**SMART TRAFFIC SYSTEM**

**A Project Report submitted in partial fulfillment of the requirements for**

**the award of the degree of**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**Submitted by**

**Vegu Mahesh Babu 121710318057**

**V.S.B. Rao 121710318055**

**Patibandla Dharmesh 121710318039**

**Gondrala Dinesh 121710318013**

**Under the esteemed guidance of**

**Mrs Manasa Vavilapalli**

**Assistant Professor,GITAM (CSE)**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**GITAM**

**(Deemed to be University)**

**VISAKHAPATNAM**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**GITAM INSTITUTE OF TECHNOLOGY**

**GITAM**

**(Deemed to be University)**

****

**DECLARATION**

We, hereby declare that the Project review entitled “**SMART TRAFFIC SYSTEM,**” is an original work done in the Department of Computer Science and Engineering, GITAM Institute of Technology, GITAM (Deemed to be University) submitted in partial fulfilment of the requirements for the award of the degree of B.Tech. in Computer Science and Engineering. The work has not been submitted to any other college or University for the award of any degree or diploma.

**Date**: 12 December 2020.

**Name: Registration No:**

Vegu Mahesh Babu 121710318057

V.S.B. Rao 121710318055

Patibandla Dharmesh 121710318039

Gondrala Dinesh 121710318013

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**GITAM INSTITUTE OF TECHNOLOGY**

**GITAM**

**(Deemed to be University)**

****

**BONAFIDE CERTIFICATE**

This is to certify that the project report entitled “**SMART TRAFFIC SYSTEM**” is a bonafide record of work carried out by **Vegu Mahesh Babu 121710318057 ,V.S.B. Rao 121710318055, Patibandla Dharmesh 121710318039, Gondrala Dinesh 121710318013**, submitted in partial fulfilment of requirement for the award of degree of Bachelors of Technology in Computer Science and Engineering.

**PROJECT GUIDE** **HEAD OF THE DEPARTMENT**

**Mrs. Manasa Vavilapalli**  **Dr.R.Sireesha**

**(Assistant Professor) (Professor)**

**CSE, GIT CSE, GIT**

**GITAM GITAM**

**ACKNOWLEDGEMENT**

We would like to thank our project guide **Mrs. Manasa Vavilapalli**, Assistant professor, Department of CSE for her stimulating guidance and profuse assistance. We shall always cherish our association with him for his guidance, encouragement and valuable suggestions throughout the progress of this work. We consider it a great privilege to work under his guidance and constant support.

We also express our thanks to the project reviewers **Venkata Ramana M, Assistant Professor, and Appa Rao G, Assistant Professor, Department of CSE, GITAM (Deemed to be University)** for their valuable suggestions and guidance for doing our project.

We consider it a privilege to express our deepest gratitude to **Dr. R. Sireesha**, Head of the Department, Computer Science Engineering for his valuable suggestions and constant motivation that greatly helped us to successfully complete this project.

Our sincere thanks to **Dr. C. Dharma Raj**, Principal, GITAM Institute of Technology, GITAM (Deemed to be University) for inspiring us to learn new technologies and tools.

Finally, we deem it a great pleasure to thank one and all that helped us directly and indirectly throughout this project.

Vegu Mahesh Babu 121710318057

V.S.B. Rao 121710318055

Patibandla Dharmesh 121710318039

Gondrala Dinesh 121710318013

**1.ABSTRACT**

Machine learning is progressing at a rapid rate. Almost every industry employs machine learning.. It's also an artificial intelligence programme in which available data is processed or supported by algorithms to process statistical data. Although Machine Learning is based on the principle of automation, it still needs human supervision. In recent years, the number of vehicles on the road has increased significantly. Congestion is a growing problem that affects everyone on a regular basis.

The use of traffic cops to manually manage traffic has not proven to be successful. The problem has not been resolved by the signal's predetermined fixed period in all situations (low and high traffic density).

Smart Traffic System is the topic of the proposed project. Convolutional neural networks (CNN) were used in this case, and Simulation of urban mobility was used for simulation (SUMO).

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
|  | Declaration | 2 |
|  | Certificate | 3 |
|  | Acknowledgements | 4 |
| 1. | Abstract | 5 |
| 2. | Introduction & Literature Review | 7-9 |
| 3. | Problem Identification And Objectives  3.1 Existing Problem  3.2 proposed Solution  3.3 Technologies/Tools Used | 10-11 |
| 4. | System Architecture And Block Diagram  4.1 Convolution Neural Networks  4.2 Layers Of CNN  4.3 Deep Q Network | 12-16 |
| 5 | Overview of technologies | 17 |
| 6. | Dataset Specifications | 18 |
| 7. | Software Specifications | 19-20 |
| 8. | Hardware Specifications | 21 |
| 9. | Programming Language | 22 |
| 10. | Implementations | 23-40 |
| 11. | Results & Analysis | 41-45 |
| 12. | Conclusion | 46 |
| 13 | Reference | 47 |

**1.INTRODUCTION**

To keep traffic flowing more smoothly, a smart traffic management system based on Machine Learning, communication, and automated algorithms is being developed. The aim is to monitor the length of a green or red light for a particular traffic light at an intersection as efficiently as possible. The traffic lights do not always flash the same shade of green or red, but should vary depending on the number of cars on the road. When there is heavy traffic in one direction, the green lights should remain on longer; when there is fewer traffic, the red lights should stay on for a longer period of time. This approach should remove inefficiencies at intersections and reduce commuting costs and emissions.

