Journal of Al-Azhar University Engineering Sector

Vol. 20, No. 75, April 2025, 511- 526

INTELLIGENT TRAFFIC SIGNAL CONTROL USING SPATIO-TEMPORAL DATA AND REINFORCEMENT LEARNING

Marwa Mohammed Saif*, Hussien Sayed Tantawy, Ashraf El-Marakby

Systems & Computers Department, Faculty of Engineering, Al-Azhar Unversity, Nasr City, Cairo, Egypt

***Correspondence:** marwamohammed.m2020@gmail.com

Citation:

M.M. Saif, H.S. Tantawy, A. El-Marakby, "Intelligent Traffic Signal Control Using Spatio-Temporal Data and Reinforcement Learning", Journal of Al-Azhar University Engineering Sector, vol. 20, pp. 511-526, 2025.

Received: 20 October 2025

Revised: 19 December 2024

Accepted: 04 January 2025

DOI: 10.21608/auej.2025.329865.1723

ABSTRACT

Intelligent Traffic Control System (ITCS) is an integral part of modern transportation system, helping to maintain smooth and safe traffic flow while reducing pollution. In this paper, we propose an intelligent traffic control system for multi-intersection networks that aims to improve traffic control systems by adjusting signal light timings to reduce waiting times and considering vehicle types and priorities. The model combines Reinforcement Learning (RL) to obtain optimal control policies, Graph Convolutional Networks (GCN) to capture spatial dependencies, Long Short-Term Memory (LSTM) to capture temporal dependencies, and Genetic Algorithms (GA) to enhance the deep network weights quickly and escape local optima. The experiment evaluates the effectiveness of various RL-based models in traffic management by evaluating the impact of GA and prioritization on ITCS models. Models are trained/tested using synthetic traffic data generated with the SUMO tool on three different-sized networks: Manhattan, Suzhou, and Cairo, with various vehicle types. The results demonstrate the distinct improvements of the LSTM-GCN-GA model in reducing waiting times. When compared with traditional models such as the Pre-Time model as in the Manhattan network, it reduced the waiting time by up to 84.81% for all vehicles and by up to 92.46% for priority vehicles. The genetic algorithm integration reduced the waiting time by up to 26.39% for all vehicles and by up to 80.21% for priority vehicles. Adding vehicle priority reduced the waiting time by up to 33.1% for all vehicles and by up to 83.82% for priority vehicles. Applying this model in real-world applications can enhance neural network efficiency, which optimize traffic flow, reduce congestion, and improve road safety.

KEYWORDS: Vehicle Priority, Spatial–Temporal dependencies, Multiple Intersections, Long Short-Term Memory, Graph Convolutional Networks, Genetic Algorithms, Traffic Control System.

التحكم الذكي في إشارات المرور باستخدام البيانات الزمانية والمكانية والتعليم المعزز

مروة محمد سيف*، حسين سيد طنطاوي، أشرف المراكبي

قسم النظم والحاسبات، كلية الهندسة، جامعة الأزهر، مدينة نصر، القاهرة، مصر.

***البريد الاليكتروني للباحث الرئيسي :** marwamohammed.m2020@gmail.com

الملخص

نظام التحكم الذكي في حركة المرور (ITCS) هو جزء لا يتجزأ من نظام النقل الحديث، حيث يساهم في الحفاظ على تدفق حركة المرور بسلاسة وأمان، مع تقليل التلوث. في هذه الورقة، نقترح نظامًا مركزيًا ذكيًا للتحكم في حركة المرور لشبكات متعددة التقاطعات يهدف إلى تحسين أنظمة التحكم في حركة المرور من خلال ضبط توقيتات إشارات المرور لتقليل أوقات الانتظار مع مراعاة أنواع المركبات وأولوياتها المختلفة. يجمع النموذج بين التعليم المعزز (RL) للحصول على سياسات التحكم المثلى، وشبكات التلافيف البيانية (GCN) لالتقاط التبعيات المكانية، والذاكرة الطويلة والقصيرة المدى (LSTM) لالتقاط التبعيات الزمنية، والخوارزميات الجينية (GA) لتعزيز أوزان الشبكة العميقة بسرعة وتجنب الأمثل المحلي (Local Optima). تم تصميم التجربة لتقييم فعالية تأثير الخوارزمية الجينية وإعطاء الأولوية على النماذج المختلفة. تم تدريب/ اختبار النماذج باستخدام بيانات حركة المرور اصطناعية تم إنشاؤها بواسطة أداة محاكاة التنقل الحضري (SUMO) تشمل أنواع مختلفة من المركبات تم توليدها بأوقات وفترات مغادرة مختلفة على ثلاث شبكات مختلفة الحجم: مانهاتن وسوتشو والقاهرة. أظهرت النتيجة تحسينات ملحوظة لنموذج LSTM-GCN-GA في تقليل أوقات الانتظار. بالمقارنة نموذج LSTM-GCN-GA مع نماذج التحكم التقليدية مثل نموذج الوقت الثابت ففي شبكة مانهاتن، فقد قلل من وقت الانتظار بنسبة تصل إلى ٨٤,٨١٪ لجميع المركبات وبنسبة تصل إلى ٩٢,٤٦٪ للمركبات ذات الأولوية. وقد أدى اضافة الخوارزمية الجينية إلى تقليل وقت الانتظار بنسبة تصل إلى ٢٦,٣٩٪ لجميع المركبات وبنسبة تصل إلى ٨٠,٢١٪ للمركبات ذات الأولوية. كما أدى إخذ أولوية المركبات بالاعتبار إلى تقليل وقت الانتظار بنسبة تصل إلى ٣٣,١٪ لجميع المركبات وبنسبة تصل إلى ٨٣,٨٢٪ للمركبات ذات الأولوية. إن تطبيق هذا النموذج في تطبيقات في العالم الحقيقي يمكن أن يعزز كفاءة الشبكة العصبية، مما يؤدي إلى تحسين تدفق حركة المرور، وتقليل الازدحام، وتحسين سلامة الطرق.

الكلمات المفتاحية : أولوية المركبات، التبعيات الزمانية والمكانية، تقاطعات متعددة، LSTM، GCN، الخوارزمية الجينية، نظام تحكم مرور.

1. INTRODUCTION

Traffic congestion is a critical problem for the large and developing cities worldwide. With economic growth and rapid urbanization, traffic congestion levels have increased, leading to negative impacts on city growth, development, and the environment. Moreover, longer travel times, increased risk of road accidents, and high fuel consumption.

Traditionally, efforts to reduce congestion have focused on expanding transportation infrastructure such as adding lanes and extending road construction. However, this strategy is not only costly but might also prove inefficient within vast road networks. Therefore, there is a need to develop intelligent traffic control systems that can be implemented on existing road networks to reduce congestion more efficiently. The Intelligent Traffic Control System (ITCS) includes many systems, such as Incident Management [1], Traffic Flow Prediction [2-3], and Traffic Signals Control [4].

In this paper, we focus on a traffic control system to adjust signal light timings. The traffic signal control system's main aim is to reduce traffic congestion and its impacts by optimizing traffic signal parameters, including: Phase Sequence arranges signal phases in the cycle, Green duration (or signal timing/split control) sets the seconds for a specific traffic movement, Cycle Length determines the time for a signal's single cycle, and Offset coordinates nearby intersections [4-5].

Signal timing control approaches are classified into three categories: Pre-time (fixed-time), actuated, and adaptive. This classification is mainly based on the data type and algorithms utilized to enhance traffic signal planning. Pre-time relies on fixed green splits determined by historical traffic demand, lacking real-time adjustment and effective control. Actuated uses sensors like induction loops and cameras to detect requests for green time, with the controller adapting cycle length based on historical data and extending green time for detected vehicles up to the maximum predefined green time. Adaptive approaches also use sensors but dynamically optimize signal timing with real-time traffic data [4-6].

In recent years, increasing the traffic volume on the roads has led to a rise in traffic congestion, delays, and accidents. These problems have not only inconvenienced commuters but have also increased economic losses. In response to these issues, Intelligent Traffic Control Systems (ITCSs) have been developed and designed to resolve these challenges. They aim to enhance the efficiency and safety of traffic flow at intersections.

