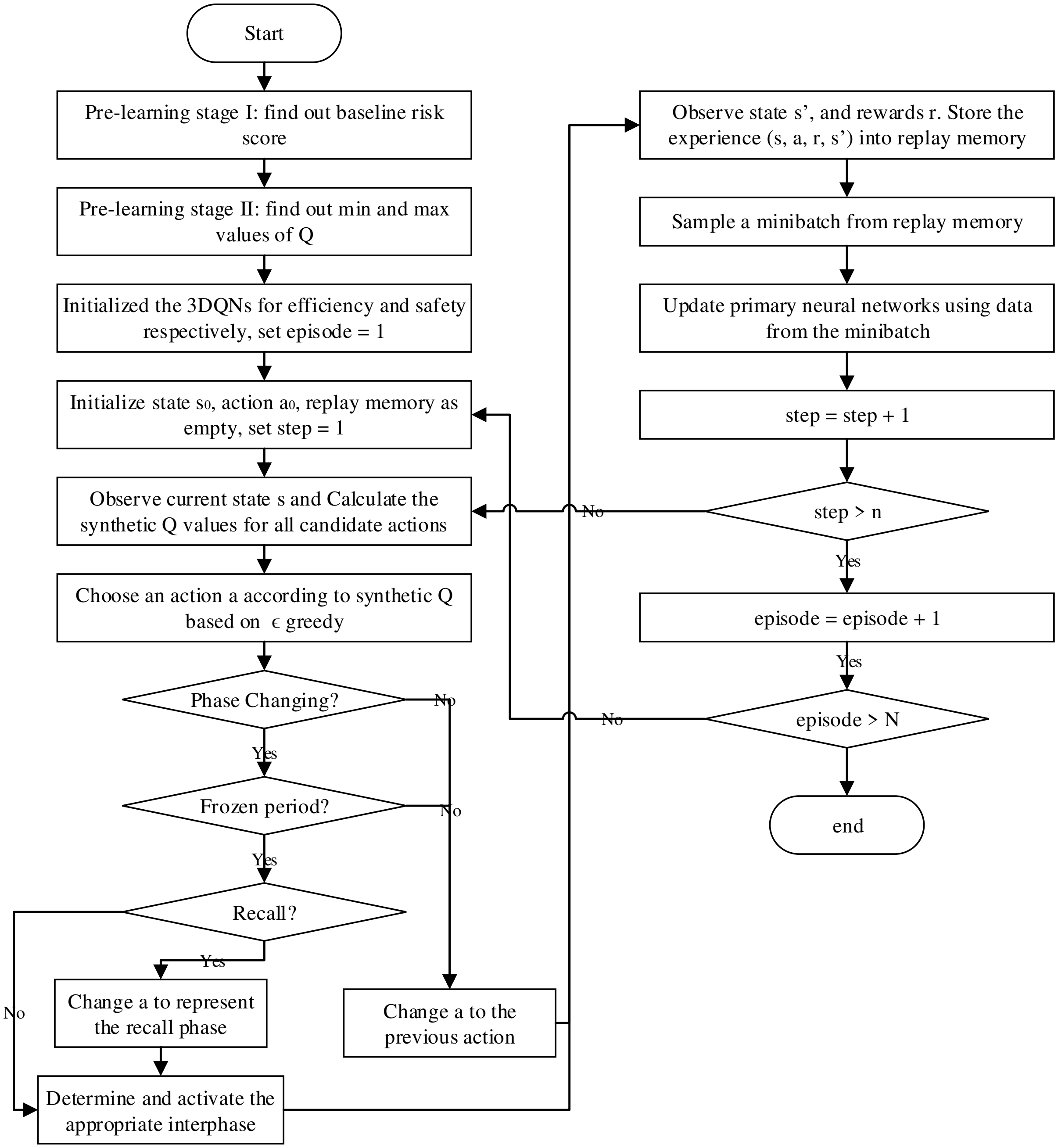

Start
Pre-learning stage I: find out baseline risk score
Pre-learning stage II: find out min and max values of Q
Initialized the 3DQNs for efficiency and safety respectively, set episode = 1
Initialize state s0, action a0, replay memory as empty, set step = 1
Observe current state s and Calculate the synthetic Q values for all candidate actions
Choose an action a according to synthetic Q based on ϵ greedy
Phase Changing?
No
Yes
Frozen period?
No
Yes
Recall?
Yes
No
Change a to represent the recall phase
Change a to the previous action
Determine and activate the appropriate interphase
Observe state s', and rewards r. Store the experience (s, a, r, s') into replay memory
Sample a minibatch from replay memory
Update primary neural networks using data from the minibatch
step = step + 1
step > n
No
Yes
episode = episode + 1
Yes
episode > N
No
end

Figure 15 ATSC Algorithm flow chart

6.4 Case Study

The proposed algorithm was tested in a simulated isolated intersection using a commercial traffic simulator Aimsun Next 8.3.0. The algorithm obtains the information from the simulator and implements its control policy to the simulated signal controllers. The simulated MORL agents were trained extensively. The performance of the well-trained agent is evaluated by the real-world signal timings and compared with an RL based ATSC optimizing only traffic efficiency (ATSC-SORL).

6.4.1 Simulation Set Up

The simulation scenario was built based on a real-world signalized intersection of North French Avenue (major arterial) and West 1st Street (minor arterial) in Seminole County, Florida. The intersection is a typical mid-size four-way intersection with moderate traffic volume. Figure 18 shows the lane configuration of the approaches of the intersection. Right turns are permitted on red after a complete stop at the stop line, and the left-turns of the east-west approaches are configured as permitted-protected. In this study, the lane-based counts of all Tuesdays, Wednesdays, and Thursdays from January 2018 to March 2018 were extracted from the Automated Traffic Signal Performance Measures (ATSPM) system. Then the average counts for every 15 minutes are used to approximate the turning movement counts for a “normal weekday”. Then they serve as the travel demands of the intersection. It should be noted that for the shared through-right-turning lane, the percentage of right turning is set as 30%. As for the eastbound, as there exists a dedicated right-turning lane, the actual right turning volume is used.

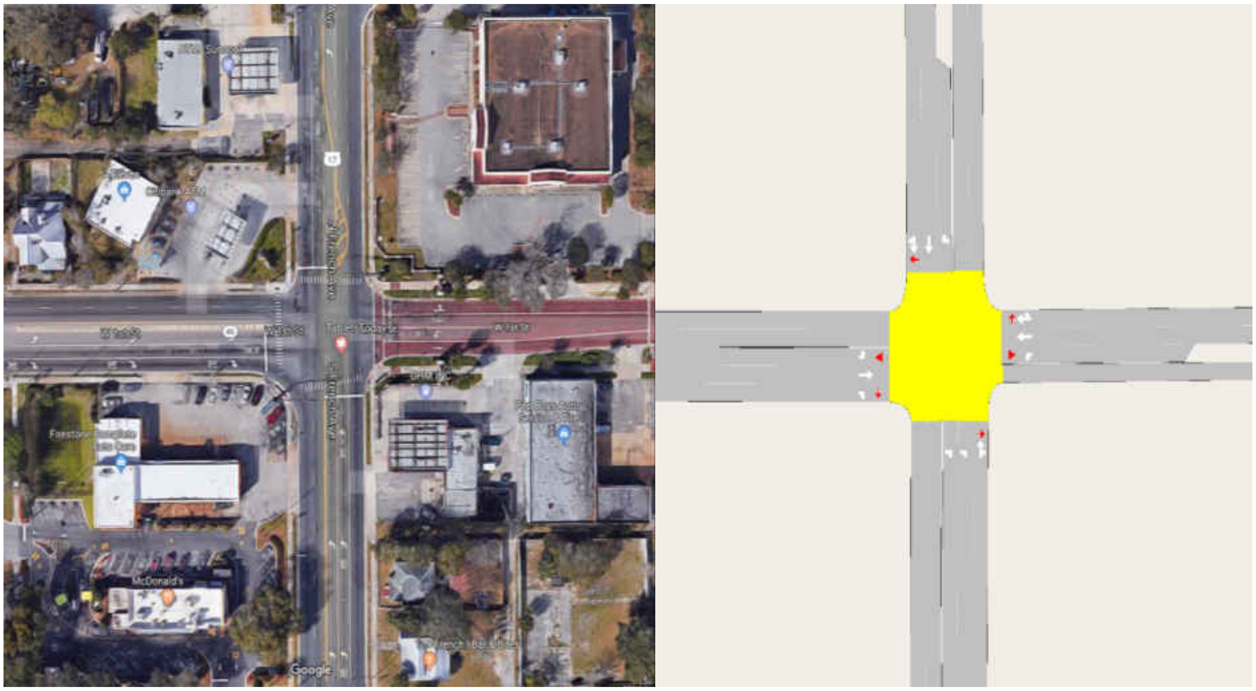


Figure 16 Lane configuration of the simulated intersection

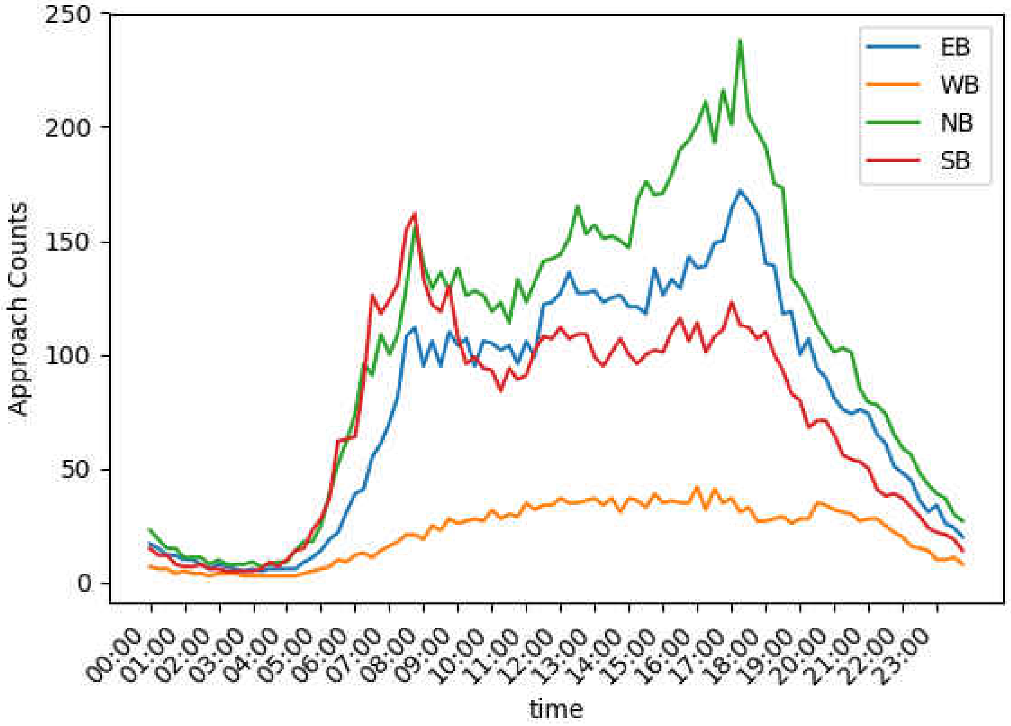


Figure 17 15-minutes counts of all approaches

Figure 17 shows the 15-minutes counts for all four approaches. It shows that the North-South approach is the major approach. The large volume of eastbound is caused by the right-turning vehicles using the dedicated right-turning lane.

