Analyzing the Robustness of Adaptive Traffic Control System Using Reinforcement Learning for Urban Traffic Flow

Preprint · July 2024

DOI: 10.13140/RG.2.2.33416.23040

1 author:

Ferdous Al Rafi
Bangladesh University of Engineering and Technology

ANALYZING THE ROBUSTNESS OF ADAPTIVE TRAFFIC CONTROL SYSTEM USING REINFORCEMENT LEARNING FOR URBAN TRAFFIC FLOW

Ferdous Al Rafi*[1]

[1] *Department of Civil Engineering, Bangladesh University of Engineering and Technology, Bangladesh,*
e-mail: 1904089@ce.buet.ac.bd

***Corresponding Author**

Abstract

This study investigated the robustness of reinforcement learning (RL) based adaptive traffic control systems (ATCS) in managing unseen traffic patterns and conditions. This research evaluated the performance of these systems by analyzing their ability to adapt and recover to changes in traffic using the microsimulation software SUMO. Two distinct traffic scenarios were prepared in simulation to evaluate performance: a synthetic scenario based on a 4x4 grid network and a real-world scenario modeled after the city of Ingolstadt, Germany. Each scenario included various cases representing different traffic patterns and conditions such as morning rush hour, evening congestion, special events, blocked roads, and faulty sensors. Following initial training on a specific case for each scenario, various RL models representing different ATCS systems underwent evaluation on unseen traffic events. The time to recover to the optimum level of performance of an RL model after encountering an unseen event, or the recovery time and the average queue length of all non-empty lanes over each timestep were used to evaluate the robustness of these models. Results of this study indicated that RL models generally performed well in managing changes in traffic flow but faced challenges with unseen conditions such as roadblocks and sensor failures. Furthermore, models with higher recovery times resulted in larger queue accumulation when encountering unseen traffic events in long-running cases.

Keywords

Adaptive Traffic Control System; Reinforcement Learning; Machine Learning; Traffic Simulation

1. Introduction

Traffic congestion is a serious issue in urban settings. Traffic delays from congestion result in lost work hours, higher fuel consumption, and increased economic costs [1–3]. To solve the problem of traffic congestion, one potential solution is to use Adaptive Traffic Control System (ATCS) to manage traffic in the city. However, implementing such systems is challenging due to the complexity of urban traffic flow [4, 5]. With this respect, many researchers explored the use of reinforcement learning (RL) for ATCS systems [5–9]. Compared to other systems, RL-based systems promise the ability to learn from traffic patterns and control traffic lights accordingly. However, current reinforcement learning (RL) models experience reduced performance when there is a shift in the underlying traffic distribution, such as encountering an unseen traffic pattern or condition [10]. There is limited research investigating the performance of reinforcement learning-based adaptive traffic control systems in response to unseen traffic events.

This study seeks to fill this gap by examining the robustness of reinforcement learning-based adaptive traffic control systems. It accomplishes this by evaluating how well these systems manage unseen traffic patterns and conditions by comparing their performance after encountering changes in traffic patterns and conditions and measuring how much time is needed to recover to the optimum level of performance.

2. Related Work

There are various studies that touch on the topic of RL models dealing with unseen traffic patterns and conditions. Rodrigues et al. discussed the effects of missing traffic sensor data, blocked roads, and changing traffic demands in the context of RL models for traffic signal control, but the study only focused on a single intersection [11]. Mei et al. examined the problem of missing traffic sensor data and addressed the problem using data imputation [12]. Alegre et al. discussed the effect of non-stationary or shifting traffic patterns on RL models with tabular Q-Learning [10]. Some researchers investigated the use of an ontology-based traffic signal control RL model to address the issue of robustness to faulty traffic sensor data and unseen traffic patterns [13–15]. Yoon et al. proposed using a graph-centric state representation to improve the performance of RL models for unseen traffic patterns [16]. Some researchers explored using meta-reinforcement learning for more efficient training for unseen traffic patterns [17–19]. Han et al. examined using adversarial reinforcement learning to improve generalization ability which helps in improving robustness by more efficient training in unseen events [20].

This study differs from other works in that it focuses on the methodology of analyzing the performance of RL-based ATCS systems, focusing on the change in performance over time. For analyzing this aspect, a performance metric called recovery time is introduced in this paper.