Many techniques have been used to optimize traffic control, such as the adaptive linear-quadratic regulator (LQR) [4], the scheduling algorithm [7], fuzzy logic [8-10], and IoT [11-14]. Among these techniques, reinforcement learning (RL) [15-18], RL with graph convolutional neural networks (GCN) [19-21], RL with recurrent neural networks (RNN) [22-23], GCN with RNN [2, 24-25], and genetic algorithms (GA) [26] have demonstrated the possible to enhance and optimize traffic control systems.

Many research studies focus on managing traffic flow at individual intersections and rely only on lane density, neglecting the impact of neighboring roads, as in [8-9]. In urban areas, the network is more complex, and designing an effective ITCS is a challenging task, requiring consideration of various factors such as the number and density of intersections, road configurations, and pedestrian movements. Furthermore, the ITCS should be scalable, adaptable, and capable of managing dynamic and uncertain traffic conditions. Therefore, researchers have been developing new algorithms and models to enable ITCSs to optimize traffic flow, reduce congestion, minimize delays, and improve multiple intersections safety, as in [19-20].

These studies have demonstrated that the proposed approaches are effective in real-world traffic scenarios, resulting in significant improvements in traffic prediction and flow. However, several challenges remain to be resolved. These challenges include enhancing model efficiency to facilitate the learning process and developing more efficient training algorithms for managing traffic lights with diverse vehicle types across the network.

This paper handles the inefficiencies of current multi-intersection traffic control systems, such as high waiting times and traffic flow issues, which do not considering traffic density and vehicle types. The objective is to develop an intelligent traffic control system that optimizes signal timing by integrating traffic density and vehicle types, taking into account the complex spatio-temporal dynamics dependencies in networks.

Here are some of the key contributions of this paper:

- Propose an intelligent traffic control system based on RL integrating GCN, LSTM, and GA.
- Demonstrating the effectiveness of integrating RL, GCN, LSTM, and GA for enhancing traffic control systems.
- Evaluating the proposed approach in simulated traffic networks, demonstrating a significant reduction in congestion compared to alternatives, while considering various vehicle types.
- Executing experiments on three diverse networks with varying intersection and road topologies, specifically Suzhou, Manhattan, and Cairo.

2. METHODOLOGY AND DESIGN

2.1. Intelligente Traffic Control (ITC)

This paper focuses on an intelligent control method for managing traffic in a multi-intersection network, as illustrated in **Fig. 1**. The model collects traffic data from the roads within the network into a centralized unit. Then it processes this data and transmits the appropriate phase for each traffic light at the intersection, as illustrated in **Fig. 2**, which shows the ITC model input and output.

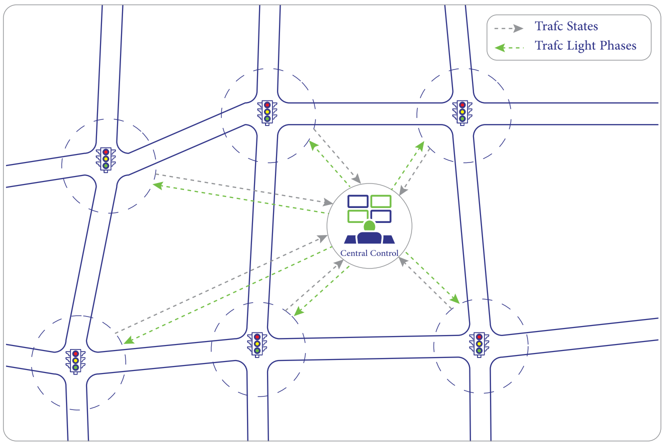

Fig. 1: Intelligent Traffic Control Model

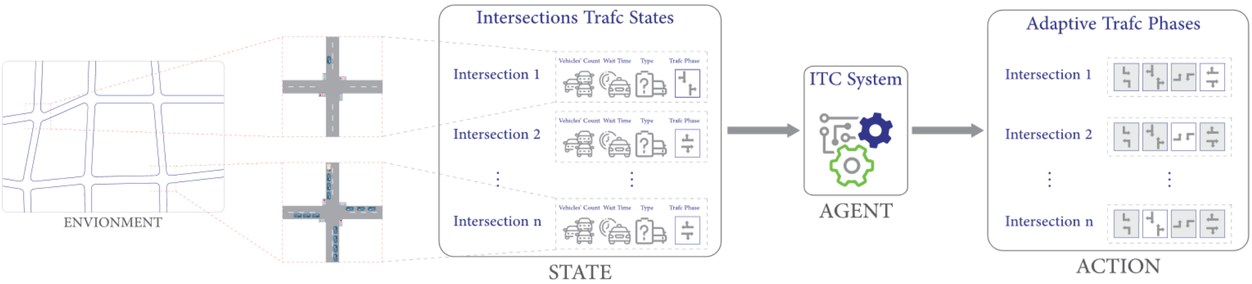

Fig. 2: Input and Output of Intelligent Traffic Control Model

2.2. ITC Model Design

The intelligent traffic control model architecture, illustrated in **Fig. 3**, includes three primary components: Deep Reinforcement Learning, Graph Convolution Long Short-Term Memory Network, and Genetic Algorithm. These approaches are utilized to train the model to observe and provide optimized traffic phases.

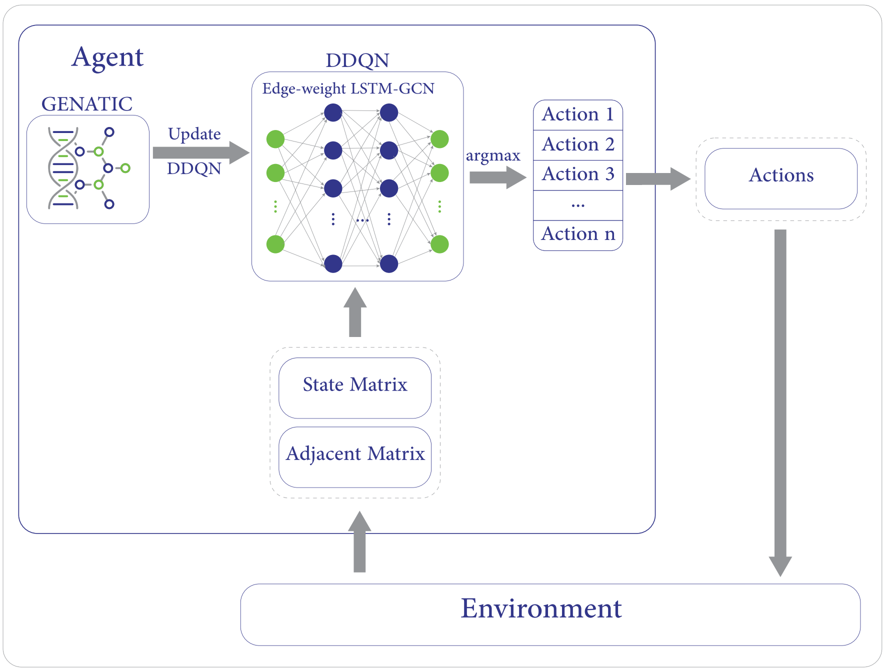

Fig. 3: Intelligent Traffic Control Architecture

2.2.1. Reinforcement Learning

Reinforcement Learning (RL) is a branch of machine learning that is based on a Markov Decision Process (MDP), which is defined by a (S; A; P; R; γ) tuple, where ***S*** is a set of discrete states in the environment, ***A*** is a set of actions that the agent can act, ***P*** is the state transition probability matrix, ***R*** denotes the reward function, and $\gamma \in [0,1]$ is a discount factor that is used to balance the importance of immediate and future rewards. The RL agent learns by interacting with the environment through a trial-and-error process, by using its own actions and experiences as feedback. This feedback can be positive or negative, as rewards or punishments, and its main objective is maximizing the reward function. RL is used in traffic control systems to interact with non-structured environments, improve responsiveness to unexpected events, continuously improve, and learn from experience.