In the field, the intersection is controlled by a coordinated actuated signal controller. In this study, a simulated controller using the timings provided by Seminole County is employed to replicate the field controller and serves as the benchmark (BC). The benchmark signal timing includes three Time of Day (TOD) plans for coordination and the signal runs fully actuated during the nighttime. Figure 20 shows the splits of TOD plans and max/min green time when the signal is fully actuated. The yellow time is five seconds and the all-red clearance time is two seconds. Another ATSC controller developed using a single objective RL algorithm (ATSC-SORL) (Gong, Abdel-Aty, Cai, et al., 2019), which aims at only optimizing traffic efficiency, is also used for comparison.

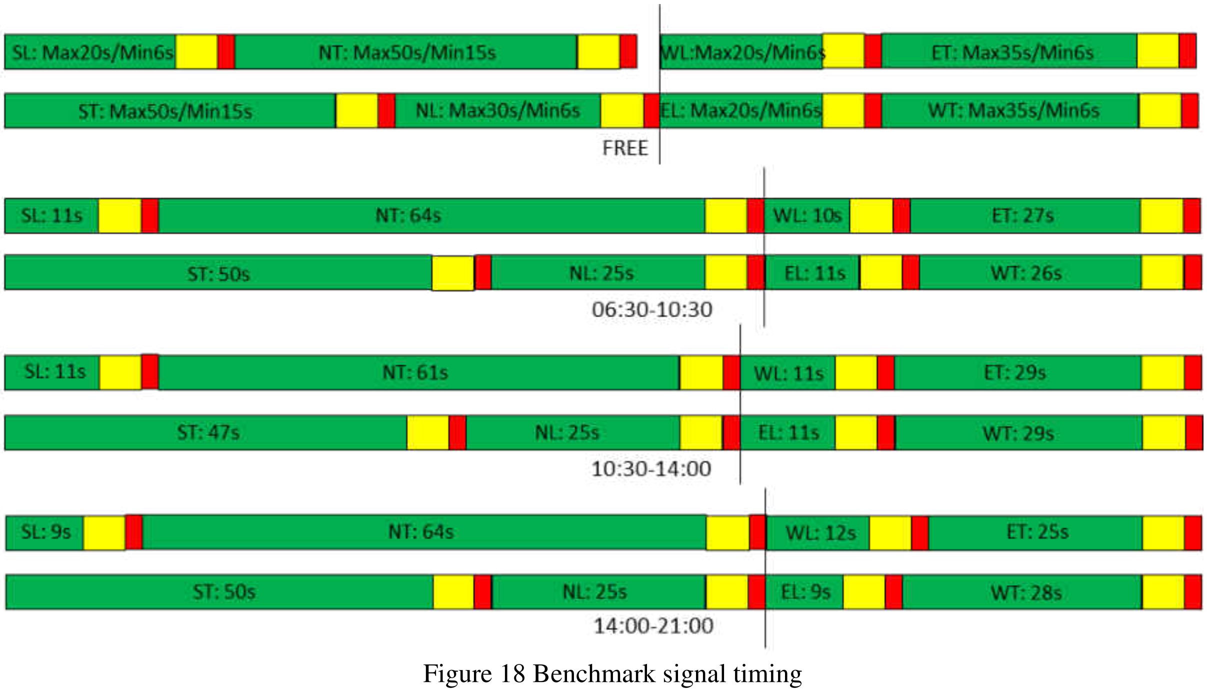


Figure 18 Benchmark signal timing

6.4.2 Real-time crash risk model

In this study, a real-time crash risk model is developed to forecast the crash odd in the next 5-10 minutes. The forecasted crash odds are used as the "risk score" to generate "safety reward".

As mentioned earlier, the real-time crash risk model is a binary classification problem, thus a binary logistic model is naturally preferred. If a crash occurred under certain conditions, the condition is classified as "crash" and vice versa. Suppose the "crash" case has the outcomes $y_i = 1$ and $y_i = 0$ with the respective probabilities of p_i and $1 - p_i,\ i = 1, 2, \ldots M.$ M represents the total number of samples. The binary logistic regression can be expressed as:

$$y_i \sim Bernoulli(p_i) \tag{22}$$

$$logit(p_i) = \ \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \cdots + \beta_K X_{Ki} \tag{23}$$

where β_0 is the intercept, $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_K)$ is the coefficients vector, and $\boldsymbol{X_i} = (x_1, x_2, \ldots, x_K)$ is the independent variable vector for the *i*th observation.

Similar to the previous study by Yuan et al. (2019b), the direct output of the binary logistic model is the predicted log crash odd of vehicles entering the intersection from a specific approach. As the number of crashes that have occurred at the test intersection is not sufficient to develop the model, crashes that have occurred in Seminole County, the same jurisdiction as the test intersection, were used. In total, data of 349 crashes from January 2017 to April 2018 were collected from Signal Four Analytics. These crashes occurring within the intersection area and the at-fault drivers were not under the influence of alcohol and drugs. Traffic data and signal timing logs of the intersections where crashes have occurred were extracted from ATSPM for a period of 10 minutes (divided into two 5-minutes time slices: slice 1 is 0-5 minutes and slice 2 is

5-10 minutes) prior to the crash occurrence. The data of different approaches were labeled using the same nomenclature as a previous study (Yuan and Abdel-Aty, 2018). The predicted approach is named as “A” approach, which is the traveling approach of the at-fault driver. “B”, “C” and “D” approaches are labeled following a clockwise sequence (please refer to Yuan and Abdel-aty, 2018 for more details). Since the crashes are rare events, the “non-crash” events are randomly sampled to generate a balanced dataset. In this study, 3215 “non-crash” events and 349 “crash” events were collected to calibrate the final model.

Table 8 shows the modeling results. The model estimation results show that the future crash potential at signalized intersections is affected by various factors that are collected from the four approaches, including the green ratio from A approach, through volume from C approach, arrive on green from D approach, etc. These results indicate that the crash odd is represented by the complicated interactions between signal timing and vehicle arrivals.

As the value outputted by the model is the predicted “risk score” for one approach, the final “risk score” of the whole intersection is defined as the average “risk score” of four approaches.

The risk score is calculated every minute using a rolling horizontal approach. For example, at 19:00, the model uses data from 18:50 to 19:00 to forecast the crash likelihood from 19:05 to 19:10; at 19:01, data from 18:51-19:01 is used to forecast the crash likelihood from 19:06 to 19:11. Because the control step is typically several seconds, the risk score might not change between two control steps. In this case, the safety reward derived by the risk score act as a “delayed” reward.

Table 8 Results of the Real-Time Crash Risk

Variable	Coefficient	Standard Error	P-Values	Nomenclature
Intercept	-2.095	0.410	0.000*	Prefix(approach label): A/B/C/D Turning Movement LT: Left turning; TH: Through Variable Type: AOG: Number of vehicles arrived at the intersection on green GR: Ratio of the green time within 5-minute Suffix(time slice): S1/S2 Example: (A_ TH _AOG_S1): Number of through vehicles arrived at the intersection on green of approach A (predicted approach) at the time slice 1(0-5 minutes before the crash occurrence)
A_LT_GR_S1	1.790	1.048	0.087**	
A_TH_AOG_S1	0.003	0.002	0.076**	
A_TH_GR_S2	-2.302	0.702	0.001*	
B_TH_GR_S2	-2.085	0.705	0.003*	
C_LT_AOG_S2	-0.018	0.010	0.068**	
C_TH_AOG_S1	0.007	0.002	0.000*	
C_TH_GR_S1	-2.167	0.624	0.001*	
C_TH_GR_S2	2.938	0.720	0.000*	
C_TH_Vol_S1	0.014	0.007	0.039*	
D_LT_AOG_S1	0.008	0.003	0.007*	
D_TH_AOG_S1	-2.229	0.972	0.022*	
D_TH_GR_S1	1.786	0.920	0.052**	

6.4.3 Algorithm Setting

Table 9 provides the algorithm setting. In general, the algorithm imitates the safety-related setting of the benchmark signal as much as possible. Moreover, the importance of traffic safety and efficiency was set to be equal.

The first phase of the pre-training was conducted using the benchmark signal controller, while the reference controller in the second phase of the pre-training is the ATSC-SORL controller.

Table 9 Parameters Used of the Algorithm

Parameter	Value	Description	Note
N_a	8	Number of actions	Number of phases
g_{min}	10 seconds (6:30-21:00) 6 seconds(otherwise)	Minimum green time	
t_y	5 seconds	Yellow time	Same as benchmark
t_{ar}	2 seconds	All-red clearance time	
p_0	NT/ST	Default phase	Major approach through
$t_{maxwaiting}$	180 seconds	Maximum allowed waiting time	
w_e	0.5	Weight of the efficiency objective	
w_s	0.5	Weight of the safety objective	
γ	0.99	Discount factor	
ϵ_e	0.9999	Ending greedy	To avoid oscillation
M	20000	Replay memory size	Roughly tuned
B	64	Minibatch size	
lr	0.00005	Learning rate	
t_{train}	1 second	length of training step	Same as control step

6.4.4 Results

To evaluate the performance of the proposed multi-objective ATSC algorithm, the well-trained control agent (ATSC-MORL), the benchmark controller (BC) and the single objective controller (ATSC-SORL) were implemented for 30 simulation days. Three kinds of performance measures were observed: average daily delay per vehicle, average daily number of stops per vehicle and the average daily intersection crash risk score. Table 10 shows the average daily performance of the 30 simulated days.