In 2014, urban dwellers made up 54 percent of the global population. The forecast was for annual growth of approximately 2% until 2020, putting more strain on cities' transportation systems. Furthermore, the high cost of living in business districts forces urban workers to live far from their places of employment or schooling. As a result, they must drive back and forth between their home and their place of employment. A fixed number of roads and highway networks must be able to handle more vehicles travelling. When faced with increased traffic, it's common practise to simply widen lanes or raise road levels.

However Cities, on the other hand, should focus on making their streets run more efficiently rather than simply expanding them or adding more highways. This leads to the proposed system, which will track the number of vehicles using a microcontroller and sensors, allowing for time-based monitoring of the system.

**2.LITERATURE REVIEW**

**2.1 Introduction:**

For optimized traffic here we used a few algorithms in Deep Q-Networks (DQN) in Convolutional Neural Networks(CNN) where these all are subsets of Machine Learning. We will address a few research papers in this section of the paper that presented various studies on the Traffic System and the methods that the researchers used to achieve their target.

**2.2 Related Work:**

With the help of, a variety of papers have been written the aim of overcoming the conventional traffic light system's drawbacks. The various methods used to determine traffic density can be classified based on vehicle detectors such as piezoelectric sensors and Inductive Loops[2], Ultrasonic Sensors[3], Infrared Sensors[4] and sound Sensors[5], Acoustic Sensors[6], Video/ Image processing techniques[7]-[9], RF based detectors[10], Fuzzy Logic Systems[11], and cloud computing systems[12]. This paper employs a variety of methods in order to produce more precise results.

1.Babu, P. R. K. S. M. R. (2016). Using the internet of things and big data, build a real-time smart traffic management system for smart cities.. 2016 International Conference on Emerging Technological Trends (ICETT).

2.Chandana K K, Dr. S. Meenakshi Sundaram, C. D. M. N. S. N. K. (2013). Using the internet of things (iot), build a smart traffic management system for congestion control and alerts.. Saudi Journal of Engineering and Technology,

3.Dave, P. N. D. M. . P. S. P. (2018). Smart traffic management system using iot. Computer Engineering and Applications: An International Journal

3.Sabeen Javaid, Ali Sufian, S. P. M. T. (2018). Smart traffic management system using internet of things. The 20th International Advanced Communication International Conference is a gathering of experts in the field of advanced communication technology (ICACT)

4.Viswanathan, V. and Santhanam, V. (2013). Traffic signal control using wireless sensor networks. ICAEE'2013 is the 2nd International Conference on Advances in Electrical and Electronics Engineering.

5.Yucheng Huang, Linbing Wang, Y. H. W. Z. Y. Z. (2018). A prototype iot based wireless sensor network for traffic information monitoring. volume 11. Zantout, S. (2017). Traffic light controller project final report.

6.Genders, Wade, and Saiedeh Razavi. "For traffic signal management, a deep reinforcement learning agent is used." arXiv preprint arXiv:1611.01142 (2017).

7.Wade Genders & Saiedeh Razavi (2019): Asynchronous n-step Q- learning adaptive traffic signal control, Journal of Intelligent Transportation Systems, DOI: 10.1080/15472450.2018.1491003

**3.PROBLEM IDENTIFICATION AND OBJECTIVES**

**3.1 EXISTING PROBLEM:**

On our indian roads, the major problem is traffic due to the heavy population. Especially in metropolitan cities the roads are always busy and It is creating major trouble day by day, in some cities it takes hours to clear the traffic. And this happens when there are more vehicles in the city and a weak traffic management system. Due to this problem sometimes ambulances and fire engines will get stuck in the traffic.

Right now, there is no efficient traffic system in india. Which was very much needed in this highly populated country. And also negligence in buying the vehicles. So, best solution is to implement a good traffic system.

**3.2Proposed Solution:**

Reinforcement is the subset of Machine learning. It is one of the paradigms of machine learning along with supervised and unsupervised learnings.

Basically reinforcement deals with intelligent agents about how they take actions when required.

It depends on a Deep Q-network algorithm with a few convolutional layers in the Convolutional Neural Network(CNN).

Using this algorithm we have created a Smart Traffic System. And we can implement this System into our indian traffic management system.

**3.3 TECHNOLOGIES / TOOLS USED:**

**1.Pycharm:** Pycharm is an open-source cross platform Integrated Development Environment for scientific programming in Python language.

**2.Sumo:** For simulation purposes here we used Simulation of urban mobility(SUMO). It is an open Source software which is used to handle Traffic systems. It is portable,highly microscopic and we can include several packages which not only shows the cars movement but also shows pedestrians and many tools for scenario creation.

**2.TENSORFLOW:** TensorFlow is an open-source library for numerical computation and large-scale Machine Learning

**3.KERAS:** Keras is an open-source library for artificial neural networks that offers a Python interface. Keras serves as a user interface for TensorFlow.