3. Methodology

In this section, the methodology for analyzing the robustness of RL-based ATCS systems is discussed. Two scenarios were prepared having multiple cases representing various traffic patterns and conditions. Multiple models were evaluated for each case using the SUMO traffic simulation software [21] and the testing framework provided by LibSignal [22]. The RL models underwent initial training before evaluating their robustness using recovery time and average non-zero queue length.

3.1 Testing Scenarios

Two distinct traffic scenarios were prepared to evaluate performance:

1. Grid 4x4
2. Ingolstadt

The networks for these two scenarios are shown in [**Fig. 1**] and [**Fig. 2**]. Further details about the two scenarios are described in the following subsections. Each scenario included various cases representing different traffic patterns and conditions. The details of the scenarios and cases are described in **Table 1** below. Notably, events representing changes in traffic flow like morning rush hour, evening congestion, and special events are considered *traffic patterns* in this paper. Events representing changes in the environment like blocked roads and faulty sensors are considered *traffic conditions* in this paper.

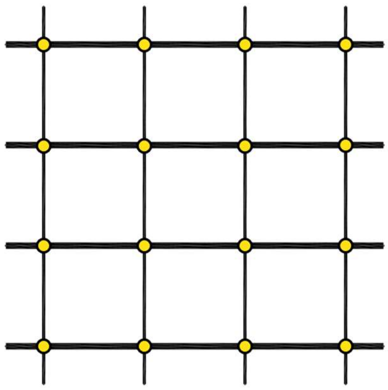


Fig. 1 Grid 4x4 scenario network. Circular yellow markers indicate the position of traffic signal controls.

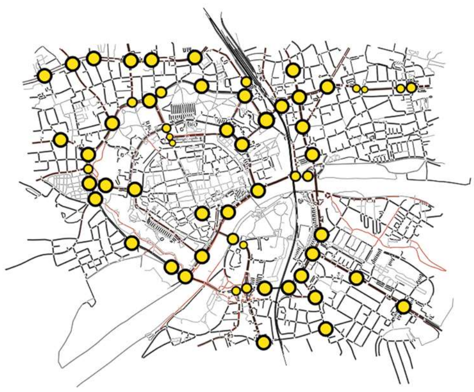


Fig. 2 Ingolstadt scenario network. Circular yellow markers indicate the position of traffic signal controls.

3.1.1 Grid 4x4

The Grid 4x4 scenario was based on a 4x4 grid network consisting of 16 traffic lights. This scenario used synthetic data for testing across ten cases, as shown in **Table 1**. Only having 16 traffic signal controls, the small size of the network made it easy to prototype and develop cases for evaluation. Five models from the LibSignal library (*IDQN* [23], *MPLight* [4], *CoLight* [24], *FixedTime,* and *MaxPressure* [25]) were used for this scenario.

The morning traffic cases involved inserting a vehicle every 3 seconds, moving from the edge nodes toward the center node. For the evening traffic cases, it followed the same pattern in reverse. For the blocked road cases, two roads were blocked, one near the center and one near the edges, preventisng vehicle passage. For the special event traffic case, vehicles moved from the uppermost to lowermost nodes at a rate of one every 3 seconds. For simulating faulty sensors in the faulty sensor cases, sensors had a 10% chance of failing to provide any data and all sensors added noise to the outputs based on a Poisson distribution with a mean of 0.5. The morning to evening traffic cases combined three patterns: morning traffic (0-4 hours), evening traffic (9-14 hours), and special event traffic (13-14 hours). Also, since the duration of these cases was 14 hours, these are long-running cases. Additionally, all cases included trips with random origins and destinations occurring every 5 seconds representing random traffic, which are added to the case traffic. The traffic comprised cars, buses, delivery vans, motorcycles, bicycles, and emergency vehicles.

3.1.2 Ingolstadt

The Ingolstadt scenario was based on real-world traffic data from the city of Ingolstadt, Germany, and consisted of 59 traffic lights. This scenario used real-world data for testing across six cases, as shown in **Table 1**. For ease of testing, the network used in this study is a simplified version of the original network [26]. This dataset was