Table 10 Average Daily Performance of the Controllers

Controller	Efficiency		Safety (Risk Score)
	Average Delay (sec)	Number of Stops	
BC	26.395	0.703	0.045
ATSC-SORL	13.434	0.691	0.072
ATSC-MORL	19.550	0.615	0.041

According to the Table 10, compared to the BC controller, ATSC-MORL controller reduced average daily delay of by 25.93% (26.395 seconds versus 19.550 seconds), average daily number of stops by 12.52% (0.703 versus 0.615) and the average daily crash risk score by 8.89% (0.045 versus 0.041). Compared with the ATSC-SORL controller, the ATSC-MORL did improve traffic safety and reduced the number of stops while increasing travel time. Interestingly, while the ATSC-SORL reduces the delay dramatically (49.1%) comparing with the benchmark, it does increase the crash likelihood.

The performance of the three controllers at different times of day was also investigated. Figure 19 shows the change of performance measures in 15-min aggregation intervals. Although ATSC-MORL performs well in most situations, there exist certain conditions that ATSC-MORL performs worse than the benchmark. First, ATSC-MORL tends to increase the average delay per vehicle dramatically when the traffic demand is extremely low (23:00-06:00, please refer to Fig 3 for the demand). However, ATSC-MORL is able to reduce the delay when the traffic demand is medium to high. This is completely opposite to the benchmark controller. It is not supersizing as the goal of the RL-based ATSC is optimizing the total delay throughout the day; therefore, increasing the delay of a small number of vehicles while reducing the delay of a large number of increases the average number of stops per vehicle when the traffic demand is low.

Second, the ATSC-MORL failed to reduce the crash likelihood when the volume is close to zero (01:30-04:30). While admittedly, its objective is optimizing the risk score throughout the day, the causation needs to be further investigated.

In conclusion, the proposed ATSC-MORL based ATSC is able to improve both traffic efficiency and safety compared with the existing field controller. As traffic safety and efficiency are likely to be competing objectives, if the ATSC does not consider traffic safety, it might lead to potential safety issues.

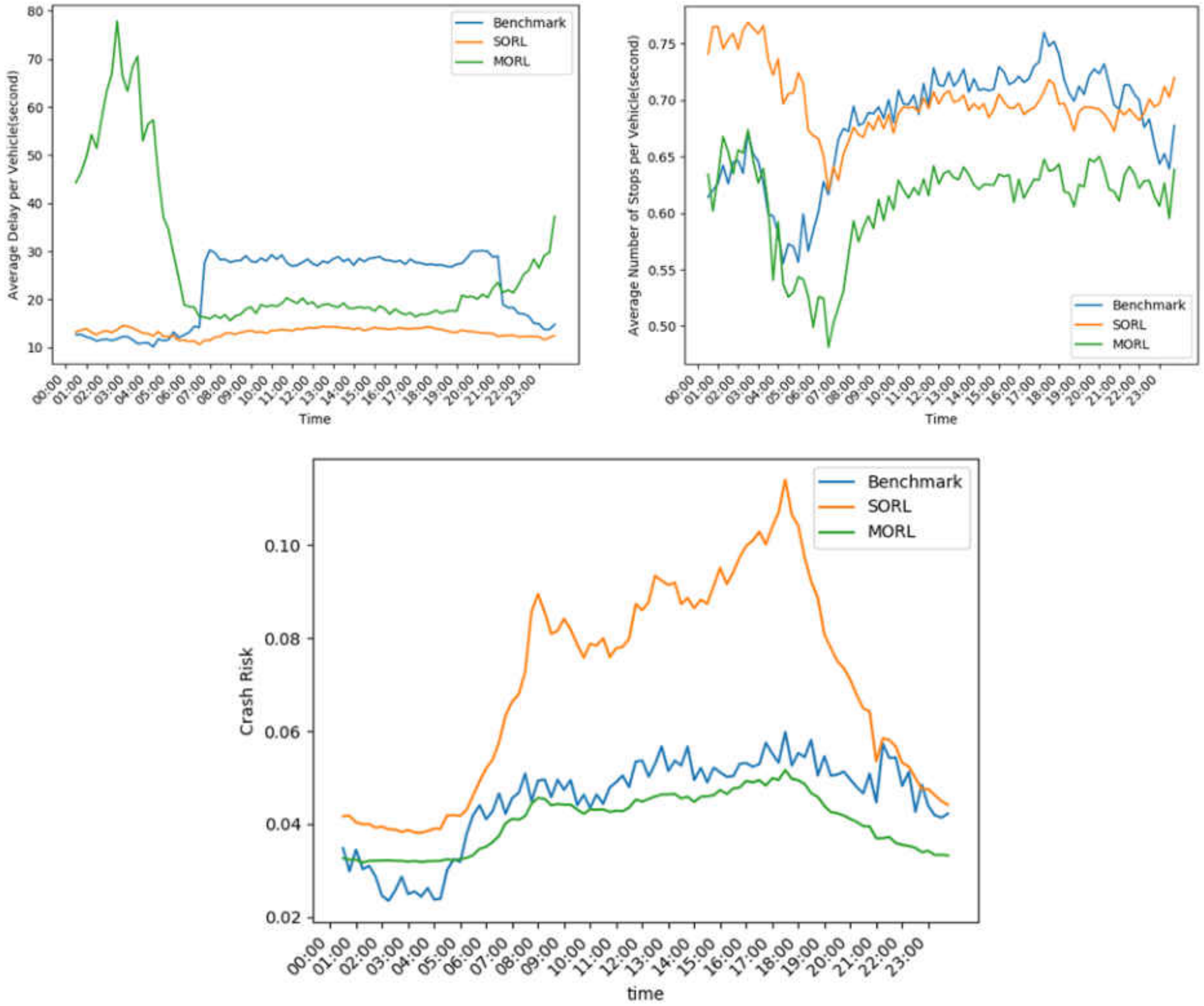


Figure 19 15-minutes aggregated performance measures of the controllers

6.5 Discussion

6.5.1 Control Policy Analysis: Opening the "Black Box"

Machine learning algorithms are criticized for their lack of interpretability, which is often referred to as the "Black Box" metaphor. While the vast majority of studies on RL-based ATSC showed their superior performance than traditional signal controllers, little attention has been given to illustrate how they achieve it. We would like to open the "Black Box" by analyzing the "optimal policy" of RL-based controllers on the test intersection. Especially there exist certain conditions that RL-based ATSC performs worse than the benchmark in terms of traffic safety. The analysis might not be comprehensive, but rather provides some insights for researchers and practitioners.

Several terms were defined to help control policy analysis:

Signal group: The set of turning movements that are controlled by the same traffic signal indications. For example, in this study, the northbound through movement and the northbound right-turning movement are controlled by the same set of signal indications. These two turning movements belong to the same signal group NT. Each phase could have a set of non-conflicting signal groups.

Signal group green interval length: The length of the time interval that the indication of a signal group is green (short for length in Table 11)

Green ratio: Ratio of the total signal group green interval length within a specific time interval (e.g. 15 minutes).

Table 11 Statistics of Green Interval Length, Activated Times and Green Ratio of Signal Groups

Controller	NL				NT			
	Length	Activated Times	Green Ratio	Flow (pdpl)	Length	Activated Times	Green Ratio	Flow (pdpl)
BC	9.55	702	7.8%	3379	45.30	1153	56.8%	6431
ATSC-SORL	8.49	1443	12.4%		17.48	2024	40.9%	
ATSC-MORL	6.39	1133	8.2%		37.02	1662	70.5%	
Controller	SL				ST			
	Length	Activated Times	Green Ratio	Flow (pdpl)	Length	Activated Times	Green Ratio	Flow (pdpl)
BC	7.15	248	2.2%	340	29.68	1272	41.6%	3251
ATSC-SORL	11.63	427	6.8%		7.94	2442	20.9%	
ATSC-MORL	6.41	136	1.2%		25.22	1881	52.1%	
Controller	EL				ET			
	Length	Activated Times	Green Ratio	Flow (pdpl)	Length	Activated Times	Green Ratio	Flow (pdpl)
BC	10.41	655	9.0%	632	9.91	912	10.2%	1614
ATSC-SORL	8.21	1137	9.0%		7.90	1178	8.5%	
ATSC-MORL	7.12	730	6.6%		6.27	851	6.5%	
Controller	WL				WT			
	Length	Activated Times	Green Ratio	Flow (pdpl)	Length	Activated Times	Green Ratio	Flow (pdpl)
BC	8.49	521	5.5%	532	8.66	763	7.6%	787
ATSC-SORL	10.38	1567	17.6%		10.26	1540	17.7%	
ATSC-MORL	7.20	725	7.6%		6.96	752	6.5%	

Table 11 presents the average length, the average green ratio, and the activated times of each turning movement. The daily traffic flow per lane is also provided as a reference. Figure 20 illustrates the average 15-minutes-aggregated green ratio for each signal group. In terms of efficiency, RL-based ATSCs (ATSC-SORL and ATSC-MORL) which have better performance exhibits shorter green intervals. In other words, they change the phase more frequently. For an isolated intersection, given that the queue is cleared, shorter green intervals reduce the waiting times of vehicles on approaches whose signal indications are red. This might lead to the delay reduction of the RL-based ATSCs.