The proposed model can control large-scale traffic intersections through a central agent. The agent obtains real-time traffic state observations through road sensors as GPS [10], video image processors [12], wireless [13], Google Traffic Map [14], ultrasonic sensors [18], or inductive loops [19]. In this study, statistical data generated by a simulation program was used for data collection. The agent then estimates the best reward based on the appropriate actions taken. The state space ***S***, action space ***A***, and reward ***R*** are defined as follows:

The ***state space S*** consists of the essential information required for traffic signal control, including the current signal, number, type, and waiting time of vehicles on all lanes entering the intersection. Equation 1 demonstrates the state space of the model.

$$state(t) = \{((v,\ wt *\ p)_1, \dots, (\dots)_l,\ TL_{current})_1, \dots, (\dots)_n\} \qquad \text{Eq. (1)}$$

Where l is the total number of lanes at the n intersections in the network, v is the total number of vehicles in each lane i at time t, wt is the average waiting time at lane i, and p is the priority of the vehicle based on its type. For instance, in a network consisting of 12 intersections, the state space ***S*** represented as a vector of size 12×21, where each row corresponds to an intersection and each column corresponds to the vector of the total vehicles, the average wait time, and the current traffic light signal at each lane of the intersection.

The **action space A** consists of different phases that the agent can act after observing the intersection's state at time t. For instance, a four-phase traffic light controller has an action space with four different phases represented by numbers from 0 to 3, as shown in **Fig. 4**. The different phases allow for various traffic flows to pass the intersection without collisions.

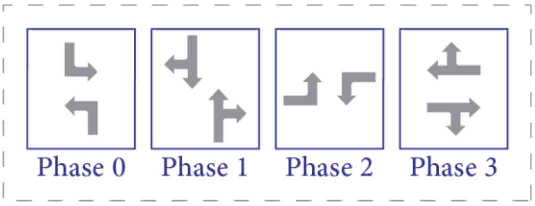

Fig. 4: An Example of Traffic Lights Actions

As in [19], we enable control of all intersections with a unified decoder-to-decoder action structure. An action probability matrix with dimensions $m \times n$ is used, where n represents the number of intersections and m represents the number of phases of each signal. This matrix satisfies $\forall a \in \boldsymbol{A}$ and is multiplied by a *mask matrix* to ensure correct control of each signal, even when different traffic lights have different phase numbers.

The reward R is determined by evaluating the agent's action after observing the state at time t. It is defined as the reduction in the average waiting time R_{wt} calculated using Equation 2, and the average emergency waiting time R_{wt_em} calculated using Equation 3.

$$R_{wt} = \sum_i avg\_WT(intersection_i^{t-1}) - \sum_j avg\_WT(intersection_j^{t}) \quad \text{Eq. (2)}$$

$$R_{wt\_em} = \sum_i avg\_WT_{em}(intersection_i^{t-1}) - \sum_j avg\_WT_{em}(intersection_j^{t}) \quad \text{Eq. (3)}$$

Intuitively, if the agent performs a "correct" action, the total accumulative average waiting time from the last step will be reduced, that is a *large* R_{wt} reward. On the other hand, if the action is "not correct", a *small* R_{wt} reward will be returned. Similarly, for the total accumulative average emergency waiting time R_{wt_em}. The total reward is calculated using Equation 4:

$$R_t = R_{wt} + R_{wt\_em} \quad \text{Eq. (4)}$$

We have chosen Double Deep Q-Networks (DDQN) as our RL algorithm for optimization. To improve training efficiency and stability, we used techniques such as experience replay, target networks, and genetic algorithms. Mathematically, the DDQN was calculated using Equation 5:

$$Q(s, a, \acute{\theta}) = Q(s, a, \theta) + \beta[r_t + \gamma \max Q(\acute{s}, \acute{a}, \emptyset) - Q(s, a, \theta)] \quad \text{Eq. (5)}$$

Where θ, $\acute{\theta}$ and $\emptyset$ denote different learnable weights, while β represents the learning rate. The term r_t denotes the instant reward at time t, and $Q(\cdot)$ represents Q networks.

The loss function L is defined as the mean square error. Using Equation 6, we calculated the loss function:

$$L(\theta) = \left((r_t + \gamma \max Q(\acute{s}, \acute{a}, \emptyset^{target})) - \left(Q(s, a, \theta^{pred})\right)\right)^2 \quad \text{Eq. (6)}$$

2.2.2. Edge Weight Graph Convolution Long Short-Term Memory Network (EW- LSTM GCN):

Capturing both spatial and temporal features is essential for optimizing traffic. Since the spatial relations between intersections form a non-Euclidean graph, traditional approaches may not be sufficient. Graph Neural Networks (GNNs) provide a more generic framework to encode these topology node relations. Therefore, we apply edge-weighted GNNs to integrate spatial and distance information, capture cross-intersection relations, and traffic flow patterns. While LSTM is applied to capture the temporal dependencies of traffic data, such as the relationship between traffic flow at different time intervals. To capture both the spatial and temporal dependencies simultaneously, we utilize a combination of GCN and LSTM networks as in [2].

The proposed model consists of three GCN layers, followed by an LSTM layer. In the initial phase, the input data series are used as the network states, while the GCN is utilized to capture the topological structure of the urban road network, that enabling the extraction of spatial features. In the second phase, the LSTM is provided with the time series enhanced with spatial features, which facilitates the information exchange between the units to capture the temporal features, and subsequently determine the appropriate action. To prevent the risk of overfitting of neural networks, dropout layers are applied after each GCN layer.

For layer-wise representations, the functioning of Graph Convolutional Networks in the multi-intersections traffic optimization problem is shown as follows:

$$F^{(l+1)} = f\left(F^{(l)}, A\right), l \geq 1 \quad \text{Eq. (7)}$$

where $F^{(l)}$ denotes the feature map of layer l, which corresponds to an $N \times F^{(l)}$ feature matrix. Each row of this matrix represents a feature representation of a node, N number of intersections, and A represents the adjacent matrix of all junctions. As illustrated in **Fig. 5**, the network structure takes the current layer's features and adjacent matrix as input and produces the features of the next layer, then the spatial features of GCN feed into the LSTM to capture the temporal features and determine the appropriate action. The initial input of the network is denoted by X, where $F^{(0)} = X$.

Additionally, we implemented an edge-weighted graph convolutional encoder, following the approach in [19], which assigns specific weights to edges based on the lane length.

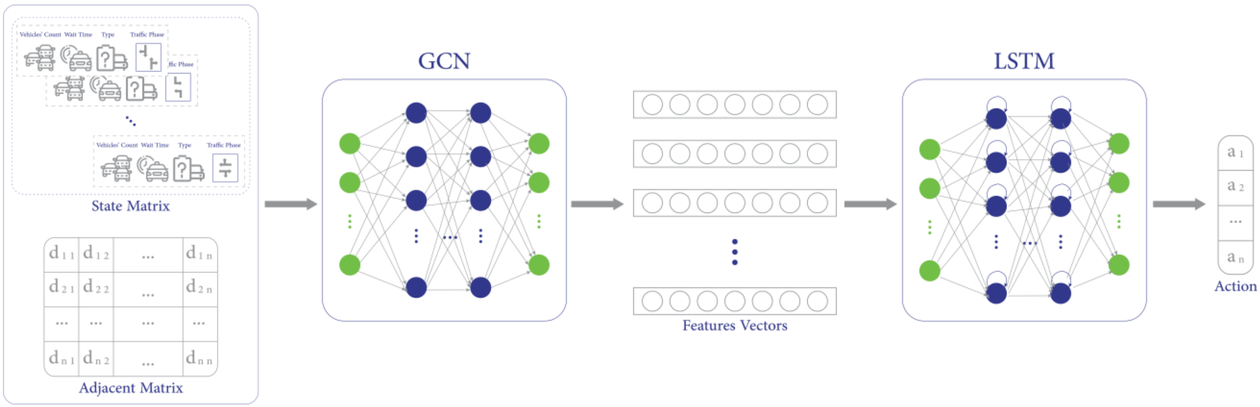

Fig. 5: GCN and LSTM Network Architecture for Traffic Control

The cells in LSTM are responsible for storing information from GCN output, which is then transmitted to an LSTM layer. The gates regulate the memory. The LSTM has three gates: the *Forget Gate*, which is responsible for discarding irrelevant data from the cell block, the *Input Gate*, which is responsible for determining which new information from the input should be stored in the cell state, and the *Output Gate*, which is responsible for determining which information to transmit from the cell state to the next hidden state in traffic control. The *Output Gate* also determines which information is no longer relevant and should be discarded, such as information on vehicles that have already passed through the intersection.