Table 1 All traffic pattern cases for each scenario

Scenario	Case No.	1	2	3	4	5	6
Grid 4x4	**Description**	Morning Traffic	Morning Traffic Scaled Up by 20%	Morning Traffic Scaled Down by 20%	Case 1 with Blocked Road	Case 2 with Blocked Road	Evening Traffic
	Case No.	7	8	**9***	**10***	-	-
	Description	Case 6 with Special Event Traffic	Case 1 with Faulty Sensors	Morning to Evening Traffic (14 Hours)	Case 9 with Faulty Sensors	-	-
Scenario	**Case No.**	1	2	3	4	**5***	**6***
Ingolstadt	**Description**	Morning Traffic (9:00 AM-10:00 AM)	Noon Traffic (1:00 PM-2:00 PM)	Evening Traffic (6:00 PM-7:00 PM)	Case 1 with Faulty Sensors	Morning to Evening Traffic (8:00 AM-8:00 PM)	Case 5 with Faulty Sensors

* These are considered long-running cases in this paper because the duration of these cases is greater than 12 hours, other cases are only 1 hour long

chosen because it is the only publicly available dataset that contained 24-hour traffic data over multiple days. In addition, this dataset also recorded the actual traffic signal control activations used in the city. These qualities made it ideal for testing the robustness of RL models and comparing them with real-world performance. Three models from the LibSignal library (*IDQN*, *CoLight,* and *MaxPressure*) were used for this scenario. The *FixedTime* model used actual traffic signal control activations included in the scenario rather than the *FixedTime* model available from LibSignal.

Since the focus of this scenario involved understanding real-world impacts, no synthetic data were used. Traffic data from 16/09/2020 provided in the dataset was used for preparing the cases. The morning, evening, and evening traffic cases were simulated by running the simulation at the specified time mentioned in **Table 1**. There were no separate blocked roads and special event cases as the dataset did not have any traffic flow data representing these cases. However, there were two cases covering faulty sensors included in the scenario, as they did not involve modifying the existing traffic flow data. The morning to evening traffic cases were 12 hours long and they are long-running cases.

3.2 Initial Training of RL Models

There were five models (*IDQN*, *MPLight*, *CoLight*, *FixedTime*, and *MaxPressure*) that were tested in this study. *IDQN*, *MPLight,* and *CoLight* are RL models, and they are trainable. *FixedTime* and *MaxPressure* are non-RL models and were provided as a baseline. *IDQN*, also referred to as *DQN* in this paper, is a deep Q-learning model that works independently at each intersection. *MPLight* also works independently but it uses shared parameters across all the agents. Unlike *IDQN* and *MPLight*, *CoLight* model agents can communicate with each other. *FixedTime* is the traditional timing-based traffic signal model that has a fixed cycle time. *MaxPressure* is a model that works on the principle of the max pressure algorithm.

For the Grid 4x4 scenario, three of the models (*IDQN*, *MPLight*, *CoLight*) were trained on *Case 1* with an alternate random traffic distribution for 200 episodes. For the Ingolstadt scenario, two of the models (*IDQN* and *CoLight*) were trained on traffic data from 8:00 AM-9:00 AM for 100 episodes. For each scenario, the models were tested for all cases with training mode on. Training mode ensured that the models continued to learn and adjust to changes in traffic while the cases were running. The epsilon and learning start parameter was set to the minimum of individual models and 70 steps respectively after initial training was done. The epsilon and learning start parameters minimized the effects of randomness of an RL model in the early stages of training models in LibSignal.

3.3 Performance Metrics

Two metrics were used to evaluate the robustness of the models: recovery time and average non-zero queue length over each timestep. The details of these metrics are described in the following subsections. For measuring recovery time, the average travel time recorded by LibSignal was used to calculate it. However, in the case of the *FixedTime* model in the Ingolstadt scenario, the statistics file generated by SUMO was used to calculate it. This was done because to use the real-world traffic signal control activations contained in the Ingolstadt scenario, the simulation had to be run outside the LibSignal framework. The two average travel times differ by a small amount. For measuring the queue length, the Queue Output of SUMO was used to obtain the queue data for each lane.

3.3.1 Recovery Time

After an RL model faces an unseen traffic event, it takes some time to train in the unseen event before it reaches the optimum level of performance. This will be referred to as a model's recovery time. Different models yield different recovery times in different cases. To measure the recovery time, the model was trained on a case after initial training was done for a pre-specified number of episodes. Then the lowest average travel time was recorded and compared with the other travel times. If three consecutive travel times were below a certain threshold percentage, then the number of episodes to reach that window was the recovery time. If there were no windows where they did not fall under the threshold, then the recovery time was taken as the highest possible episodes for the scenario. However, at the start of the training of each case, even after taking precautions to minimize randomness, the average travel time was a little higher than what was expected after loading in a trained model. So when calculating the recovery time, the minimum recovery time was taken as 1 hour. It is worth mentioning that each episode has 3600 timesteps, each timestep representing 1 second in the simulation. So, each episode can be said to cover an hour of traffic flow. The recovery time was reported in hours rather than number of episodes.