**4.SYSTEM ARCHITECTURE**

We used the Convolution Neural Networks Algorithm in this project to optimise traffic and provide vehicles with less waiting time at traffic intersections. Most widely used algorithm is convolutional neural networks. Our project "SMART TRAFFIC SYSTEM" is concerned with traffic optimization using a few algorithms in CNN. For the implementation of this model, we have also imported various Python libraries such as TensorFlow and Keras.

**4.1 Convolutional Neural Networks**

A Convolutional Neural Network (CNN) is a Deep Learning algorithm that can take an image as input, assign importance (learnable weights and biases) to various aspects/objects in the image, and distinguish one from the other.

**4.2 LAYERS OF CNN:(Convolution Neural Networks)**

CNN contains 4 layers

1.Convolution Layer

2.Max Pooling Layer

3.Flatten Layer

4.Fully Connected Layer

**1.Convolution Layer:**

The first layer to remove features from an input image is the Convolution Layer. It learns image features from tiny squares of input data, preserving the relationship between pixels.

**2.Max Pooling Layer:**

This layer is used to decrease the dimensionality of an input sample. When the images are too large, it usually decreases the number of parameters. As a result, the output after the max pooling layer would have the most extrusive features of the preceding function of the map. In general, this layer takes a 2\*2 matrix and selects the block with the highest value and outputs it.

**3.Flatten Layer:**

The flatten layer receives the pooled layer's output as an input. This layer is primarily used to transform an n-dimensional array or matrix into a 1-dimensional array or matrix.

**4.Fully Connected Layers:**

It is the final layer of CNN and is a very critical one. It uses weights to predict the correct mark based on the inputs from the function analysis. It gives each label's final probabilities.

**4.3 Deep Q-Network:**

Many AI labs have been researching deep reinforcement learning (DRL) in recent years, as it is expected to become a key technology in the future. At Keio University, In the following papers, I will present my survey of recent DRL. They all presume that readers have a clear understanding of reinforcement learning.

Deep Q-Network (DQN) was used, which is DeepMind's first deep reinforcement learning tool. Since deep neural networks can enable RL to directly deal with high-dimensional states like images, thanks to techniques used in DQN, several research institutes have entered this field since the paper was published in Nature in 2015.

To estimate Q values directly from images, DQN uses three convolutional layers and two completely connected layers. Linear models, on the other hand, have only one completely connected layer, with some learning techniques discussed in the following section. Both models use the Q learning method to learn Q values. Since DNN is easily overfitting in online reinforcement learning, naive DQN has weak performance, much worse than linear models, as seen in the table above.

DQN uses four key strategies to solve unstable learning.

**Experience Replay**

**Target Network**

**Clipping Rewards**

**Skipping Frames**

**Experience Replay:**

Reinforcement Learning for Robots Using Neural Networks was published in 1993. suggested Experience Replay. Present episodes are easily overfitted by DNN. It's difficult to create different experiences once DNN has been overfitted. Experience Replay solves this problem by storing experiences such as state changes, rewards, and behaviours, which are required data for Q learning, and creating mini-batches to update neural networks. The following benefits are expected from this technique.

-reduces the association between DNN update experiences

-mini-batches improve learning speed

-avoids disastrous forgetting by reusing previous transitions

**Target Network:**

With DNN, the aim function is frequently modified in TD error calculation. Training is complicated due to the inconsistency of the goal feature. As a result, the Target Network technique fixes target function parameters and replaces them with the most recent network every thousands of steps.

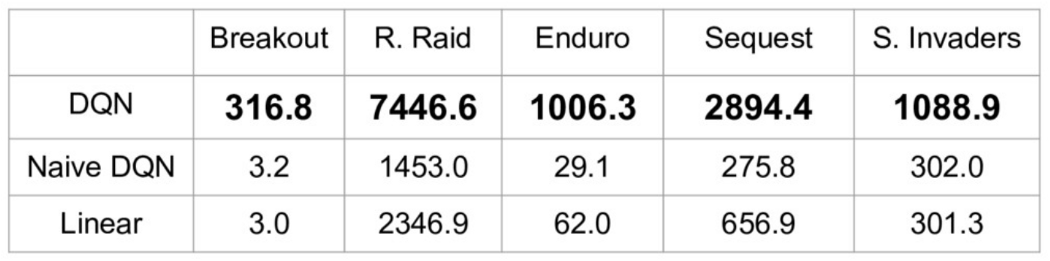
**Clipping Rewards**

Each game has its own scoring system. In Pong, for example, when a player wins a game, he or she receives one point. Otherwise, teams will receive a -1 point penalty. Players in SpaceInvaders, on the other hand, receive 10~30 points for beating invaders. This distinction would make preparation insecure. As a result of the Clipping Rewards technique, all positive rewards are set to +1 and all negative rewards are set to -1.

**Skipping Frames**

ALE has a rendering speed of 60 images per second. People, on the other hand, do not act in a split second. Every frame, AI does not need to measure Q values. DQN calculates Q values every four frames and uses the previous four frames as inputs in the Skipping Frames technique. This lowers the computational cost and increases the number of interactions.

DQN is able to achieve stable training using all of the strategies mentioned below.



**5. OVERVIEW OF THE TECHNOLOGIES**

**Neural Networks:**

A neural network is inspired by the structure of a human brain. It is basically a deep learning model used for unsupervised learning It is a collection of interconnected entities known as nodes. Each node in a neural network is responsible for simple computations.