2.2.3. Genetic Algorithm

Genetic Algorithm (GA) is an adaptive heuristic search algorithm and a type of evolutionary algorithm (EA). It operates on the concept of passing the weights of two good neural networks, would be a better neural network. Based upon on the natural selection's ideas and genetics that keep the fittest. In such as this process keeps the strongest weights while the weakest are eliminated. GA is used in optimization and search problems to generate high-quality solutions. Utilizing GA for optimizing neural network weights offers several benefits: GA explores a broad solution space simultaneously, possibly discovering the best weight configurations. Its population-based approach, using crossover and mutation helps escape local optima.

We have integrated the GA into our traffic control model to accelerate the search for optimal phases and reduce overall waiting time in the network, as illustrated in **Fig. 6**. The model automatically uses GA when the total waiting time exceeds a predefined *threshold*. This constraint on GA utilization accelerates the network's training process, as GA demands a large amount of time.

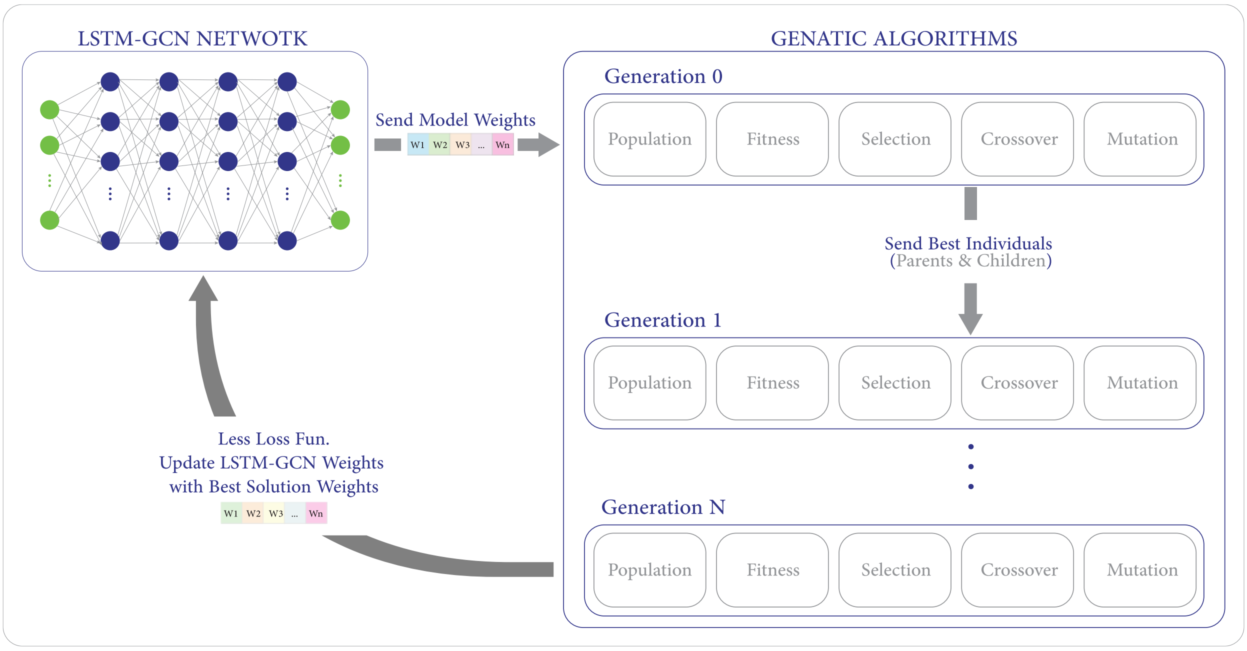

Fig. 6: Genetic Algorithm for LSTM-GCN Network's Weights Enhancement

In genetic algorithms, the initial population of possible solutions is referred to generation 0, and subsequent populations are denoted as generation 1, generation 2, and so on. The process of generating new generations continues until a specific condition is met, such as the maximum number of generations. The process to generate a new generation involves several stages. The different stages are:

Initialization: The GA process begins with generating a group of individuals called population, each represented by a vector of weights called a chromosome. In Generation 0, the first chromosome takes the weights from the online network of DDQN, and the rest generated randomly. From Generation 1 until Generation i, the individuals are taken from old parents and new children. These individuals are characterized by a group of parameters known as Genes.

Fitness: The individuals are evaluated using a fitness function that provides a fitness score. Individuals with higher fitness scores have a better chance of being selected for reproduction.

Selection: Individuals with the best fitness score are chosen for reproduction to enhance the weights of a deep neural network, then replaced the individuals with their offspring.

Reproduction: This phase involves creating offspring using two functions:

Crossover: create new weights to improve the network's performance by exchanging the weights of the two individuals at a random point.

Mutation: introduces random genes into the offspring (new children) to maintain population diversity by flipping bits in the chromosome to produce new offspring. This helps to resolve early convergence and enhance variety.

The objective of RL is to develop models that can navigate complex environments effectively, and make decisions based on learned knowledge, that enabling the agent to perform optimally. In this learning process, the loss as a metric to evaluate how well the model approximates the optimal values or actions. As illustrated in **Fig. 6**, when using GA to enhance the weights of a neural network, the vector of weights of the best solution obtained from GA that has best fitness, will used to update the weights of the *online network*. This update occurs only if only the loss between the GA model and the *target network* is lower than the loss of the *online network* of DDQN and the *target network*. This means that the GA model performs better in learning and predicting the optimal values or actions compared to the *online network* of DDQN.

3. EXPERIMENTS

This section presents the evaluation of integrating RL with various deep network models, including GCN and LSTM-GCN. It explores the impact of GA on deep networks learning, and the effect of considering vehicle priority at traffic control systems design. This evaluation is a critical measure in evaluating the model's performance and effectiveness.

3.1. Simulaton Settings

Simulating realistic traffic flows is critical for the optimal traffic reduction, especially during major events or in public areas like airports, schools, and hospitals. Additionally, simulating the impact of vehicles and intelligent transportation technologies is better for understanding their capability in reducing traffic flows. The Simulation of Urban Mobility (SUMO) platform, is designed for modeling and managing traffic flows in large microscopic networks. We utilize the SUMO platform to simulate three different types of networks: Suzhou with 12 intersections, Manhattan with 22 intersections, and Cairo with 57 intersections. **Fig. 7** shows the netwoks structures. Synthetic traffic data is applied during the simulation process for all networks.

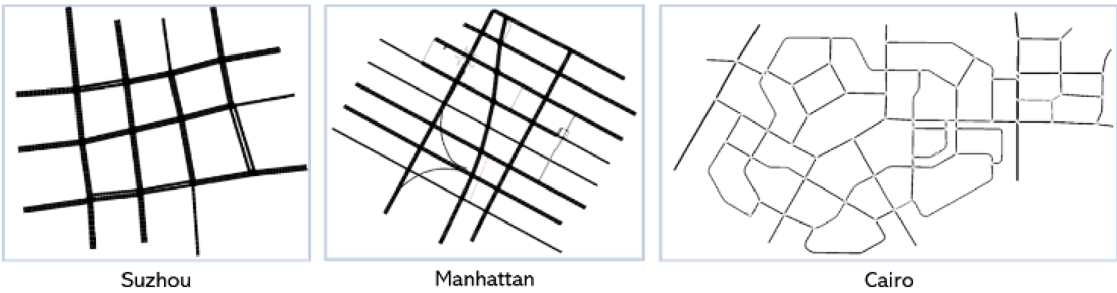

Fig. 7: Experiment Networks Structures

3.2. Software Tools and Environment

In our experiment, we used the Python programming language and various Python libraries, including torch and pygad.torchga. Google Colab, was used for machine learning training. The central control agent interacted with the traffic environment using the TraCI package of SUMO.