For the Grid 4x4 scenario, the models were trained for 20 episodes from *Case 1* to *Case 8* after initial training was done. For the Ingolstadt scenario, the models were trained for 10 episodes from *Case 1* to *Case 4* after initial training was done.

3.3.2 Average Non-zero Queue Length

Average queue length is the average length of queues in vehicles in all lanes for a certain period. Average non-zero queue length is like the metric of queue length used in other papers with two differences. Firstly, it filters out the empty queues when calculating the average. This helps in the case of large networks like Ingolstadt, where only a small fraction of the roads experience significant traffic. Secondly, the queue length is observed over each timestep. This is helpful in the case of long-running cases (12 hours+) where this metric is easier to measure and compare the performance of the model in different periods.

For each scenario, the models were run for one episode with training mode for each case after the initial training was done. The data obtained indicated the performance of the model right after it was deployed after initial training, rather than the optimum level of performance. For the Grid 4x4 scenario, *Case 9* and *Case 10* were tested and for the Ingolstadt scenario, *Case 5* and *Case 6* were tested.

4. Results and Discussions

In this section, the results of the initial training and the evaluations of the models using recovery time and average non-zero queue length for all cases for each scenario are discussed. The training curve for training the models on each case after initial training is done is shown in [**Fig. 3**] and [**Fig. 4**]. The calculated recovery time for each scenario and model is shown in [**Fig. 5**]. The average non-zero queue length over each timestep for each scenario is shown in [**Fig. 6**].

4.1 Initial Training

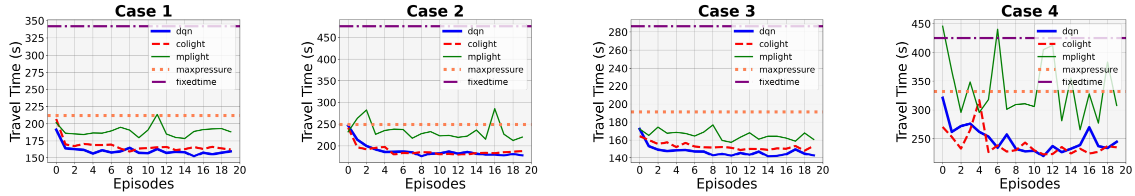
The final average travel time after initial training for all models for each scenario is listed in **Table 2**. Notably, *FixedTime* performed much better in the Ingolstadt scenario compared to the Grid 4x4 scenario. This happened because it used traffic signal data from the real city of Ingolstadt and the existing traffic control system is well optimized for the observed traffic. Even so, if properly trained, RL models like *IDQN* outperformed it. However, as will be demonstrated in the following subsections, it has some difficulties when encountering new traffic events and underperforms compared to the existing traffic control system in some cases.

Table 2 Final average travel time after initial training in seconds

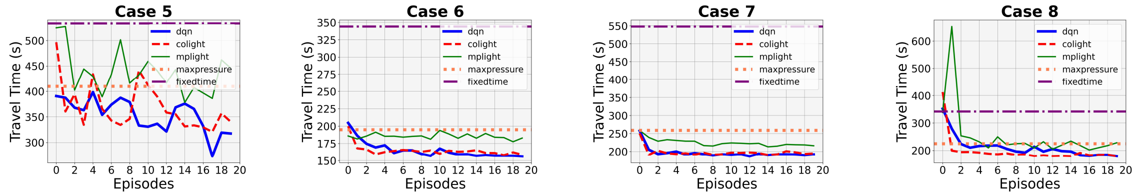
	IDQN	MPLight	CoLight	MaxPressure	FixedTime
Grid 4x4	158.0	200.7	162.8	224.9	341.9
Ingolstadt	301.7	-	392.1	480.17	351.1

4.2 Recovery Time of Models

In general, the models experienced an increase of 15% and 21% in average travel time compared to the optimum level for Grid 4x4 and Ingolstadt scenarios respectively across all cases.