To solve AI problems, we use a biological network made of neurons. Weights are assigned based upon the connections. There will be hidden layers present in the network. The output range is generally binary or a continuous value based on the kind of activation functions. It ranges between 0 and 1 or in between 1 and –1.

**5.1 Overview of Neural Networks:**

Neurons are generally connected to each other. Connections between neurons are linked which leads to increase in the number of connections. They are typically connected from Axons to Dendrites although it is possible to establish synapses and another connection. Artificial Intelligence aims to mimic some properties of neuro biological network. For the creation of Software agents neural networks are widely used in Speech Recognition. The main focus of neural network is to process the knowledge inside the biological neuron. They don’t distinguish between memory and processing.

**5.2 Application of Neural Networks:**

There are many applications of Neural Networks in various fields.

They are used in Email – spam filtering, game playing, pattern recognition, sequence recognition, diagnosis, financial application, processing, visualization.

**6. DATASET SPECIFICATIONS**

There will be no datasets for reinforcement learning. Reinforcement learning is mainly about Studying environments. Unlike supervised learning in reinforcement learning doesn't have any datasets. The supervised learning model is trained with datasets and performs operations accordingly. But the reinforcement agent decides what action to be taken next using previous experiences.

**7.Software Specifications**

**Software:**

* TensorFlow
* Flask
* Pycharm
* Sumo

**Pycharm:**

It is an open source platform Integrated Development Environment for scientific programming in Python language. Generally Pycharm contains many modules and packages. It has a command line interface. We can even look for the packages in options. It is an ide for professional developers. It boosts code quality and also gives testing assistances

It includes a code analyzer, a graphical debugger, an integrated unit tester, and version control system integration.

**TensorFlow:**

TensorFlow is an open-source library for numerical computation and large-scale Machine Learning.

It is a package used widely for building Neural Networks. It was designed for both research and development purposes.

**Flask:**

Flask is a web application framework written in Python. It is developed by Armin Ronacher.

The flask package is installed from Python Package Index (PPI).

**Sumo:** For simulation purposes here we used Simulation of urban mobility(SUMO). It is an open Source software which is used to handle Traffic systems. It is portable,highly microscopic and we can include several packages which not only shows the cars movement but also shows pedestrians and many tools for scenario creation.

**8.HARDWARE SPECIFICATIONS**

|  |  |  |
| --- | --- | --- |
| **DESCRIPTION** | **MINIMUM** | **REQUIRED** |
| RAM | 2GB | 8GB |
| Cache | 256MB | 512MB |
| Storag**e** | 40GB | 120GB |
| Display | 15” VGA | 17” SVGA(LCD) |

**9. PROGRAMMING LANGUAGE**

As a part of developing the project we need a Computer Programming language which is very efficient and easy to understand and write the code. We used **Python Programming language for developing Smart traffic system.**

Python is a dynamically semantic, interpreted, object-oriented high-level programming language. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it ideal as a scripting or glue language for rapid application development for connecting existing components. Modules and packages are supported by Python, which facilitates software modularity and code reuse. The Python interpreter and its comprehensive standard library are free to download and distribute in source or binary form for all major platforms.

Python is common among programmers because of the improved productivity it offers. The edit-test-debug cycle is extremely quick because there is no compilation phase. A bug or poor input would never trigger a segmentation fault when debugging Python programmes. Instead, when the interpreter finds a mistake, it throws an exception. The interpreter prints a stack trace if the programme fails to detect the exception. Inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code one line at a time, and so on are all possible with a source level debugger.The debugger is written entirely in Python. On the other hand, adding a few print statements to the source code is always the fastest way to debug a programme; the fast edit-test-debug cycle allows this easy approach quite efficiently.