The proposed model and other competing models are trained for 500 epochs and tested for 20 epochs. Each epoch comprising 1000 simulation steps executed at a rate of 10 steps per second (the total time is approximately 10000 seconds). For all RL-based approaches, the buffer size of experience replay is 2000, the batch size is between 50 and150, and the target network weights are updated every 100 steps. The learning rate is 0.0001, the discount factor γ is set to 0.9. The exploration rate of the agent starts from 0.5 and gradually decreases at each learning step until it reaches 0.01. In the GCN models, there are 2 hidden layers with 128 neurons in Layer 1 and 64 neurons in Layer 2. The LSTM-GCN models include 3 hidden layers: Layer 0 with 256 neurons, Layer 1 with 128 neurons, and Layer 2 with 64 neurons. A dropout of 0.3 is applied after each layer, and ReLU is employed as the activation function. For the GA models, the total number of generations is set to 100, with 5 parents. GA is employed when the wait cost exceeds 50 thousand seconds. The agent acts action every a second, and every 20 seconds the wait time is calculated.

During the training/testing process, three distinct types of networks were used: Suzhou, Manhattan, and Cairo. For each network, vehicles with different priorities were generated with varying departure times and periods, Table 1 shows the vehicle details during the training/testing process.

Table 1: Priority and Probability of Vehicle Types in Simulation

Vehicle Type	Priority	Generating a Vehicle Probability at Each Time Step			
		Training	Test-1	Test-2	Test-3
Normal	1	0.8	0.4	0.7	1.0
Emergency	10	30.0	20.0	25.0	30.0
VIP	7	-	30.0	35.0	40.0

3.3. The Evaluation Methodology

The proposed traffic optimization model uses the average wait time cost of each vehicle type at each traffic light as the evaluation metrics to measure various algorithms. When a suitable action is executed, the average waiting time cost at time step t will decrease, leading to a reduced wait time at intersection. Conversely, the model will receive a higher average waiting time cost if an unsuitable action is executed. Reducing wait-time cost means effective model learning.

3.4. Performance Evaluation

In our experiment, we conducted the performance evaluations of RL with various deep network models. We evaluated the impact of the GA and the vehicle type on RL model. Then we compared the performance of the LSTM-GCN-GA with priority model to the LSTM-GCN with priority, the LSTM-GCN-GA without priority, the GCN-GA with priority, and Pre-Time models.

3.4.1. Genetic Algorithm (GA)

The integration of GA with RL models achieved significant reductions in the average waiting time. For example, in the Manhattan network as shown in **Fig. 8**, the GA models reduced the average wait time of vehicles by 65.29% for the GCN model, and by 23.72% for the LSTM-GCN model, as shown in **Fig. 8A**. Additionally, as shown in **Fig. 8B** the GA models reduced the average emergency wait time by 13.36% for the GCN model, and by 80.21% for the LSTM-GCN model.

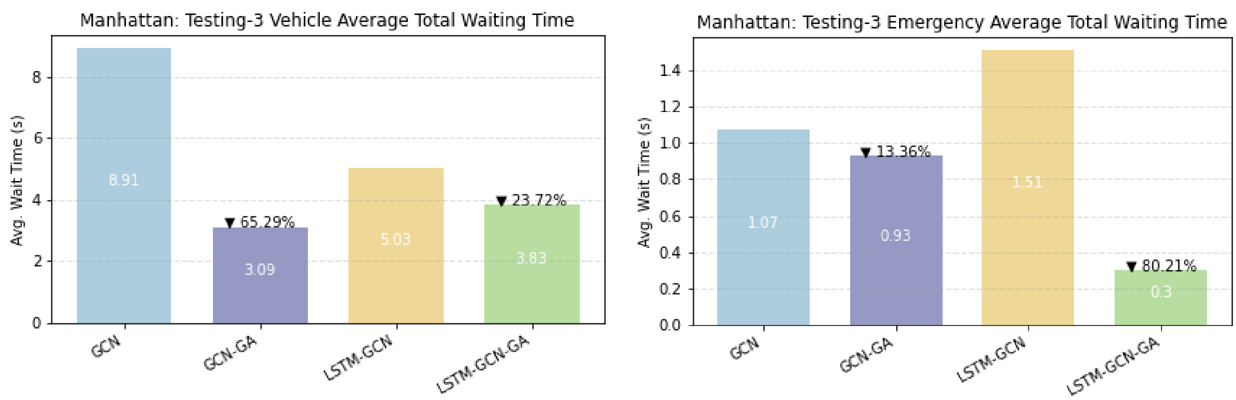

Fig. 8: RL with GA Models Average Wait Time in Manhattan Network: (A) All Vehicles; (B) Emergency Vehicles

3.4.2. Vehicle Priority

Taking the priority of the vehicle into account achieved significant reductions in the average waiting time. For example, in the Manhattan network as shown in **Fig. 9**, the models with priorities reduced the average wait time of vehicles by 64.85% for the GCN model, and by 21.07% for the LSTM-GCN model as shown in **Fig. 9A**. Additionally, as shown in **Fig. 9B** the models with priorities reduced the average emergency wait time by 22.28% for the GCN model, and by 83.82% for the LSTM-GCN model.

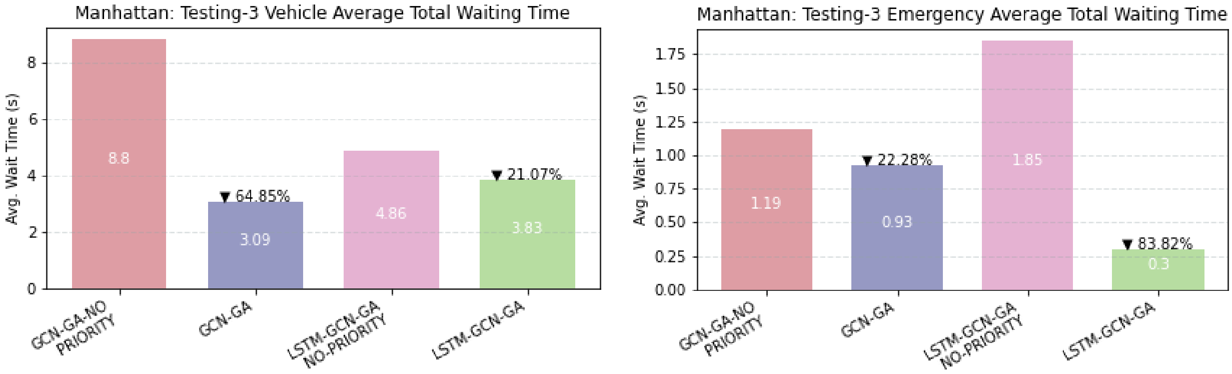

Fig. 9: RL models with/without Priority Average Wait Time in Manhattan Network: (A) All Vehicles; (B) Emergency Vehicles

3.4.2. The Testing

During the testing process which was performed on three distinct types of networks: Suzhou, Manhattan, and Cairo. For each network, the vehicles were generated with three varying departure times and periods: Test-1, Test 2, and Test-3. In Test-1, the test was implemented with heavy traffic; in Test-2, with moderate traffic; and in Test-3, with light traffic. The evaluation compares the performance of the LSTM-GCN-GA with priority model to the LSTM-GCN with priority, the LSTM-GCN-GA without priority, the GCN-GA with priority, and Pre-Time models.

Table 2 shows the testing results of the LSTM-GCN-GA model enhancement percentages compared to the different models. The results of Test-1 in Suzhou, Manhattan, and Cairo networks are shown in Fig. 10**, Fig. 11**, and **Fig. 12**, respectively. While **Fig. 13 , Fig. 14**, and **Fig. 15** are the results of Test-2 in Suzhou, Manhattan, and Cairo networks, respectively. And

Fig. 16 ,

Fig. 17**,** and **Fig. 18** are shown the results of Test-3 in Suzhou, Manhattan, and Cairo networks, respectively.