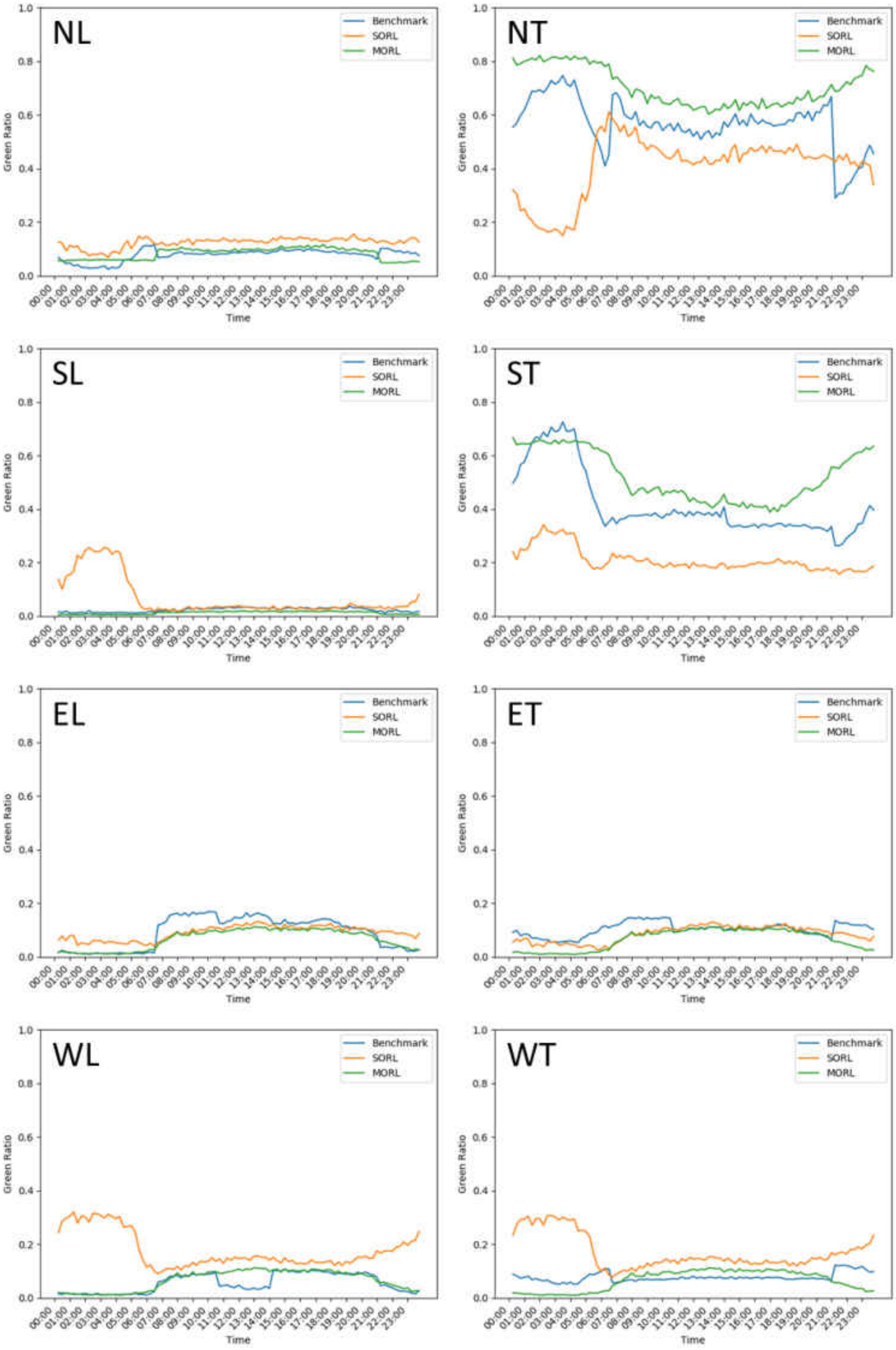


Figure 20 15-minutes aggregated green ratios of the controllers

As for traffic safety, ATSC-MORL and BC whose risk score are less than that of ATSC-SORL favor the major approach through movement (NT-ST). More specifically, for ATSC-MORL, the green ratio of the NT signal group (with the highest flow) is significantly larger while the green ratio of the SL signal group (with the lowest flow) is significantly smaller. One possible explanation is that such a policy reduces the probability of the occurrence of conflicts with the least sacrificing of traffic efficiency. Consider an extreme case. If the southbound left-turning is completely prohibited, the conflict points with northbound through movement and east-west approaches are eliminated. Since the flow of southbound left-turning is lowest among eight signal groups, prohibiting it would not lead to a huge increase of delay. However, a legal turning movement could not be prohibited if there exists demand, therefore, a compromise is achieved by reducing the activation of the SL signal group. The assumption could also reveal the reason why the ATSC-MORL performs worse when the volume is close to zero. In such conditions, the green ratio of ATSC-MORL is actually less than that of the benchmark.

To support the assumption, a hypothetical coordinated actuated signal controller (AD) is created by adjusting the benchmark controller. From 6:30 to 21:00, the length of SL phase was reduced to the minimum green time (six seconds) while the length of NT phase was increased accordingly. Table 12 shows the average daily performance of the AD controller during 30 test simulated days. It is not surprising that the average delay is higher than the BC as the southbound left-turning movement was intentionally delayed without the remedy of ATSC. This might also imply that traffic safety and efficiency are competing objectives. The risk score is significantly less than the BC controller and even a little bit less than the ATSC-MORL controller. The green ratio of SL signal group of AD controller is actually 1.1%, which is less than that of ATSC-

MORL controller. Therefore, the aforementioned assumption could be regarded as the abstracted knowledge learned by the RL-agent and it could be transferable to a different type of signal controller.

Table 12 Groups Average Daily Performance of the Adjusted Controller

Controller	Efficiency		Safety
	Average Delay (sec)	Number of Stops	(Risk Score)
BC	26.395	0.703	0.045
ATSC-MORL	19.550	0.615	0.041
AD	31.423	0.648	0.039

The result of the control policy analysis of simulated RL-based ATSCs could also serve as a reference for improving the existing signal control system if the infrastructure in the field is not ready to adopt RL-based ATSCs. However, as the proposed RL-based ATSCs are site-specific, if it is transferred to other locations, it is recommended to re-train it in a traffic simulation that replicates local traffic conditions.

6.5.2 Other Considerations of the Signal Settings

ATSCs improve traffic efficiency and/or safety by dynamically adjusting the signal timings. However, other signal settings beyond the timing parameters are also known to have impact on either efficiency or safety. Two tests were conducted to investigate how these factors influence traffic efficiency and/or safety.

Coordinated versus Non-coordinated

As there are no universally accepted rules to select the benchmarking controller, this study chooses to use a signal controller that replicates the field one. One might find that the field controller is designed for coordination yet this study focuses on an isolated intersection. Therefore, the performance of another hypothetical controller (HAC) that runs fully actuated throughout the day was compared with BC and ATSC-MORL (Table 13). The timing of the HAC was set as the same as BC when it runs fully actuated to avoid reevaluating safety-related timing parameters (such as minimum green time and passage time). According to Table 13, the performance of ATSC-MORL is better than HAC in terms of both traffic efficiency and safety, which further confirms the superiority of ATSC-MORL. Compared with coordinated BC, HAC reduces delay yet increases the number of stops. It is expected as the objective of coordination is to reduce the stops of the major approach through movement.

Permissive versus protected left-turn

It is well known that the protected left-turning is safer than permissive/permissive-protected left-turning yet is more detrimental to traffic efficiency. An interesting test scenario was developed to investigate the outcome if the permissive-protected left-turning of the east-west approach was changed to protected. Another multi-objective RL-agent (ATSC-MORLP) was trained under such conditions. According to Table 13, ATSC-MORLP created excessive delay as it prohibits permissive left-tuning. However, it did reduce the risk score comparing with ATSC-MORL (see also Figure 21), especially when the risk score is relatively high (from 15:00-19:00). This might imply that if the current crash risk is high, prohibiting permissive left-turning temporarily might be a potential solution.

Table 13 Average Daily Performance of the Other Tested Controllers

Controller	Efficiency		Safety (Risk Score)
	Average Delay (sec)	Number of Stops	
BC	26.395	0.703	0.045
ATSC-MORL	19.550	0.615	0.041
HAC	20.045	0.729	0.051
ATSC-MORLP	28.068	0.672	0.040*
** a paired T-test was conducted using the data of 30 simulated days to investigate whether the average risk scores are statistically significantly different between ATSC-MORL controlled scenarios and ATSC-MORLP controlled scenarios. The results showed that the difference is statistically significant at 0.0001 level.*			

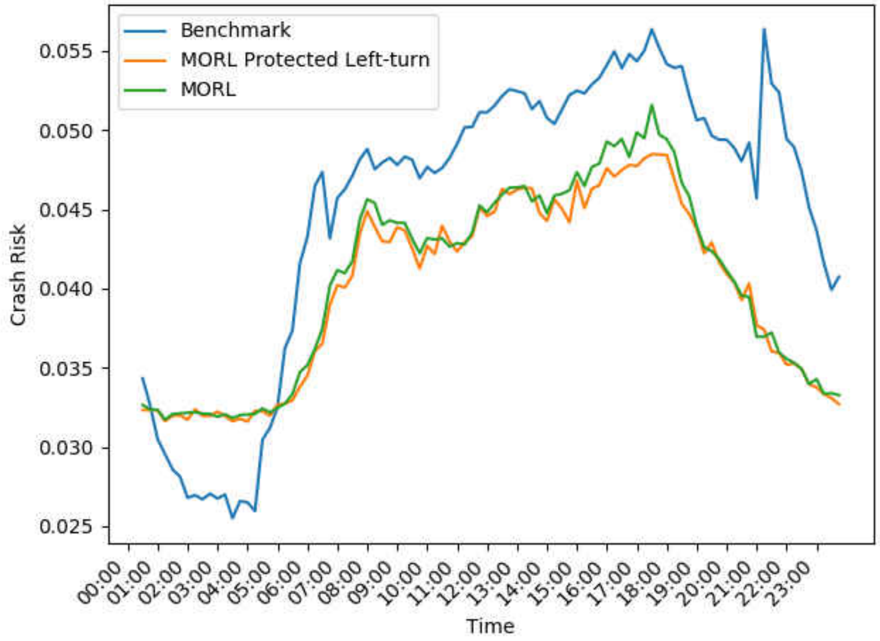


Figure 21 15-minutes aggregated risk score of the controllers

6.5.3 Hybrid Controller: A Better Solution

For any practical problems, there is always a trade-off between computational efficiency and algorithm's performance. While the weighted sum approach used in this study is computationally inexpensive, it is not guaranteed to be Pareto-optimal (Vamplew et al., 2008). Specifically, the ATSC-MORL controller performs worse than the BC controller does when the travel volume is extremely low. Therefore, a hybrid controller that changes its backend algorithm based on the traffic volume might have better performance.