**10. IMPLEMENTATION**

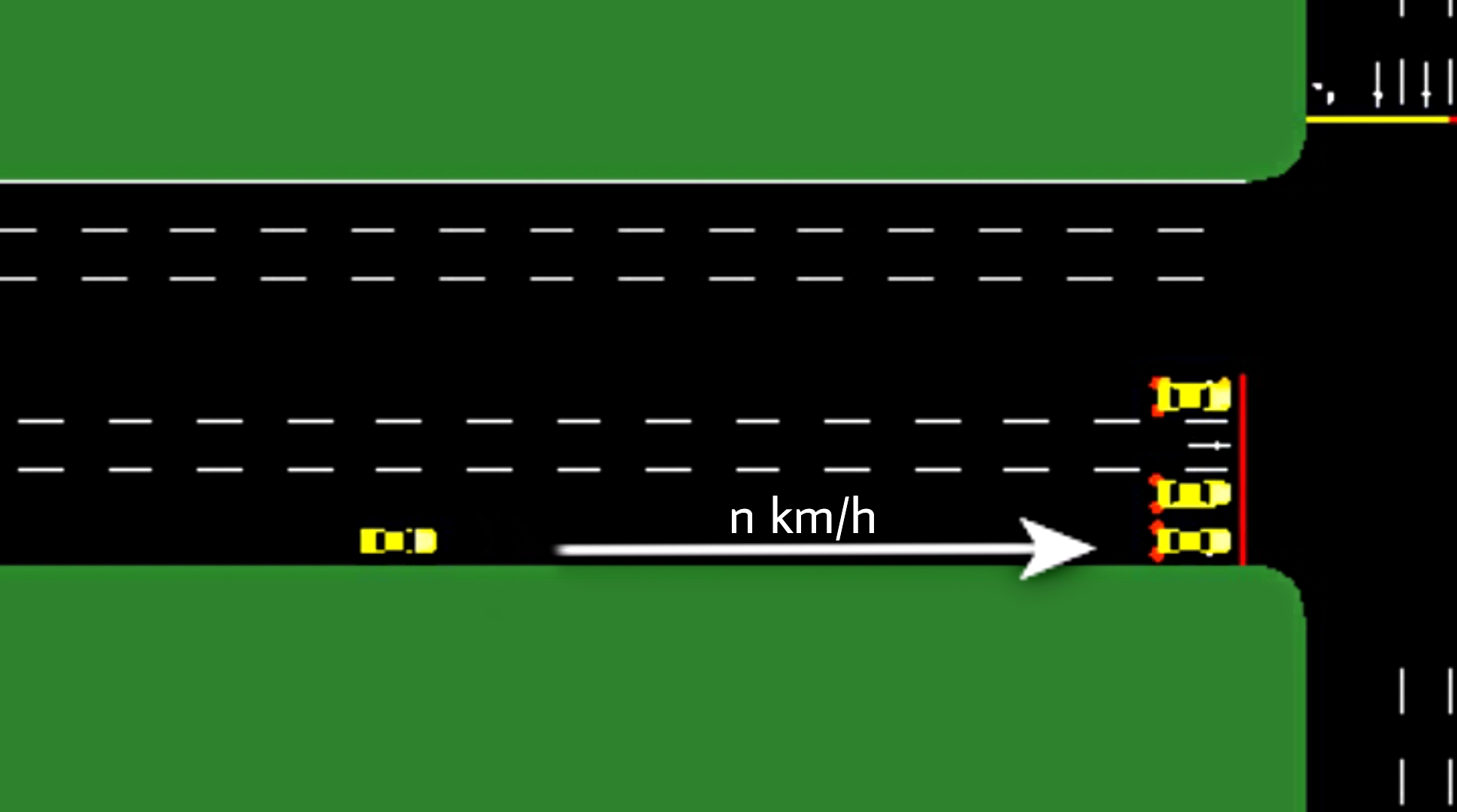
Simulation of Urban MObility (SUMO) is used to model the intersection. TraCI API is offered as one of SUMO's tools for communicating with Python.

The Deep Q-learning algorithm was chosen to perform Reinforcement Learning. To perform acts, an agent must first observe the situation on the path, which entails constructing two matrices.

The first matrices divides the road into 16 segments, each measuring 7 meters, and tests if the car is in the correct segment. The lane is the section here. We took four different roads in total. We used four lanes for each route. We can see the figure below.



It's time to move on to the second matrix after testing the cars in the first. The importance of this matrix cannot be overstated. This matrix measures the speed of the car approaching the Traffic Junction. It determines the amount of time it takes to get to the traffic junction.



**10.1 UML DIAGRAMS:**

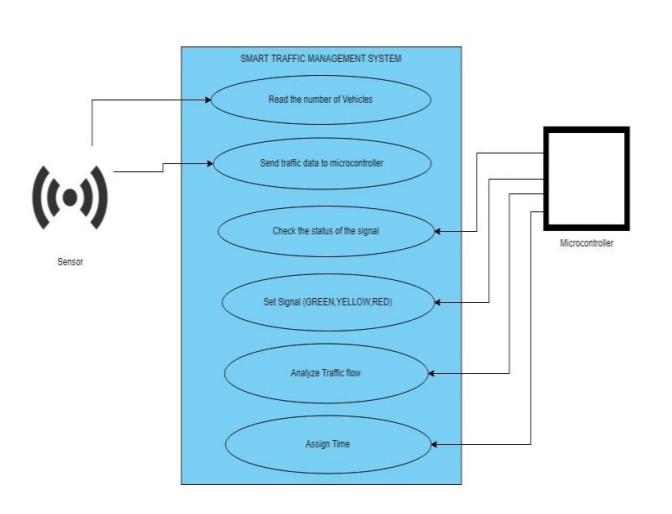
UML stands for Unified Modelling Language. It is a general-purpose modelling language. UMLconsists of 2 main components meta data and notation. UML is a standard language for programming software system object documentation for business and other non-software systems.

UML is the best practice to model structures that are large and complex. To explain the software project architecture UML uses graphical notations.