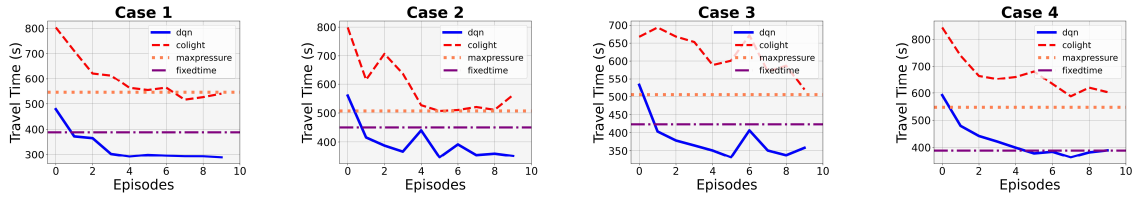


(a) Case 1 **(b)** Case 2 **(c)** Case 3 **(d)** Case 4



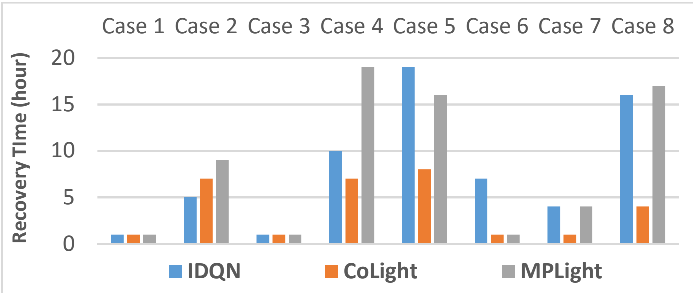
(e) Case 5 **(f)** Case 6 **(g)** Case 7 **(h)** Case 8

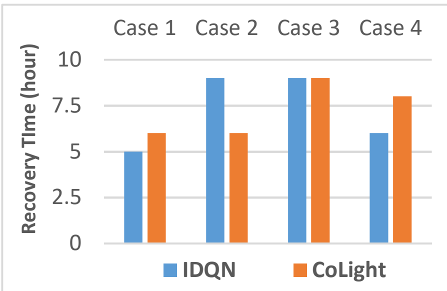
Fig. 3 Grid 4x4 case training curves for calculating the recovery time of the RL models. Three RL models are tested, and two non-RL models are included as baseline.



(a) Case 1 **(b)** Case 2 **(c)** Case 3 **(d)** Case 4

Fig. 4 Ingolstadt case training curves for calculating the recovery time of the RL models. Two RL models are tested, and two non-RL models are included as baseline.



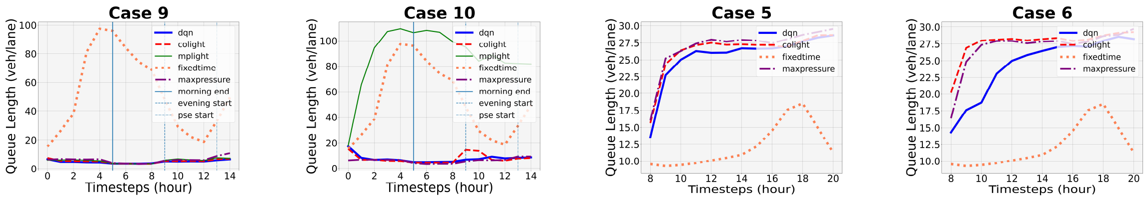


(a) Grid 4x4 Recovery Times **(b)** Ingolstadt Recovery Times

Fig. 5 Recovery Time for all cases for all models. The recovery time is obtained by comparing the lowest found average travel time and comparing with the other travel times during the case training process and noting the number of episodes or hours to reach the lowest average travel time.

Focusing on the cases, for the Grid 4x4 scenario, change in traffic pattern has a relatively low recovery time (*Case 1*, *Case 3*, *Case 6*, *Case 7*) except for *Case 2* as shown in [**Fig. 5(a)**]. The RL models needed more time to recover when faced with an increase in traffic flow. However, in the blocked road and faulty sensor cases, all models needed a higher recovery time. This indicates that RL models are relatively better at dealing with changes in traffic but struggle when dealing with unseen conditions like blocked roads and faulty sensors.