A simple hybrid controller (HS) was proposed based on the local condition to test the feasibility of aforementioned concept. When the sum of 15-minute-flow-rates of all turning movements are below 150 vehicles per hour per lane, HS employs BC algorithm. Otherwise, HS employs ATSC-MORL algorithm. This eventually leads to a time-of-day-plan-like controller. From 0:00 to 5:00, BC is activated while ATSC-MORL is activated from 5:00-24:00. Table 16 shows the performance of HS controller compared with BC and ATSC-MORL controller. The HS controller slightly reduces the delay by 5.1% and reduces the average risk score by 2.5% compared with ATSC-MORL. The 15-minutes performance curve is not provided as the performance curve of HS is identical to the activated backend algorithm.

Table 14 Daily Performance of the Hybrid Controller Compared with the Benchmark and ATSC-MORL Controller

Controller	**Efficiency**		**Safety**
	Average Delay (sec)	**Number of Stops**	**(Risk Score)**
BC	26.395	0.703	0.045
ATSC-MORL	19.550	0.615	0.041
HS	18.538	0.615*	0.040*

* a *paired T-test was conducted using the data of 30 simulated days to investigate whether the number of stops and average safety scores are statistically significantly different between ATSC-MORL controlled scenarios and HS controlled scenarios. The results showed that the difference of number of the number stops is not statistically significant but the difference of average safety scores is statistically significant at 0.0001 level.*

Other types of hybrid controller such as the hybrid of ATSC-MORL and ATSC-MORLP could also be employed to improve traffic safety if necessary.

6.6 Summary and Conclusions

To improve the traffic safety of the signalized intersection, this Chapter proposes a safety-oriented adaptive signal control algorithm to simultaneously optimize traffic efficiency and safety. The control agent takes high-resolution real-time traffic data as its input and selects appropriate signal phases every second to reduce vehicles' delay and the crash risk of the intersection. A multi-objective reinforcement learning framework using double dueling deep neural network is utilized as the backend algorithm to solve the discrete optimization problem. The weighted sum approach, one of the single policy multi-objective reinforcement learning algorithms, is employed to deal with the trade-off between traffic safety and efficiency.

The proposed algorithm was trained and evaluated in a simulated isolated intersection in Seminole County, Florida, built based on field observed traffic data. A real-time crash prediction model is calibrated using local crash data to provide the crash risk in the near future. The performance of the well-trained algorithm was evaluated by the real-world signal timings provided by the local jurisdiction. The evaluation results showed that the algorithm improves both traffic efficiency and safety compared with the benchmark. In addition, compared with an adaptive traffic signal optimizing only traffic efficiency, it did improve traffic safety significantly but with a slight deterioration of traffic efficiency. This might imply the traffic safety and efficiency are two competing objectives. Practitioners should take the trade-off into consideration.

A brief analysis of control policies of different signal controller reveals how the RL-based ATSCs are able to improve traffic efficiency and safety. The abstracted control rules from the analysis could serve as a reference for improving existing signal control systems if the infrastructure in the field is not ready to adopt RL-based ATSCs. However, as the proposed RL-based ATSCs are site-specific, it is recommended to train the RL-based ATSC in a traffic simulation that replicates the local condition. A hybrid controller that changes its backend algorithm based on traffic volume is also proposed to improve the performance of MORL controlling algorithm if the well-trained MORL is not Pareto-optimal.

Admittedly, there are several limitations. As the weighted sum approach is not guaranteed to be Pareto-optimal, the study could be improved by calculating the Pareto-front using more computationally efficient algorithms. Meanwhile, other kinds of safety measures such as traffic conflicts could be tested as the safety objective using the proposed algorithm.

CHAPTER 7: CONCLUSION

7.1 Summary

This dissertation aims to provide ATSCs that are more practical-ready. More specifically, the main objectives of this dissertation are to (1) understand the types of traffic data on arterials that can be utilized as the data feeds of DRL-based ATSC; (2) develop algorithms to improve existing traffic data that could be used as the data feed of ATSC ; (3) develop a network-level ATSC algorithm based on decentralized Multi-Agent Reinforcement Learning (MARL) based on limited traffic information and real-world traffic dynamics ; (4) conduct an exploratory study on a multi-objective ATSC aiming at simultaneously optimizing traffic safety and efficiency in real-time .

In Chapter 3, various traffic data collection systems on arterials are reviewed. Existing traffic data collection systems in the field include traffic counts from inductive loop detectors installed for signal actuation, travel time data from BDS and segment-based speed data from the private sectors. However, the geographical coverage of infrastructure-based systems is limited. Thus, the widely available segment-based speed data from the private sectors could be served as valuable data feeds of ATSCs. However, the quality of data from the private sector is often a concern, thus rigorous validations or even necessary augmentations are needed. Moreover, state-of-the-practice ATSCs requires customized detection layout for their own control logic. Especially some vendors depend on the video detection systems.

In Chapter 4, experiments were conducted to evaluate the feasibility of BLE for estimating the traffic counts. Its detection rate and range were assessed. A two-step framework is then proposed for identifying the pedestrians and bicyclists from stationary objects and motorized travelers using one of the popular machine learning algorithms, one-class support vector machine. The proposed system is validated by the benchmark count data from video footage.

In Chapter 5, to evaluate the effectiveness of DRL-based ATSC on the real-world traffic dynamics, a network-level decentralized ATSC was developed using double dueling deep Q network in the multi-agent reinforcement learning framework. The algorithm was trained and evaluated in a simulated real-world traffic network. The simulated network replicates the geometric design of the real-world roadways and the simulation scenarios were calibrated by real-world traffic data from Automatic Traffic Signal Performance Measures (ATSPM). Besides, the proposed ATSC employs camera-like traffic detectors to capture the locations of vehicles close to the intersections rather than assuming the information of every single vehicle is known. The proposed algorithm was evaluated by the real-world coordinated actuated signals.

In Chapter 6, to improve traffic safety pro-actively, this study proposes a safety-oriented ATSC algorithm to simultaneously optimize traffic efficiency and safety. A multi-objective deep reinforcement learning framework is utilized as the backend algorithm. The proposed algorithm was trained and evaluated on a simulated isolated intersection built based on real-world traffic data. A real-time crash prediction model was calibrated to provide the safety measure. The performance of the algorithm was evaluated by the real-world signal timing provided by the local jurisdiction. The results showed that the algorithm improves both traffic efficiency and safety compared with the benchmark.

7.2 Implications

Chapter 4 concludes that comparing to ordinary Bluetooth and Wi-Fi, BLE is more suitable for estimating the counts of pedestrians and bicyclists because of its high detection rate and reasonable detection range. The developed machine-learning-based algorithm is able to estimate the counts of pedestrians and bicyclists of a mixed-traffic environment. The average absolute percentage error is 6.35%. To the best of the author's knowledge, it is the first time that BLE technology is used to estimate traffic counts.

Chapter 5 introduced a DRL-based ATSC based on real-world traffic dynamics. The proposed algorithm outperforms the coordinated actuated signal used in the field. It is able to reduce 10.27% of the travel time and almost half of (46.46%) of the total delay. This work proved that without knowing the status of every vehicle, the DRL-based ATSC could successfully adapt to the real-world complicated traffic dynamics in the network-level. This work makes a step forward for the real-world implementation of the DRL-based ATSC.

Chapter 6 successfully developed a DRL-based safety-oriented ATSC. The proposed algorithm is able to reduce the crash risk by around 9% than the benchmark actuated traffic signal, given that the DRL-based ATSC only optimizing the traffic efficiency might be detrimental to traffic safety. Moreover, abstracted control rules of the proposed ATSC could help the traditional signal controllers to improve traffic safety, which might be beneficial if the infrastructure is not ready to adopt ATSCs. A hybrid controller is also proposed to provide further traffic safety improvement if necessary. To the best of the author’s knowledge, the proposed algorithm is the first successful attempt in developing adaptive traffic signal system optimizing traffic safety.

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I. J., Harp, A., Irving, G., Isard, M., Jia, Y., Józefowicz, R., Kaiser, L., Kudlur, M., … Zheng, X. (2016). TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. *CoRR*, *abs/1603.0*. http://arxiv.org/abs/1603.04467

Abdel-Aty, M., Dilmore, J., & Dhindsa, A. (2006). Evaluation of variable speed limits for real-time freeway safety improvement. *Accident Analysis and Prevention*, *38*(2), 335–345. https://doi.org/10.1016/j.aap.2005.10.010

Abdel-Aty, M., Uddin, N., Pande, A., Abdalla, M. F., & Hsia, L. (2004). Predicting Freeway Crashes from Loop Detector Data by Matched Case-Control Logistic Regression. *Transportation Research Record*, *1897*(1), 88–95. https://doi.org/10.3141/1897-12

Abdoos, M, Mozayani, N., & Bazzan, A. L. C. (2011). Traffic light control in non-stationary environments based on multi agent Q-learning. *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 1580–1585. https://doi.org/10.1109/ITSC.2011.6083114

Abdoos, Monireh, Mozayani, N., & Bazzan, A. L. C. (2014). Hierarchical control of traffic signals using Q-learning with tile coding. *Applied Intelligence*, *40*(2), 201–213.

Abdulhai, B., Pringle, R., & Karakoulas, G. J. (2003). Reinforcement learning for true adaptive traffic signal control. *Journal of Transportation Engineering*, *129*(3), 278–285.