The experiment results showed that the LSTM-GCN-GA model significantly reduced waiting times compared to the Pre-Time model in various scenarios in the mid-sized network such as Manhattan. The model also showed pretty effectiveness in both smaller and larger networks, like Suzhou and Cairo, respectively. When comparing the LSTM-GCN-GA model to the LSTM-GCN and LSTM-GCN-GA-NO-PRIORITY models, the LSTM-GCN-GA model achieved significant reductions in average waiting times in various scenarios in the mid-sized network, such as Manhattan, and limited effectiveness in smaller and larger networks, such as Suzhou and Cairo, respectively. On the other hand, the GCN-GA model achieved pretty reductions in average waiting times in the mid-sized network as Manhattan under light traffic conditions and slight reductions in larger networks like Cairo under moderate traffic conditions. However, in the other scenarios, the LSTM-GCN-GA model proved more effective with slight reductions compared to the GCN-GA model.

Table 2: The Testing Results of The Enhancement of the LSTM-GCN-GA Model Compared to Different Models in Suzhou, Manhattan, and Cairo Networks. (-Reduce, + Increase) %

Model	Suzhou			Manhattan			Cairo		
	Test-1	Test-2	Test-3	Test-1	Test-2	Test-3	Test-1	Test-2	Test-3
Pre-Time	- 28.84	- 29.71	- 30.28	- 46.14	- 80.86	- 84.81	- 30.28	- 33.24	- 37.0
GCN-GA	- 3.52	- 1.65	- 4.34	- 13.72	- 17.21	+23.95	- 2.35	+ 1.41	- 0.93
LSTM-GCN	- 2.33	- 2.25	- 3.56	- 18.17	- 26.39	- 23.72	- 3.58	- 2.2	- 5.35
LSTM-GCN-GA-NO-PRIORITY	- 3.45	- 1.96	- 8.16	- 33.1	- 27.32	- 21.07	- 2.4	- 1.76	- 7.7