For the Ingolstadt scenario, it showed a slightly different picture. Here, change in traffic has a high recovery time across the board as shown in [**Fig. 5(b)**]. One possible explanation for this is that the difference in network topology and traffic flow data affected the recovery time of a model. Another possible explanation is that training the models just for a small period of one hour was not adequate to deal with change in traffic in the context of a large real-world network. Also worth noting is that the *IDQN* model achieved better average travel time than *FixedTime* for the shift in traffic cases (*Case 1* to *Case 3*) but suffered in the faulty sensor case where it performed worse than the baseline *FixedTime* model for around 5 hours as shown in [**Fig. 4**]. Faulty sensors had a larger effect on the performance drop compared to the shifts in traffic. This supports the same observation of RL models struggling with unseen conditions like in the Grid 4x4 scenario.



(a) Grid 4x4 Case 9 **(b)** Grid 4x4 Case 10 **(c)** Ingolstadt Case 5 **(d)** Ingolstadt Case 6

Fig. 6 Average non-zero queue length over each timestep for long-running cases. Models that manage to accumulate traffic slowly see a rapid and sustained increase in average queue length. Models with higher recovery times are better able to accommodate unseen traffic events.

Focusing on the models, for the Grid 4x4 scenario, *IDQN* and *CoLight* performed the best across all cases for reaching the optimum level of performance, in terms of average travel time, as seen in [**Fig. 3**]. *CoLight* had the lowest average recovery time across all cases at 3.75 hours. Notably, *CoLight* had the fastest recovery in the faulty sensor case as seen in [**Fig. 3(h)**] and [**Fig. 5**], in contrast to other models having a higher recovery time. One possible explanation for this is that the effects of missing information resulting from faulty sensors are mitigated when traffic signals communicate with each other as in *CoLight*.

For the Ingolstadt scenario, *IDQN* performed the best across all cases as observed in [**Fig. 4**]. However, in the case of faulty sensors, it performed worse than the baseline model of *FixedTime* initially. This supports the observation that RL models deal with changes in traffic flow better compared to unseen traffic conditions like faulty sensors as noted previously.

An interesting observation, although not directly related to robustness analysis, can be noted for the results of the *CoLight* model for the Ingolstadt scenario. *CoLight* performed poorly when faced with traffic shifts. Even in the initial training results **Table 2**, *CoLight* performed worse than one of the baseline models of *FixedTime*. The simpler model of *IDQN* where agents do not communicate with each other performed the best across all models. A possible explanation for this may be that *CoLight* struggles to optimize for large networks like the Ingolstadt scenario (59 traffic lights). In the LibSignal paper, for the Manhattan network containing 196 traffic signals, *IDQN* had the best average travel time with 1319.49 seconds compared with *CoLight* achieving 1493.42 seconds. With this respect, the results regarding *CoLight* in a large network like the Ingolstadt scenario in this study are consistent with other works.

4.3 Non-zero Average Queue Length for Models

For the Grid 4x4 scenario, all RL models had low queue length in *Case 9* as shown in [**Fig. 6(a)**]. In *Case 10*, *IDQN* and *CoLight* performed well in keeping queue lengths low but *MPLight* performed worse than one of the baseline models of *FixedTime* as shown in [**Fig. 6(b)**]. For *MPLight*, the queue length increased very quickly and remained high. Another interesting observation is that for the *IDQN* and *CoLight* models, there was a spike in queue length right at the start of the case, which indicated increased average queue length due to faulty sensors. However, these models brought the average queue length down in the next three hours. *CoLight* experienced another spike right when the evening traffic started and recovered to similar levels to *IDQN* later. One possible explanation for this observation can be found by looking at the minimum average recovery time of the models in **Table 3**. *MPLight* has the highest recovery time compared to the other models. A lower recovery time means the model will be able to decrease the queue length before it starts accumulating. So, the reason the *MPLight* model maintained a high queue throughout *Case 10* might be that due to its high recovery time, it was not able to recover to its optimum level fast enough to counter the vehicle accumulation. For *IDQN* and *CoLight* models, their low recovery time enabled them to deal with the queue spike encountered at the start and at the evening start event of the case. So, lower recovery time leads to lower queue accumulation and vice versa.

For the Ingolstadt scenario, the effect of recovery time was more pronounced. Although the *IDQN* model performed the best when trained on a single case for 10 episodes as seen in [**Fig. 4**], due to the high recovery time, even the best-performing RL model performed poorly in terms of maintaining low queue length when tested in a long-running case like *Case 3* and *Case 4* as shown in [**Fig. 6(c)**] and [**Fig. 6(d)**]. The curves show a similar pattern when running the *MPLight* model in *Case 10* of the Grid 4x4 scenario. Like in the Grid 4x4 scenario, higher recovery time led to higher queue accumulation in the Ingolstadt scenario.