Abedi, N., Bhaskar, A., Chung, E., & Miska, M. (2015). Assessment of antenna characteristic effects on pedestrian and cyclists travel-time estimation based on Bluetooth and WiFi MAC addresses. *Transportation Research Part C: Emerging Technologies*, *60*, 124–141. https://doi.org/https://doi.org/10.1016/j.trc.2015.08.010

Araghi, B. N., Christensen, L. T., Krisnan, R., Olesen, J. H., & Lahrman, H. (2013). Improving The Accuracy Of Bluetooth Based Travel Time Estimation Using Low-Level Sensor Data. *Transportation Research Board 92th Annual Meeting*, *1750*(January), 1–11. https://doi.org/10.3141/2338-04

Araghi, B. N., Krishnan, R., & Lahrmann, H. (2016). Mode-Specific Travel Time Estimation Using Bluetooth Technology. *Journal of Intelligent Transportation Systems*, *20*(3), 219–228. https://doi.org/10.1080/15472450.2015.1052906

Arel, I., Liu, C., Urbanik, T., & Kohls, A. G. (2010). Reinforcement learning-based multi-agent system for network traffic signal control. *IET Intelligent Transport Systems*, *4*(2), 128–135. https://doi.org/10.1049/iet-its.2009.0070

Arroyo, V. A., Bennett, S. E., Butler, D. H., Dougherty, M., Stewart Fotheringham, A., Halikowski, J. S., Dot, A., Michael Hancock, P. W., Hanson, S., Heminger, S., Hendrickson, C. T., Knatz, G., Osterberg, D. A., Rosenbloom, S., Schwartz, H. G., Sinha, K. C., Steudle, K. T., Dot, M., Gary Thomas, L. C., … Zukunft, P. F. (2015). NCHRP Report 812 – Signal Timing Manual, Second Edition. In *Nchrp: Vol. AASHTO, FH*.

Aslani, M., Mesgari, M. S., & Wiering, M. (2017). Adaptive traffic signal control with actor-critic methods in a real-world traffic network with different traffic disruption events. *Transportation Research Part C: Emerging Technologies*, *85*, 732–752. https://doi.org/https://doi.org/10.1016/j.trc.2017.09.020

Balaji, P. G., German, X., & Srinivasan, D. (2010). Urban traffic signal control using reinforcement learning agents. *IET Intelligent Transport Systems*, *4*(3), 177–188. https://doi.org/10.1049/iet-its.2009.0096

Barcelo, J. (2018). *AIMSUN MICROSCOPIC TRAFFIC SIMULATOR: A TOOL FOR THE ANALYSIS AND ASSESSMENT OF ITS SYSTEMS*.

Bathaee, N., Mohseni, A., Park, S., Porter, J. D., & Kim, D. S. (2018). A cluster analysis approach for differentiating transportation modes using Bluetooth sensor data. *Journal of Intelligent Transportation Systems*, *22*(4), 353–364. https://doi.org/10.1080/15472450.2018.1457444

Bazzan, A. L. C., de Oliveira, D., & da Silva, B. C. (2010). Learning in groups of traffic signals. *Engineering Applications of Artificial Intelligence*, *23*(4), 560–568.

Bertsekas, D., & Tsitsiklis, J. (1996). Neuro-Dynamic Programming. In *Third World Planning Review - THIRD WORLD PLAN REV* (Vol. 27). https://doi.org/10.1007/978-0-387-74759-0_440

Bhaskar, A., & Chung, E. (2013). Fundamental understanding on the use of Bluetooth scanner as a complementary transport data. *Transportation Research Part C: Emerging Technologies*, *37*, 42–72. https://doi.org/10.1016/j.trc.2013.09.013

Board, T. R., & National Academies of Sciences and Medicine, E. (2010). *Adaptive Traffic Control Systems: Domestic and Foreign State of Practice*. The National Academies Press. https://doi.org/10.17226/14364

Bretherton, R. D. (1990). SCOOT URBAN TRAFFIC CONTROL SYSTEM—PHILOSOPHY AND EVALUATION11Crown Copyright. The views expressed in this Paper are not necessarily those of the Department of Transport. Extracts from the text may be reproduced, except for commercial purposes, provided t. In J.-P. PERRIN (Ed.), *Control, Computers, Communications in Transportation* (pp. 237–239). Pergamon. https://doi.org/https://doi.org/10.1016/B978-0-08-037025-5.50040-2

Bullock, D. M., Haseman, R., Wasson, J. S., & Spitler, R. (2010). Automated Measurement of Wait Times at Airport Security: Deployment at Indianapolis International Airport, Indiana. *Transportation Research Record*, *2177*(1), 60–68. https://doi.org/10.3141/2177-08

Camponogara, E., & Kraus, W. (2003). Distributed learning agents in urban traffic control. *Portuguese Conference on Artificial Intelligence*, 324–335.

Camurri, M., Mamei, M., & Zambonelli, F. (2006). *Urban Traffic Control with Co-Fields* (pp. 239–253). https://doi.org/10.1007/978-3-540-71103-2_14

Casas, N. (2017). *Deep Deterministic Policy Gradient for Urban Traffic Light Control*. http://arxiv.org/abs/1703.09035

Chu, T, Qu, S., & Wang, J. (2016). Large-scale traffic grid signal control with regional

Reinforcement Learning. *2016 American Control Conference (ACC)*, 815–820. https://doi.org/10.1109/ACC.2016.7525014

Chu, Tianshu, Qu, S., & Wang, J. (2016). Large-scale traffic grid signal control with regional Reinforcement Learning. In *Proceedings of the American Control Conference* (Vols. 2016-July, pp. 815–820). https://doi.org/10.1109/ACC.2016.7525014

Chung, W., Abdel-Aty, M., Park, H.-C., Cai, Q., Rahman, M., Gong, Y., & Ponnaluri, R. (2020). Development of Decision Support System for Integrated Active Traffic Management Systems Considering Travel Time Reliability. *Transportation Research Record*, *2674*(2), 167–180. https://doi.org/10.1177/0361198120905591

da Silva, B. C., Bazzan, A. L. C., Andriotti, G. K., Lopes, F., & de Oliveira, D. (2006). ITSUMO: An Intelligent Transportation System for Urban Mobility. In T. Böhme, V. M. Larios Rosillo, H. Unger, & H. Unger (Eds.), *Innovative Internet Community Systems* (pp. 224–235). Springer Berlin Heidelberg.

de Oliveira, D., Bazzan, A. L. C., da Silva, B. C., Basso, E. W., Nunes, L., Rossetti, R., de Oliveira, E., da Silva, R., & Lamb, L. (2006). Reinforcement Learning based Control of Traffic Lights in Non-stationary Environments: A Case Study in a Microscopic Simulator. *EUMAS*.

der Pol, E., & Oliehoek, F. A. (2016). Coordinated Deep Reinforcement Learners for Traffic Light Control. *NIPS'16 Workshop on Learning, Inference and Control of Multi-Agent Systems*.

Du, Y., Yue, J., Ji, Y., & Sun, L. (2017). Exploration of optimal Wi-Fi probes layout and estimation model of real-time pedestrian volume detection. *International Journal of Distributed Sensor Networks*, *13*(11), 1550147717741857. https://doi.org/10.1177/1550147717741857

El-Tantawy, S, & Abdulhai, B. (2012). Multi-Agent Reinforcement Learning for Integrated Network of Adaptive Traffic Signal Controllers (MARLIN-ATSC). In *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on* (Vol. 14, Issue September 2015, pp. 319–326). https://doi.org/10.1109/ITSC.2012.6338707

El-Tantawy, S, & Abdulhai, B. (2010). An agent-based learning towards decentralized and coordinated traffic signal control. *13th International IEEE Conference on Intelligent Transportation Systems*, 665–670. https://doi.org/10.1109/ITSC.2010.5625066

El-Tantawy, Samah, Abdulhai, B., & Abdelgawad, H. (2014). Design of reinforcement learning parameters for seamless application of adaptive traffic signal control. *Journal of Intelligent Transportation Systems*, *18*(3), 227–245.

Elefteriadou, L., Kondyli, A., & George, B. St. (2014). Comparison of Methods for Measuring Travel Time at Florida Freeways and Arterials. In *Transportation Research Center, University of Florida* (Issue July). https://doi.org/10.1007/s13398-014-0173-7.2

Federal Highway Administration. (2017). *2017 National Household Travel Survey*.

Feng, Y., Zheng, J., & Liu, H. X. (2018). Real-Time Detector-Free Adaptive Signal Control with Low Penetration of Connected Vehicles. *Transportation Research Record*, *2672*(18), 35–

44. https://doi.org/10.1177/0361198118790860

Fink, J., Kwigizile, V., & Oh, J.-S. (2016). Quantifying the impact of adaptive traffic control systems on crash frequency and severity: Evidence from Oakland County, Michigan. *Journal of Safety Research*, *57*, 1–7. https://doi.org/10.1016/J.JSR.2016.01.001

Friesen, M. R., & McLeod, R. D. (2015). Bluetooth in Intelligent Transportation Systems: A Survey. *International Journal of Intelligent Transportation Systems Research*, *13*(3), 143–153. https://doi.org/10.1007/s13177-014-0092-1

Gao, J., Shen, Y., Liu, J., Ito, M., & Shiratori, N. (2017). Adaptive Traffic Signal Control: Deep Reinforcement Learning Algorithm with Experience Replay and Target Network. In *arXiv* (pp. 1–10). https://arxiv.org/pdf/1705.02755.pdf%0Ahttp://arxiv.org/abs/1705.02755

Genders, W., & Razavi, S. (2016). Using a deep reinforcement learning agent for traffic signal control. *ArXiv Preprint ArXiv:1611.01142*.

Gomez, C., Oller, J., & Paradells, J. (2012). Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology. *Sensors*, *12*(9), 11734–11753.

Gong, Y., Abdel-Aty, M., Cai, Q., & Rahman, M. S. (2019). Decentralized network level adaptive signal control by multi-agent deep reinforcement learning. *Transportation Research Interdisciplinary Perspectives*, *1*, 100020. https://doi.org/https://doi.org/10.1016/j.trip.2019.100020

Gong, Y., Abdel-Aty, M., & Park, J. (2019). Evaluation and augmentation of traffic data including Bluetooth detection system on arterials. *Journal of Intelligent Transportation*

Systems, 1–13. https://doi.org/10.1080/15472450.2019.1632707

Goodall, N. J., Smith, B. L., & Park, B. (Brian). (2013). Traffic Signal Control with Connected Vehicles. *Transportation Research Record*, *2381*(1), 65–72. https://doi.org/10.3141/2381-08

Gu, S., Lillicrap, T., Sutskever, I., & Levine, S. (2016). Continuous deep q-learning with model-based acceleration. *International Conference on Machine Learning*, 2829–2838.