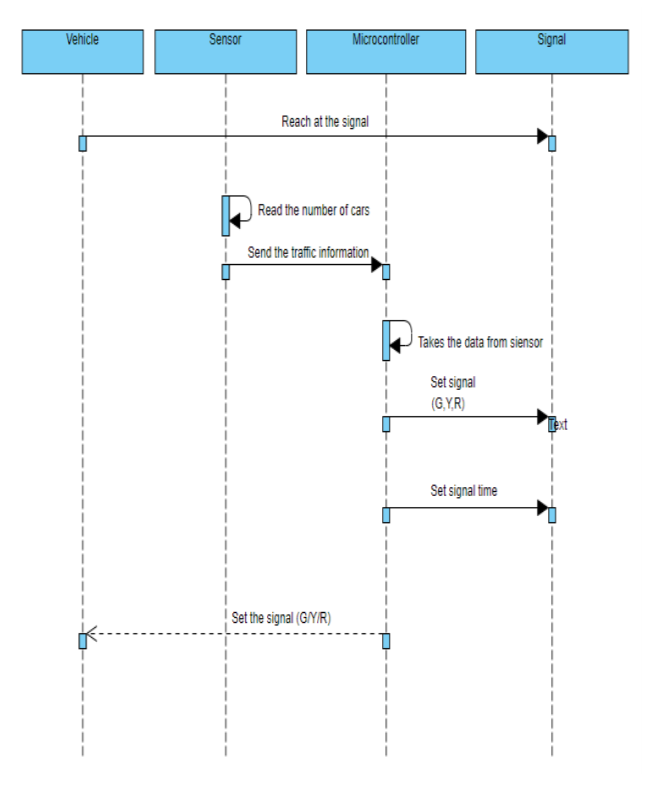
**10.1.1 USE CASE DIAGRAM:**

A UML is defined and generated from the analysis of use case. The relation between actors their goals and their dependencies are provided with the help of system functionality. The demonstration of functionalities is performed for respective actors.