Table 3 Minimum Average Recovery Time in hours for all cases for each scenario for all models

	IDQN	**MPLight**	**CoLight**
Grid 4x4	7.9	8.5	3.8
Ingolstadt	7.3	-	7.3

4.4 Limitations

In terms of limitations, the primary weakness of this study is that it focused on breadth rather than depth for analyzing robustness. There were five models evaluated using 16 cases representing various traffic events across two scenarios for understanding the general effects of unseen traffic events. But all the cases were run once rather than being conducted multiple times. However, due to the difficulty of running many experiments, especially for large networks like Ingolstadt, this arrangement was needed to accomplish this project with limited time and resources. So, the obtained recovery time can become more accurate by running the cases multiple times and taking the average value of the performance metrics for multiple runs of each case.

5. Conclusion

This study examines the robustness of reinforcement learning-based adaptive traffic control systems, analyzing their performance and recovery time after encountering unseen traffic patterns and conditions. It uses both synthetic and real-world data to conduct the analysis. The results of the analysis indicate that RL models are good at dealing with fluctuations in traffic flow but struggle when dealing with unseen traffic conditions like blocked roads and faulty sensors. Moreover, in long-running cases, higher recovery time leads to increased queue accumulation when encountering new traffic events. When deploying RL systems in practical settings, these ideas would aid in building the robustness needed to meet real-world standards.

Several directions for future research exist. One of them is making the procedure for calculating recovery time more rigorous and repeatable. Another interesting approach is to find the relationship between recovery time and queue accumulation. Additionally, beyond the robustness of RL models, investigating how network size influences RL model performance and optimizing RL models for larger networks presents another promising area of study.

The raw data and the code for this study can be found in https://github.com/Red-Pheonix/robust_data.

Acknowledgment

The author would like to thank Dr. Md. Shamsul Hoque, Professor, Department of Civil Engineering, Bangladesh University of Engineering and Technology for providing valuable advice and guidance for conducting this research.

References

1. Pishue B (2024) 2023 INRIX Global Traffic Scorecard
2. Schrank D, Albert L, Jha K, Eisele B (2024) 2023 URBAN MOBILITY REPORT
3. Chakraborty S (2010) Traffic congestion in Dhaka city and its economic impact. Traffic congestion in Dhaka city: Its impact on business and some remedial measures
4. Chen C, Wei H, Xu N, Zheng G, Yang M, Xiong Y, Xu K, Li Z (2020) Toward A Thousand Lights: Decentralized Deep Reinforcement Learning for Large-Scale Traffic Signal Control. Proceedings of the AAAI Conference on Artificial Intelligence 34:3414–3421. https://doi.org/10.1609/AAAI.V34I04.5744
5. Wang Y, Yang X, Liang H, Liu Y (2018) A Review of the Self-Adaptive Traffic Signal Control System Based on Future Traffic Environment. J Adv Transp 2018:1096123. https://doi.org/10.1155/2018/1096123
6. Jing P, Huang H, Chen L (2017) An Adaptive Traffic Signal Control in a Connected Vehicle Environment: A Systematic Review. Information 2017, Vol 8, Page 101 8:101. https://doi.org/10.3390/INFO8030101
7. Miletić M, Ivanjko E, Gregurić M, Kušić K (2022) A review of reinforcement learning applications in adaptive traffic signal control. IET Intelligent Transport Systems 16:1269–1285. https://doi.org/10.1049/ITR2.12208
8. Wei H, Zheng G, Gayah V, Li Z (2021) Recent Advances in Reinforcement Learning for Traffic Signal Control. ACM SIGKDD Explorations Newsletter 22:12–18. https://doi.org/10.1145/3447556.3447565
9. Noaeen M, Naik A, Goodman L, Crebo J, Abrar T, Abad ZSH, Bazzan ALC, Far B (2022) Reinforcement learning in urban network traffic signal control: A systematic literature review. Expert Syst Appl 199:116830. https://doi.org/10.1016/J.ESWA.2022.116830
10. Alegre LN, Bazzan ALC, da Silva BC (2021) Quantifying the Impact of Non-Stationarity in Reinforcement Learning-Based Traffic Signal Control. PeerJ Comput Sci 7:1–20. https://doi.org/10.7717/PEERJ-CS.575/TABLE-1
11. Rodrigues F, Azevedo CL (2019) Towards Robust Deep Reinforcement Learning for Traffic Signal Control: Demand Surges, Incidents and Sensor Failures. 2019 IEEE Intelligent Transportation Systems Conference, ITSC 2019 3559–3566. https://doi.org/10.1109/ITSC.2019.8917451