Haghani, A., Hamedi, M., Sadabadi, K., Young, S., & Tarnoff, P. (2010). Data Collection of Freeway Travel Time Ground Truth with Bluetooth Sensors. *Transportation Research Record: Journal of the Transportation Research Board*, *2160*(2160), 60–68. https://doi.org/10.3141/2160-07

Heinen, M. R., Bazzan, A. L. C., & Engel, P. M. (2011). Dealing with continuous-state reinforcement learning for intelligent control of traffic signals. *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 890–895. https://doi.org/10.1109/ITSC.2011.6083107

Houli, D., Zhiheng, L., & Yi, Z. (2010). Multiobjective Reinforcement Learning for Traffic Signal Control Using Vehicular Ad Hoc Network. *EURASIP J. Adv. Signal Process*, *2010*, 7:1--7:7. https://doi.org/10.1155/2010/724035

Hummer, J. E., Rouphail, N. M., Toole, J. L., Patten, R. S., Schneider, R. J., Green, J. S., Hughes, R. G., & Fain, S. J. (2006). *Evaluation of Safety, Design, and Operation of Shared-Use Paths—Final Report*.

Jiaqi, M., D., F. M., Fang, Z., Jia, H., K., H. D., & O., C. M. (2016). Estimation of Crash Modification Factors for an Adaptive Traffic-Signal Control System. *Journal of Transportation Engineering*, *142*(12), 4016061. https://doi.org/10.1061/(ASCE)TE.1943-5436.0000890

Jin, J., & Ma, X. (2015). Adaptive Group-based Signal Control by Reinforcement Learning. *Transportation Research Procedia*, *10*, 207–216. https://doi.org/https://doi.org/10.1016/j.trpro.2015.09.070

Jin, J., & Ma, X. (2018). Hierarchical multi-agent control of traffic lights based on collective learning. In *Engineering Applications of Artificial Intelligence* (Vol. 68, pp. 236–248). https://doi.org/10.1016/j.engappai.2017.10.013

José, J., Díaz, V., Belén, A., González, R., & Wilby, M. R. (2015). *Bluetooth Traffic Monitoring Systems for Travel Time Estimation on Freeways*. *17*(1), 1–10.

Joyoung, L., (Brian), P. B., & Ilsoo, Y. (2013). Cumulative Travel-Time Responsive Real-Time Intersection Control Algorithm in the Connected Vehicle Environment. *Journal of Transportation Engineering*, *139*(10), 1020–1029. https://doi.org/10.1061/(ASCE)TE.1943-5436.0000587

Karlsson, J. (1997). *Learning to solve multiple goals*. University of Rochester.

Kasten, O., & Langheinrich, M. (2001). First Experiences with Bluetooth in the Smart-Its Distributed Sensor Network. *Workshop on Ubiquitous Computing and Communications, PACT 2001, Barcelona, Spain, September 8-12, 2001*, *00*.

Khamis, M. A., & Gomaa, W. (2014). Adaptive multi-objective reinforcement learning with hybrid exploration for traffic signal control based on cooperative multi-agent framework. *Engineering Applications of Artificial Intelligence*, *29*, 134–151.

Khattak, Z. H., Magalotti, M. J., & Fontaine, M. D. (2018). Estimating safety effects of adaptive signal control technology using the Empirical Bayes method. *Journal of Safety Research*, *64*, 121–128. https://doi.org/10.1016/J.JSR.2017.12.016

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *ArXiv Preprint ArXiv:1412.6980*.

Krajzewicz, D., Erdmann, J., Behrisch, M., & Bieker, L. (2012). Recent Development and Applications of {SUMO - Simulation of Urban MObility}. *International Journal On Advances in Systems and Measurements*, *5*(3&4), 128–138.

Kurkcu, A., & Ozbay, K. (2017). Estimating Pedestrian Densities, Wait Times, and Flows with Wi-Fi and Bluetooth Sensors. *Transportation Research Record*, *2644*(1), 72–82. https://doi.org/10.3141/2644-09

Kuyer, L., Whiteson, S., Bakker, B., & Vlassis, N. (2008). Multiagent Reinforcement Learning for Urban Traffic Control Using Coordination Graphs. In W. Daelemans, B. Goethals, & K. Morik (Eds.), *Machine Learning and Knowledge Discovery in Databases* (pp. 656–671). Springer Berlin Heidelberg.

LA, P., & Bhatnagar, S. (2011). Reinforcement Learning With Function Approximation for Traffic Signal Control. *IEEE Transactions on Intelligent Transportation Systems*, *12*(2), 412–421. https://doi.org/10.1109/TITS.2010.2091408

Lee, J., Abdel-Aty, M., Xu, P., & Gong, Y. (2019). *Is the Safety-In-Numbers Effect Still Observed in Areas with Low Pedestrian Activities?: A Case Study from Central Florida*.

Lemos, L. L., Bazzan, A. L. C., & Pasin, M. (2018). Co-Adaptive Reinforcement Learning in Microscopic Traffic Systems. *2018 IEEE Congress on Evolutionary Computation (CEC)*, 1–8. https://doi.org/10.1109/CEC.2018.8477713

Li, L., Lv, Y., & Wang, F.-Y. (2016a). Traffic signal timing via deep reinforcement learning. *IEEE/CAA Journal of Automatica Sinica*, *3*(3), 247–254.

Li, L., Lv, Y., & Wang, F.-Y. (2016b). Traffic signal timing via deep reinforcement learning. In *IEEE/CAA Journal of Automatica Sinica* (Vol. 3, Issue 3, pp. 247–254). https://doi.org/10.1109/JAS.2016.7508798

Li, P., Abdel-Aty, M., & Yuan, J. (2020). Real-time crash risk prediction on arterials based on LSTM-CNN. *Accident Analysis & Prevention*, *135*, 105371. https://doi.org/https://doi.org/10.1016/j.aap.2019.105371

Liang, X., Du, X., Wang, G., & Han, Z. (2018). Deep Reinforcement Learning for Traffic Light Control in Vehicular Networks. In *Ieee Transactions on Vehicular Technology* (Vol. 1, Issue Xx, pp. 1–11). https://arxiv.org/pdf/1803.11115.pdf

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. (2015). Continuous control with deep reinforcement learning. *CoRR*, *abs/1509.0*. http://arxiv.org/abs/1509.02971

LIN, L.-J. (1993). Reinforcement Learning for Robots Using Neural Networks. *Ph. D. Thesis, Carnegie Mellon University*. https://ci.nii.ac.jp/naid/20000106896/en/

Lin, Y., Dai, X., Li, L., & Wang, F.-Y. (2018). *An Efficient Deep Reinforcement Learning Model for Urban Traffic Control*.

Liu, C., Xu, X., & Hu, D. (2015). Multiobjective Reinforcement Learning: A Comprehensive Overview. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *45*(3), 385–398. https://doi.org/10.1109/TSMC.2014.2358639

Liu, S., McGree, J., White, G., & Dale, W. (2014). Transport mode identification by clustering travel time data. *ANZIAM Journal*, *56*, 95–116.

Maas, A. L., Hannun, A. Y., & Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. *Proc. Icml*, *30*(1), 3.

Malinovskiy, Y., Saunier, N., & Wang, Y. (2012). Analysis of Pedestrian Travel with Static Bluetooth Sensors. *Transportation Research Record*, *2299*(1), 137–149. https://doi.org/10.3141/2299-15

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., Silver, D., & Kavukcuoglu, K. (2016). Asynchronous Methods for Deep Reinforcement Learning. *ICML*.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*, 529.

http://dx.doi.org/10.1038/nature14236

Mousavi, S. S., Schukat, M., & Howley, E. (2017). *Traffic Light Control Using Deep Policy-Gradient and Value-Function Based Reinforcement Learning*. https://doi.org/10.1049/iet-its.2017.0153

Mousavi, S. S., Schukat, M., & Howley, E. (2018). Deep Reinforcement Learning: An Overview. In Y. Bi, S. Kapoor, & R. Bhatia (Eds.), *Proceedings of SAI Intelligent Systems Conference (IntelliSys) 2016* (pp. 426–440). Springer International Publishing.

Muresan, M., Fu, L., & Pan, G. (2018). Adaptive Traffic Signal Control with Deep Reinforcement Learning – An Exploratory Investigation. *97th Annual Meeting of the Transportation Research Board*.

Newzoo. (2019). *Top 20 Countries/Markets by Smartphone Users*.

Ohlms, P. B., Dougald, L. E., & MacKnight, H. E. (2019). Bicycle and Pedestrian Count Programs: Scan of Current U.S. Practice. *Transportation Research Record*, *2673*(3), 74–85. https://doi.org/10.1177/0361198119834924

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2012). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830. https://doi.org/10.1007/s13398-014-0173-7.2

Rahman, M. H., Abdel-Aty, M., Lee, J., & Rahman, M. S. (2019). Enhancing traffic safety at school zones by operation and engineering countermeasures: A microscopic simulation approach. *Simulation Modelling Practice and Theory*, *94*, 334–348. https://doi.org/https://doi.org/10.1016/j.simpat.2019.04.001

Rahman, M. S., Abdel-Aty, M., Lee, J., & Rahman, M. H. (2019a). Safety benefits of arterials' crash risk under connected and automated vehicles. *Transportation Research Part C: Emerging Technologies*, *100*, 354–371. https://doi.org/https://doi.org/10.1016/j.trc.2019.01.029

Rahman, M. S., Abdel-Aty, M., Lee, J., & Rahman, M. H. (2019b). *Understanding the Safety Benefits of Connected and Automated Vehicles on Arterials' Intersections and Segments*.