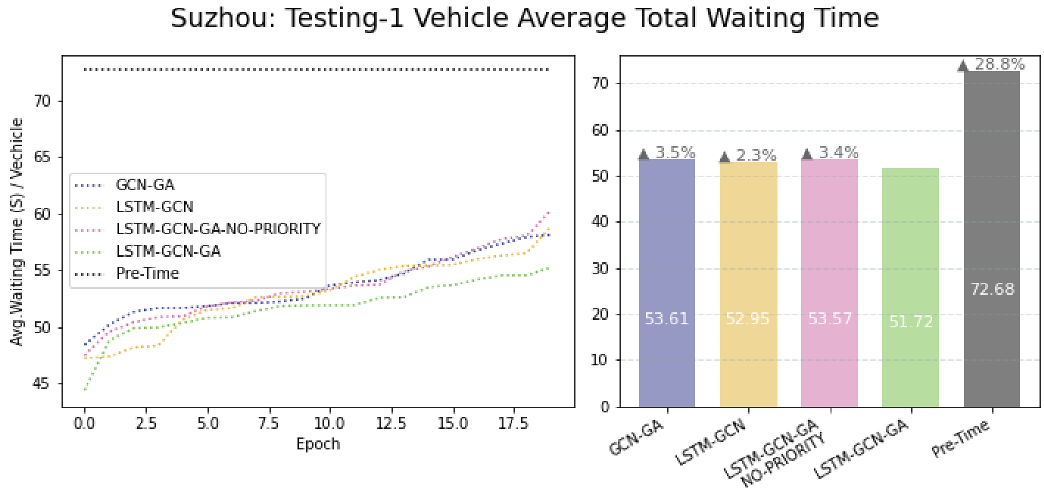

Fig. 10: Test-1 Average Waiting Time in Suzhou Network

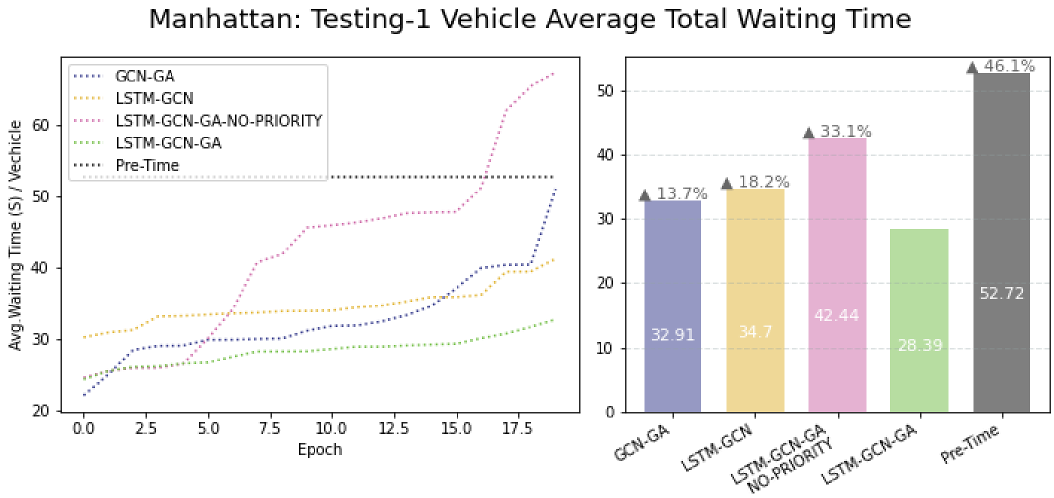

Fig. 11: Test-1 Average Waiting Time in Manhattan Network

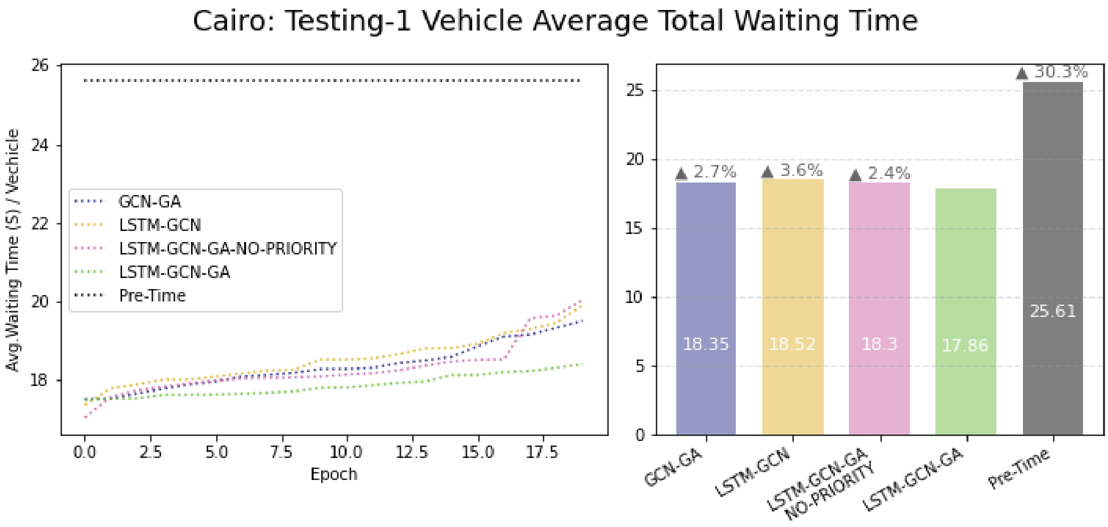

Fig. 12: Test-1 Average Waiting Time in Cairo Network

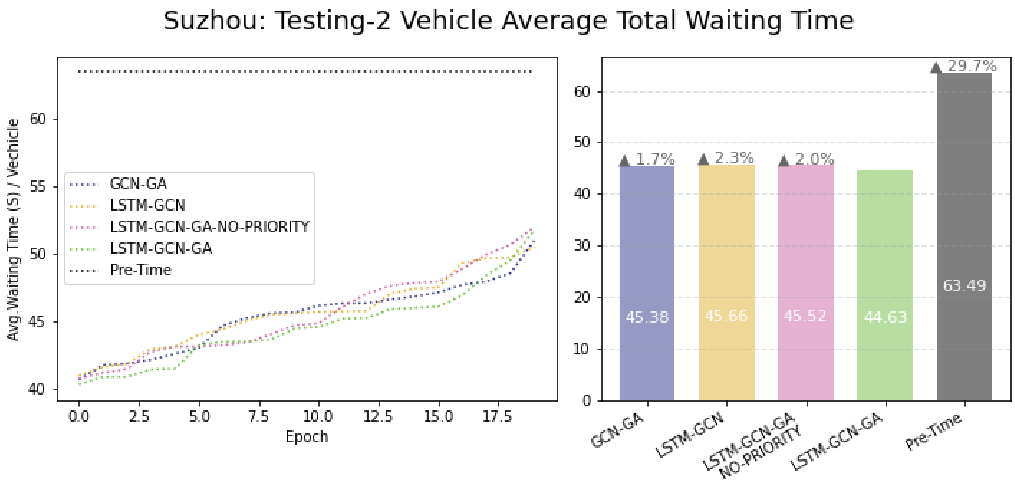

Fig. 13: Test-2 Average Waiting Time in Suzhou Network

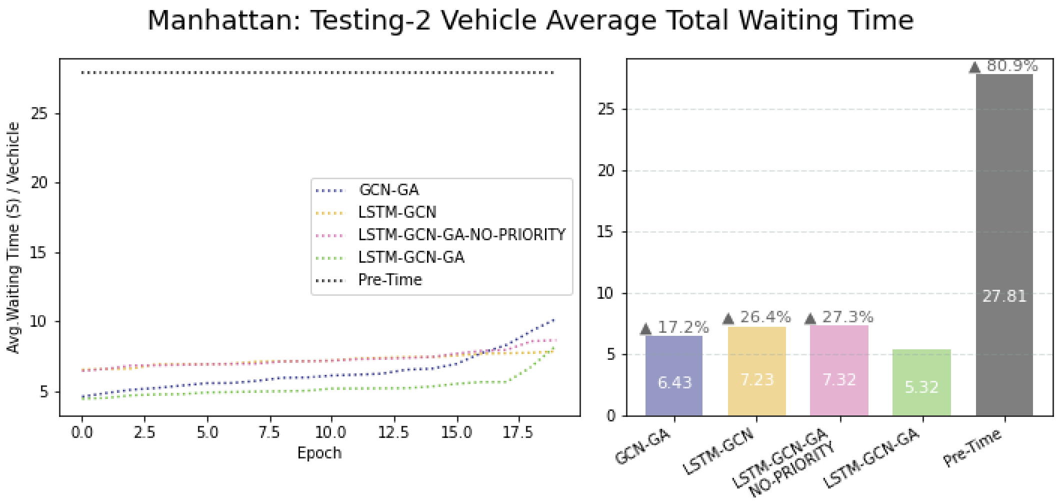

Fig. 14: Test-2 Average Waiting Time in Manhattan Network

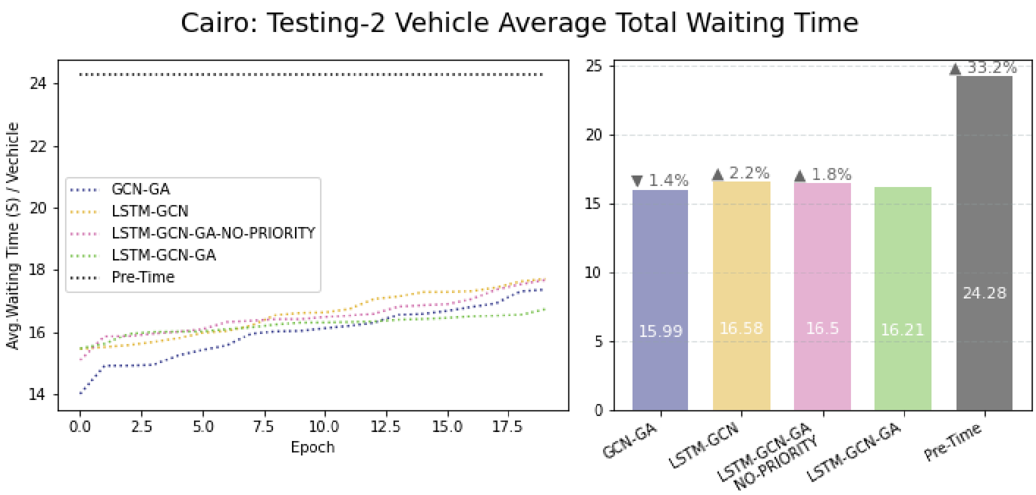

Fig. 15: Test-2 Average Waiting Time in Cairo Network

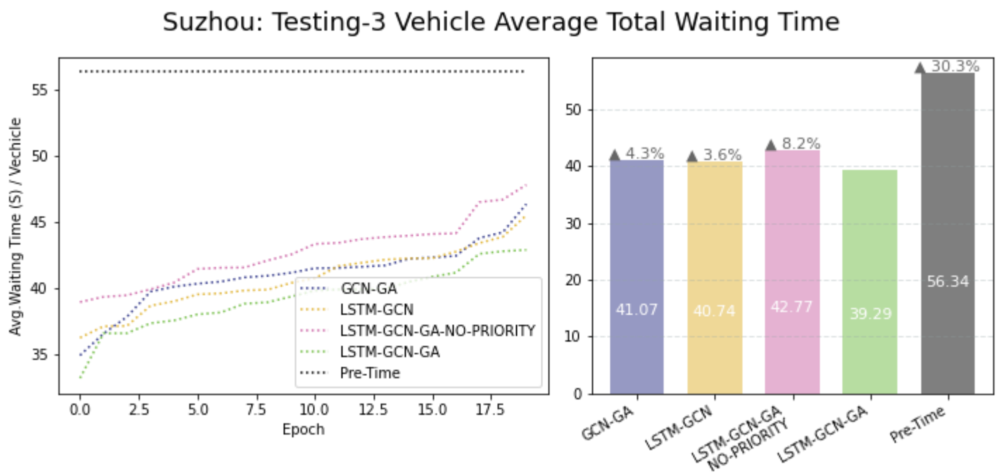

Fig. 16: Test-3 Average Waiting Time in Suzhou Network

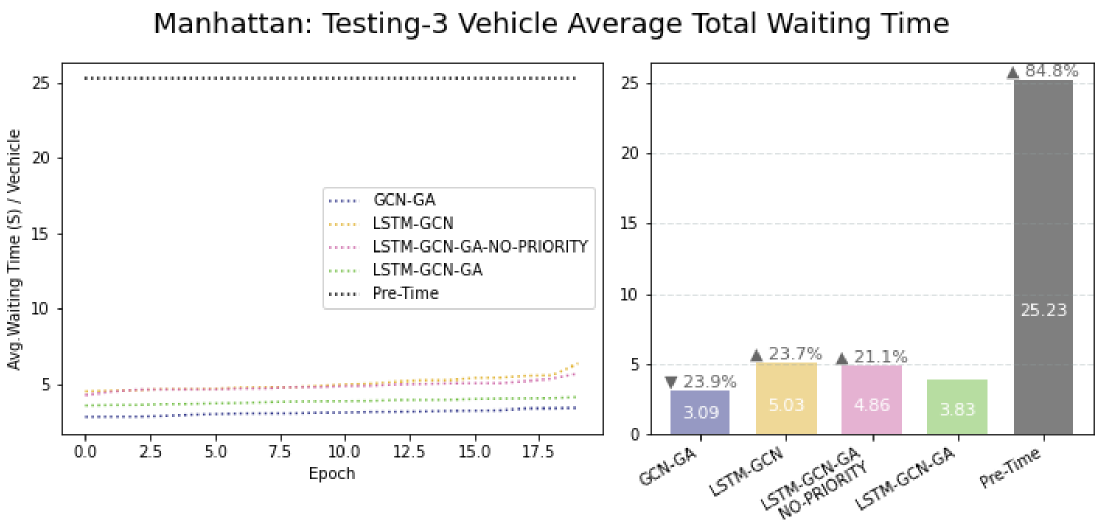

Fig. 17: Test-3 Average Waiting Time in Manhattan Network

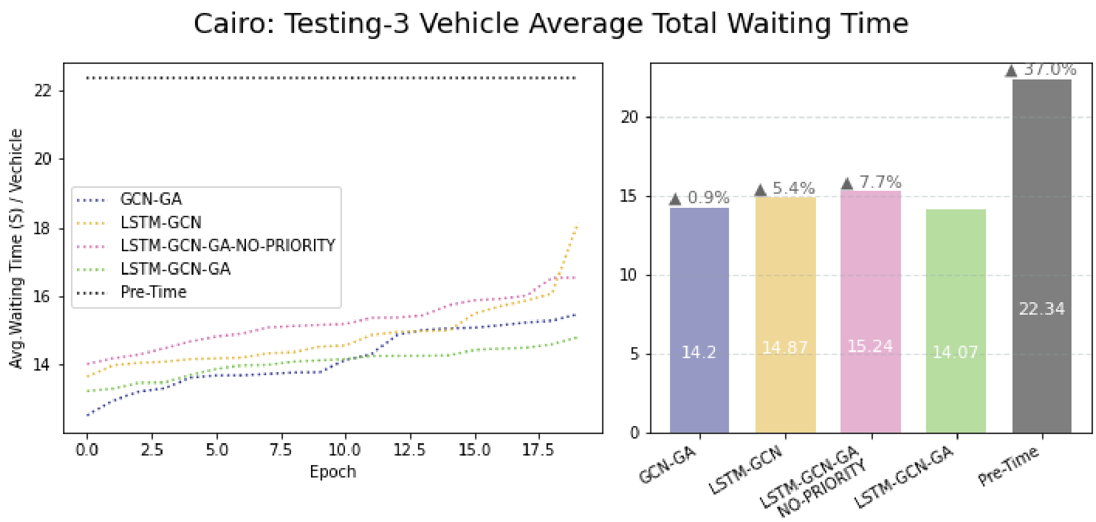

Fig. 18: Test-3 Average Waiting Time in Cairo Network

CONCLUSION AND FUTURE WORK

This paper aimed to develop an intelligent traffic control system to optimize traffic lights, and minimize waiting times, while considering the vehicle type priority, and maximizing their passage through the network, and avoid collisions. The model combines RL, GCN, LSTM, and GA.

The experiment results showed the effectiveness of the LSTM-GCN-GA with priority model significantly reducing waiting times in various scenarios and enhancing overall transportation efficiency.

In the future, we will focus on deploying the LSTM-GCN-GA algorithm in real-time scenarios, considering factors such as weather and accidents, applying it in large and complex intersections, and integrating the autonomous vehicles into the network. Additionally, the model may be expanded to detect accidents and suggest the best pathways.

REFERENCES

[1] Karmakar, G., Chowdhury, A., Kamruzzaman, J., & Gondal, I. (2020). A smart priority-based traffic control system for emergency vehicles. IEEE Sensors Journal, 21(14), 15849-15858.

[2] Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., ... & Li, H. (2019). T-gcn: A temporal graph convolutional network for traffic prediction. IEEE transactions on intelligent transportation systems, 21(9), 3848-3858.

[3] Zhang, D., & Kabuka, M. R. (2018). Combining weather condition data to predict traffic flow: a GRU-based deep learning approach. IET Intelligent Transport Systems, 12(7), 578-585.

[4] Wang, H., Zhu, M., Hong, W., Wang, C., Tao, G., & Wang, Y. (2020). Optimizing signal timing control for large urban traffic networks using an adaptive linear quadratic regulator control strategy. IEEE Transactions on Intelligent Transportation Systems, 23(1), 333-343.

[5] Qadri, S. S. S. M., Gökçe, M. A., & Öner, E. (2020). State-of-art review of traffic signal control methods: challenges and opportunities. European transport research review, 12, 1-23.

[6] Rafter, C. B., Anvari, B., Box, S., & Cherrett, T. (2020). Augmenting traffic signal control systems for urban road networks with connected vehicles. IEEE Transactions on Intelligent Transportation Systems, 21(4), 1728-1740.

[7] Aljaafreh, A., & Al Oudat, N. (2014, March). Optimized timing parameters for real-time adaptive traffic signal controller. In 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation (pp. 244-247). IEEE.

[8] Hung, P. D., & Giang, D. T. (2021). Traffic light control at isolated intersections in case of heterogeneous traffic. Soft Computing for Biomedical Applications and Related Topics, 269-280.

[9] Vuong, X. C., Mou, R. F., Vu, T. T., & Van Nguyen, H. (2021). An adaptive method for an isolated intersection under mixed traffic conditions in Hanoi based on ANFIS using VISSIM-MATLAB. IEEE Access, 9, 166328-166338.