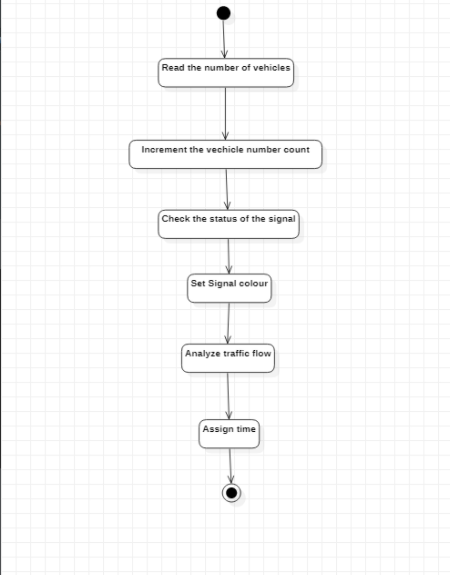
**Use case diagram:**



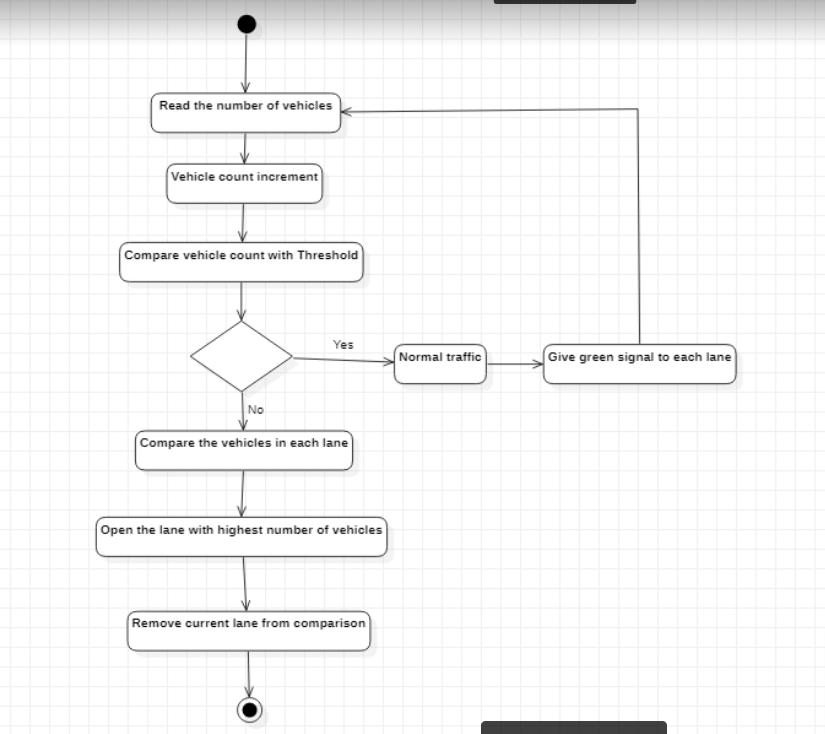
**Sequence diagram**



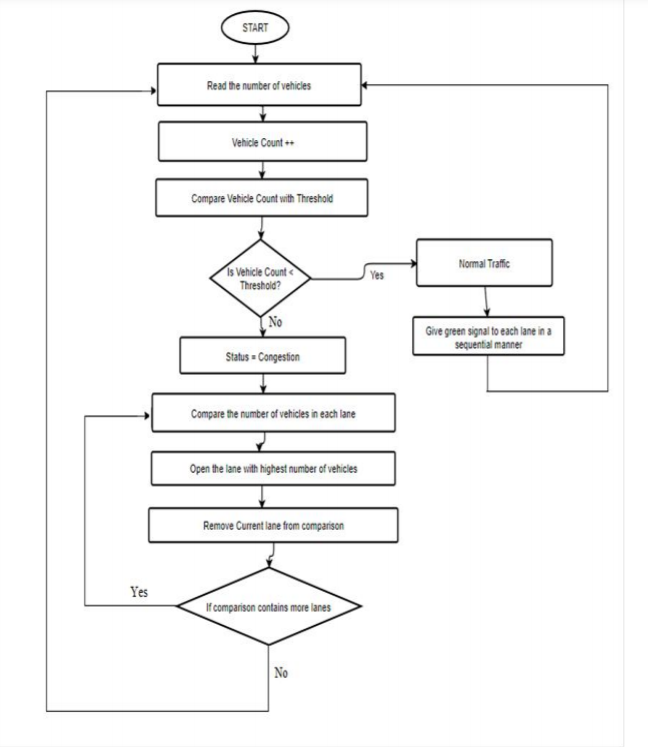
**State-Chart Diagram:**

****