Richter, S., Aberdeen, D., & Yu, J. (2007). Natural Actor-Critic for Road Traffic Optimisation. In B Schölkopf, J. C. Platt, & T. Hoffman (Eds.), *Advances in Neural Information Processing Systems 19* (pp. 1169–1176). MIT Press. http://papers.nips.cc/paper/3087-natural-actor-critic-for-road-traffic-optimisation.pdf

Sabra, Z. A., Gettman, D., Venkata Nallamothu, & Pecker, C. (2013). *Enhancing Safety and Capacity in an Adaptive Signal Control System (Phase 2)*. Sabra, Wang & Associates, Inc. https://doi.org/10.13140/RG.2.2.16217.83044

Salkham, A. ad, Cunningham, R., Garg, A., & Cahill, V. (2008). A collaborative reinforcement learning approach to urban traffic control optimization. *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 02*, 560–566.

Schauer, L., Werner, M., & Marcus, P. (2014). Estimating crowd densities and pedestrian flows using wi-fi and bluetooth. *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, 171–177.

Schölkopf, Bernhard, Platt, J. C., Shawe-Taylor, J. C., Smola, A. J., & Williamson, R. C. (2001). Estimating the Support of a High-Dimensional Distribution. *Neural Comput.*, *13*(7), 1443–1471. https://doi.org/10.1162/089976601750264965

Sharma, S., Lüßmann, J., & So, J. (2019). Controller Independent Software-in-the-Loop Approach to Evaluate Rule-Based Traffic Signal Retiming Strategy by Utilizing Floating Car Data. *IEEE Transactions on Intelligent Transportation Systems*, *20*(9), 3585–3594. https://doi.org/10.1109/TITS.2018.2877585

Shoufeng, L., Ximin, L., & Shiqiang, D. (2008). Q-Learning for adaptive traffic signal control based on delay minimization strategy. *Networking, Sensing and Control, 2008. ICNSC 2008. IEEE International Conference On*, 687–691.

Singer, J., Robinson, A. E., Krueger, J., Atkinson, J. E., & Myers, M. C. (2013). *Travel Time on Arterials and Rural Highways: State-of-the-Practice Synthesis on Arterial Data Collection Technology, FHWA-HOP-13-028. April*, 60. http://www.ops.fhwa.dot.gov/publications/fhwahop13028/index.htm

Singh, S. P., & Sutton, R. S. (1996). Reinforcement learning with replacing eligibility traces. *Machine Learning*, *22*(1), 123–158. https://doi.org/10.1007/BF00114726

Smith, M., Duncan, G., & Druitt, S. (1995). PARAMICS: microscopic traffic simulation for congestion management. *IEE Colloquium on Dynamic Control of Strategic Inter-Urban Road Networks*, 8/1-8/3. https://doi.org/10.1049/ic:19950249

Stevanovic, A., Stevanovic, J., & Kergaye, C. (2013). Optimization of traffic signal timings based on surrogate measures of safety. *Transportation Research Part C: Emerging Technologies*, *32*, 159–178. https://doi.org/10.1016/J.TRC.2013.02.009

Stevanovic, A., Stevanovic, J., So, J., & Ostojic, M. (2015). Multi-criteria optimization of traffic signals: Mobility, safety, and environment. *Transportation Research Part C: Emerging Technologies*, *55*, 46–68. https://doi.org/https://doi.org/10.1016/j.trc.2015.03.013

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine Learning*, *3*(1), 9–44. https://doi.org/10.1007/BF00115009

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Tageldin, A., Sayed, T., Zaki, M. H., & Azab, M. (2014). A safety evaluation of an Adaptive Traffic Signal Control system using Computer Vision. *Advances in Transportation Studies*, *Special Vol2*, 83–98. https://login.ezproxy.net.ucf.edu/login?auth=shibb&url=https://search.ebscohost.com/login.aspx?direct=true&db=aph&AN=97143590&site=eds-live&scope=site

Tan, T. (n.d.). *Centralized Traffic Grid Signal Control via Deep Reinforcement Learning*.

Thorpe, T. L., & Anderson, C. W. (1996). *Traffic Light Control Using SARSA with Three State Representations*.

Tsitsiklis, J. N., & Roy, B. Van. (1997). An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, *42*(5), 674–690. https://doi.org/10.1109/9.580874

Vamplew, P., Yearwood, J., Dazeley, R., & Berry, A. (2008). *On the Limitations of Scalarisation for Multi-objective Reinforcement Learning of Pareto Fronts BT - AI 2008: Advances in Artificial Intelligence* (W. Wobcke & M. Zhang (eds.); pp. 372–378). Springer Berlin Heidelberg.

Van Der Pol, E., & Oliehoek, F. A. (2016). Coordinated Deep Reinforcement Learners for Traffic Light Control. In *NIPS'16 Workshop on Learning, Inference and Control of Multi-Agent Systems* (Issue Nips). http://www.fransoliehoek.net/docs/VanDerPol16LICMAS.pdf%0Ahttp://elisevanderpol.nl/papers/vanderpol_oliehoek_nipsmalic2016.pdf

van Hasselt, H., Guez, A., & Silver, D. (2015). Deep Reinforcement Learning with Double Q-learning. *CoRR*, *abs/1509.0*. http://arxiv.org/abs/1509.06461

Vidhate, D. A., & Kulkarni, P. (2017). Cooperative multi-agent reinforcement learning models (CMRLM) for intelligent traffic control. *2017 1st International Conference on Intelligent Systems and Information Management (ICISIM)*, 325–331. https://doi.org/10.1109/ICISIM.2017.8122193

Wang, L., Abdel-Aty, M., & Lee, J. (2017). Implementation of Active Traffic Management Strategies for Safety on Congested Expressway Weaving Segments. *Transportation Research Record*, *2635*(1), 28–35. https://doi.org/10.3141/2635-04

Wang, Z., Schaul, T., Hessel, M., Van Hasselt, H., Lanctot, M., & De Freitas, N. (2015). Dueling network architectures for deep reinforcement learning. *ArXiv Preprint ArXiv:1511.06581*.

Watkins, C. J. C. H., & Dayan, P. (1992). Q-learning. *Machine Learning*, *8*(3), 279–292. https://doi.org/10.1007/BF00992698

Wei, H., Zheng, G., Gayah, V., & Li, Z. (2019). *A Survey on Traffic Signal Control Methods*.

Wiering, M. A. (2000). Multi-agent reinforcement learning for traffic light control. In *In Machine Learning: Proceedings of the Seventeenth International Conference* (pp. 1151–1158).

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, *8*(3), 229–256. https://doi.org/10.1007/BF00992696

Wu, Y.-J., Zhang, G., & Wang, Y. (2012). Link-Journey Speed Estimation for Urban Arterial Performance Measurement Using Advance Loop Detector Data under Congested Conditions. *Journal of Transportation Engineering*, *138*(11), 1321–1332. https://doi.org/10.1061/(ASCE)TE.1943-5436.0000429

Xu, L.-H., Xia, X.-H., & Luo, Q. (2013). The study of reinforcement learning for traffic self-adaptive control under multiagent markov game environment. *Mathematical Problems in Engineering*, *2013*.

Yang, S., & Wu, Y.-J. (2018). Travel mode identification using bluetooth technology. *Journal of Intelligent Transportation Systems*, *22*(5), 407–421.

Yu, R., & Abdel-Aty, M. (2014). An optimal variable speed limits system to ameliorate traffic safety risk. *Transportation Research Part C: Emerging Technologies*, *46*, 235–246. https://doi.org/https://doi.org/10.1016/j.trc.2014.05.016

Yuan, J., & Abdel-aty, M. (2018). Approach-level real-time crash risk analysis for signalized intersections. *Accident Analysis and Prevention*, *119*(April), 274–289. https://doi.org/10.1016/j.aap.2018.07.031

Yuan, J., & Abdel-Aty, M. (2018). Approach-level real-time crash risk analysis for signalized intersections. *Accident Analysis & Prevention*, *119*, 274–289. https://doi.org/https://doi.org/10.1016/j.aap.2018.07.031

Yuan, J., Abdel-Aty, M., Gong, Y., & Cai, Q. (2019a). Real-Time Crash Risk Prediction using Long Short-Term Memory Recurrent Neural Network. *Transportation Research Record*, 0361198119840611.

Yuan, J., Abdel-Aty, M., Gong, Y., & Cai, Q. (2019b). Real-Time Intersection Crash Risk Prediction Using Long Short-Term Memory Recurrent Neural Networks. *Transportation Research Record: Journal of the Transportation Research Board*.

Yuan, J., Abdel-aty, M., Wang, L., Lee, J., Yu, R., & Wang, X. (2018). Utilizing bluetooth and adaptive signal control data for real-time safety analysis on urban arterials. *Transportation Research Part C*, *97*(October), 114–127. https://doi.org/10.1016/j.trc.2018.10.009

Yue, L., Abdel-Aty, M., Lee, J., & Farid, A. (2019). Effects of Signalization at Rural Intersections Considering the Elderly Driving Population. *Transportation Research Record*, *2673*(2), 743–757. https://doi.org/10.1177/0361198119825834

Yue, L., Abdel-Aty, M., Wu, Y., & Wang, L. (2018). Assessment of the safety benefits of vehicles' advanced driver assistance, connectivity and low level automation systems. *Accident Analysis & Prevention*, *117*, 55–64. https://doi.org/https://doi.org/10.1016/j.aap.2018.04.002

Yue, L., Abdel-Aty, M., Wu, Y., Zheng, O., & Yuan, J. (2020). In-depth approach for identifying crash causation patterns and its implications for pedestrian crash prevention. *Journal of Safety Research*, *73*, 119–132. https://doi.org/https://doi.org/10.1016/j.jsr.2020.02.020

Zhang, S., Abdel-Aty, M., Yuan, J., & Li, P. (2020). Prediction of Pedestrian Crossing Intentions at Intersections Based on Long Short-Term Memory Recurrent Neural Network. *Transportation Research Record*, 0361198120912422. https://doi.org/10.1177/0361198120912422

Zhu, L., Li, K., Liu, Z., Wang, F., & Tang, K. (2019). A Group-Based Signal Timing Optimization Model Considering Safety for Signalized Intersections with Mixed Traffic Flows. *Journal of Advanced Transportation*, *2019*, 1–13. https://doi.org/10.1155/2019/2747569