[10] Dong, C., Yang, K., Guo, J., Chen, X., Dong, H., & Bai, Y. (2019, July). Analysis and control of intelligent traffic signal system based on adaptive fuzzy neural network. In 2019 5th international conference on transportation information and safety (ICTIS) (pp. 1352-1357). IEEE.

[11] Elsayed, A., Mohamed, K., & Harb, H. (2023). Enhanced Traffic Congestion Management with Fog Computing: A Simulation-based Investigation using iFog-Simulator. arXiv preprint arXiv:2311.01181.

[12] Alharbi, A., Halikias, G., Sen, A. A. A., & Yamin, M. (2021). A framework for dynamic smart traffic light management system. International Journal of Information Technology, 13, 1769-1776.

[13] Hossan, S., & Nower, N. (2020). Fog-based dynamic traffic light control system for improving public transport. Public Transport, 12, 431-454.

[14] Mondal, M. A., & Rehena, Z. (2019, May). An IoT-based congestion control framework for intelligent traffic management system. In International Conference on Artificial Intelligence and Data Engineering (pp. 1287-1297). Singapore: Springer Nature Singapore.

[15] Kumar, N., Rahman, S. S., & Dhakad, N. (2020). Fuzzy inference enabled deep reinforcement learning-based traffic light control for intelligent transportation system. IEEE Transactions on Intelligent Transportation Systems, 22(8), 4919-4928.

[16] Chu, T., Wang, J., Codecà, L., & Li, Z. (2019). Multi-agent deep reinforcement learning for large-scale traffic signal control. IEEE Transactions on Intelligent Transportation Systems, 21(3), 1086-1095.

[17] Ge, H., Song, Y., Wu, C., Ren, J., & Tan, G. (2019). Cooperative deep Q-learning with Q-value transfer for multi-intersection signal control. IEEE Access, 7, 40797-40809.

[18] Natafgi, M. B., Osman, M., Haidar, A. S., & Hamandi, L. (2018, November). Smart traffic light system using machine learning. In 2018 IEEE International Multidisciplinary Conference on Engineering Technology (IMCET) (pp. 1-6). IEEE.

[19] Wang, H., Chen, H., Wu, Q., Ma, C., & Li, Y. (2021). Multi-intersection traffic optimisation: A benchmark dataset and a strong baseline. IEEE Open Journal of Intelligent Transportation Systems, 3, 126-136.

[20] Hu, X., Zhao, C., & Wang, G. (2020). A traffic light dynamic control algorithm with deep reinforcement learning based on GNN prediction. arXiv preprint arXiv:2009.14627.

[21] Nishi, T., Otaki, K., Hayakawa, K., & Yoshimura, T. (2018, November). Traffic signal control based on reinforcement learning with graph convolutional neural nets. In 2018 21st International conference on intelligent transportation systems (ITSC) (pp. 877-883). IEEE.

[22] Zeng, J., Hu, J., & Zhang, Y. (2018, June). Adaptive traffic signal control with deep recurrent Q-learning. In 2018 IEEE intelligent vehicles symposium (IV) (pp. 1215-1220). IEEE.

[23] Choe, C. J., Baek, S., Woon, B., & Kong, S. H. (2018, November). Deep q learning with LSTM for traffic light control. In 2018 24th Asia-Pacific Conference on Communications (APCC) (pp. 331-336). IEEE.

[24] Lv, M., Hong, Z., Chen, L., Chen, T., Zhu, T., & Ji, S. (2020). Temporal multi-graph convolutional network for traffic flow prediction. IEEE Transactions on Intelligent Transportation Systems, 22(6), 3337-3348.

[25] Cui, Z., Henrickson, K., Ke, R., & Wang, Y. (2019). Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting. IEEE Transactions on Intelligent Transportation Systems, 21(11), 4883-4894.

[26] Diveev, A. I., & Sofronova, E. A. (2019, April). A mathematical model and control problems of traffic flows in urban road networks. In 2019 6th International Conference on Control, Decision and Information Technologies (CoDIT) (pp. 837-842). IEEE.