12. Mei H, Li J, Shi B, Wei H (2023) Reinforcement Learning Approaches for Traffic Signal Control under Missing Data. IJCAI International Joint Conference on Artificial Intelligence 2023-August:2261–2269. https://doi.org/10.24963/ijcai.2023/251
13. Ghanadbashi S, Golpayegani F (2021) An Ontology-Based Intelligent Traffic Signal Control Model. IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC 2021-September:2554–2561. https://doi.org/10.1109/ITSC48978.2021.9564962
14. Ghanadbashi S, Safavifar Z, Taebi F, Golpayegani F (2024) Handling uncertainty in self-adaptive systems: an ontology-based reinforcement learning model. J Reliab Intell Environ 10:19–44. https://doi.org/10.1007/S40860-022-00198-X/METRICS
15. Ghanadbashi S, Golpayegani F (2022) Using ontology to guide reinforcement learning agents in unseen situations: A traffic signal control system case study. Applied Intelligence 52:1808–1824. https://doi.org/10.1007/S10489-021-02449-5/TABLES/6
16. Yoon J, Ahn K, Park J, Yeo H (2021) Transferable traffic signal control: Reinforcement learning with graph centric state representation. Transp Res Part C Emerg Technol 130:103321. https://doi.org/10.1016/J.TRC.2021.103321
17. Zang X, Yao H, Zheng G, Xu N, Xu K, Li Z (2020) MetaLight: Value-Based Meta-Reinforcement Learning for Traffic Signal Control. Proceedings of the AAAI Conference on Artificial Intelligence 34:1153–1160. https://doi.org/10.1609/AAAI.V34I01.5467
18. Zhu L, Peng P, Lu Z, Tian Y (2023) MetaVIM: Meta Variationally Intrinsic Motivated Reinforcement Learning for Decentralized Traffic Signal Control. IEEE Trans Knowl Data Eng 35:11570–11584. https://doi.org/10.1109/TKDE.2022.3232711
19. Huang X, Member DW, Jenkin M, Boulet B (2021) ModelLight: Model-Based Meta-Reinforcement Learning for Traffic Signal Control
20. Han G, Han Y, Wang H, Ruan T, Li C (2024) Coordinated Control of Urban Expressway Integrating Adjacent Signalized Intersections Using Adversarial Network Based Reinforcement Learning Method. IEEE Transactions on Intelligent Transportation Systems 25:1857–1871. https://doi.org/10.1109/TITS.2023.3314409
21. Lopez PA, Behrisch M, Bieker-Walz L, Erdmann J, Flotterod YP, Hilbrich R, Lucken L, Rummel J, Wagner P, Wiebner E (2018) Microscopic Traffic Simulation using SUMO. IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC 2018-November:2575–2582. https://doi.org/10.1109/ITSC.2018.8569938
22. Mei H, Lei X, Da L, Shi B, Wei H (2023) Libsignal: an open library for traffic signal control. Mach Learn. https://doi.org/10.1007/S10994-023-06412-Y
23. Liang X, Du X, Wang G, Han Z (2019) A Deep Reinforcement Learning Network for Traffic Light Cycle Control. IEEE Trans Veh Technol 68:1243–1253. https://doi.org/10.1109/TVT.2018.2890726
24. Wei H, Xu N, Zhang H, Zheng G, Zang X, Chen C, Zhang W, Zhu Y, Xu K, Li Z (2019) Colight: Learning network-level cooperation for traffic signal control. International Conference on Information and Knowledge Management, Proceedings 1913–1922. https://doi.org/10.1145/3357384.3357902
25. Varaiya P (2013) Max pressure control of a network of signalized intersections. Transp Res Part C Emerg Technol 36:177–195. https://doi.org/10.1016/J.TRC.2013.08.014
26. Harth M, Langer M, Bogenberger K (2022) Automated Calibration of Traffic Demand and Traffic Lights in SUMO Using Real-World Observations. SUMO Conference Proceedings 2:133–148. https://doi.org/10.52825/SCP.V2I.120