**Activity Diagram:**

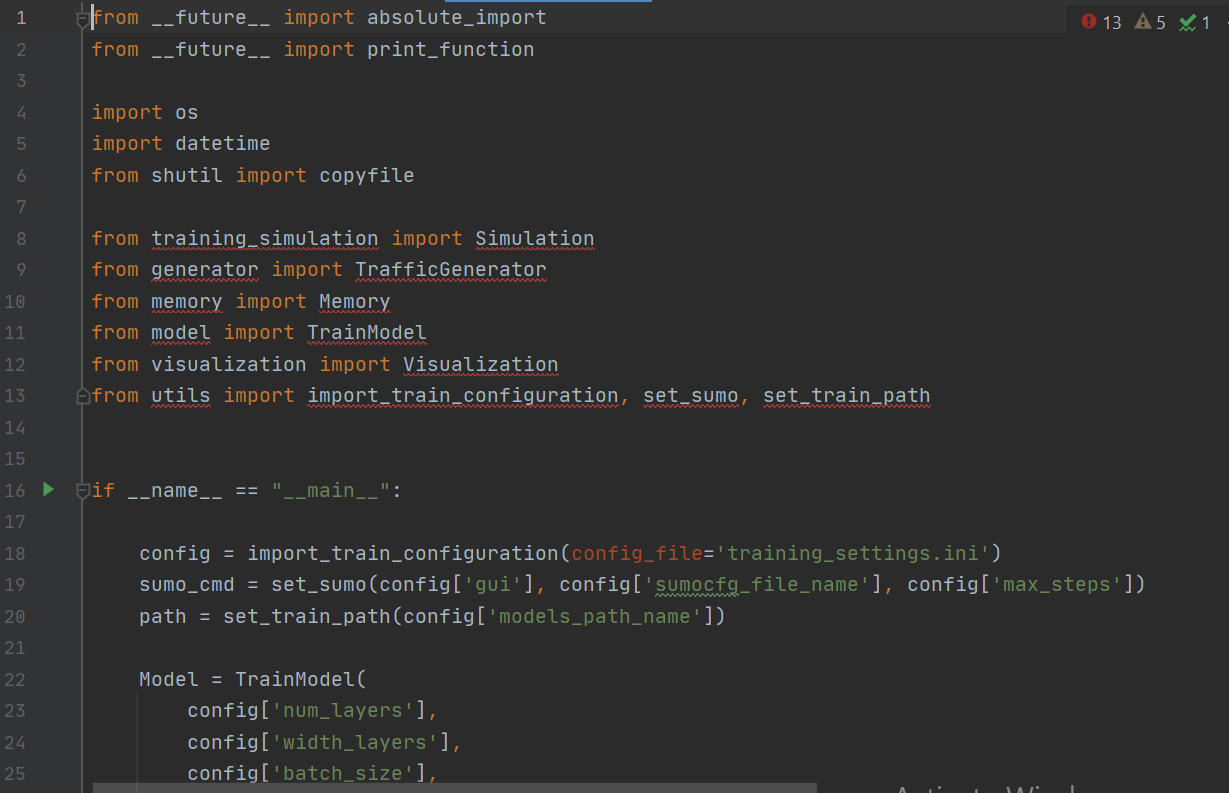
****

**Flowchart**

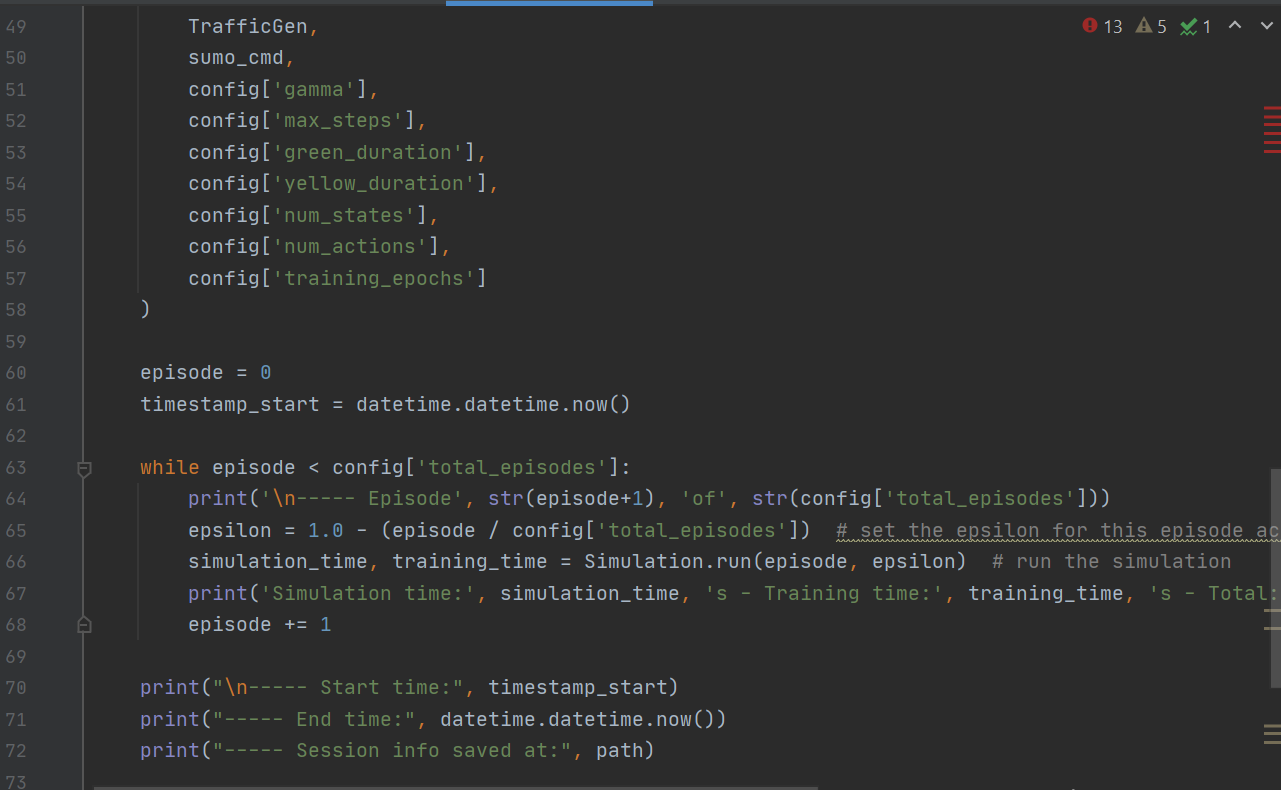
****

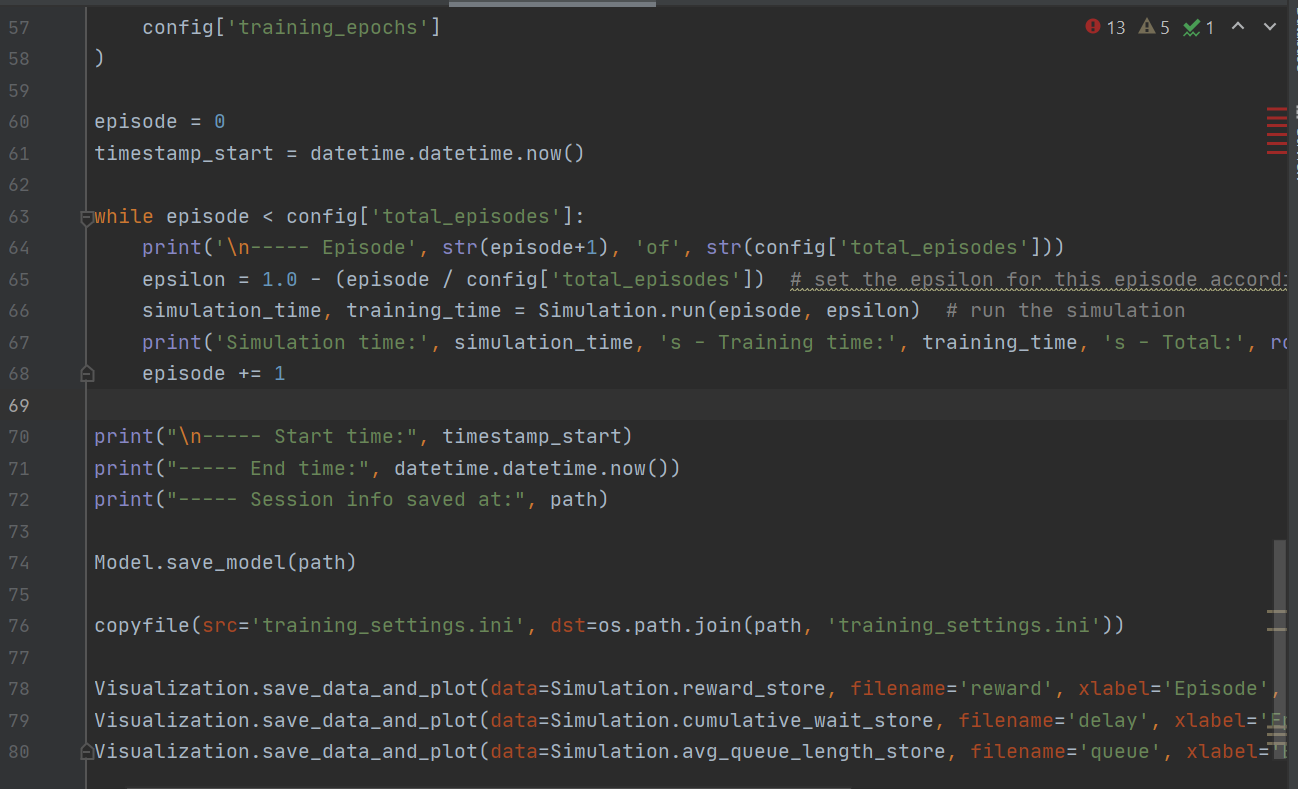
**10.2 Coding:**

**train\_main.py**

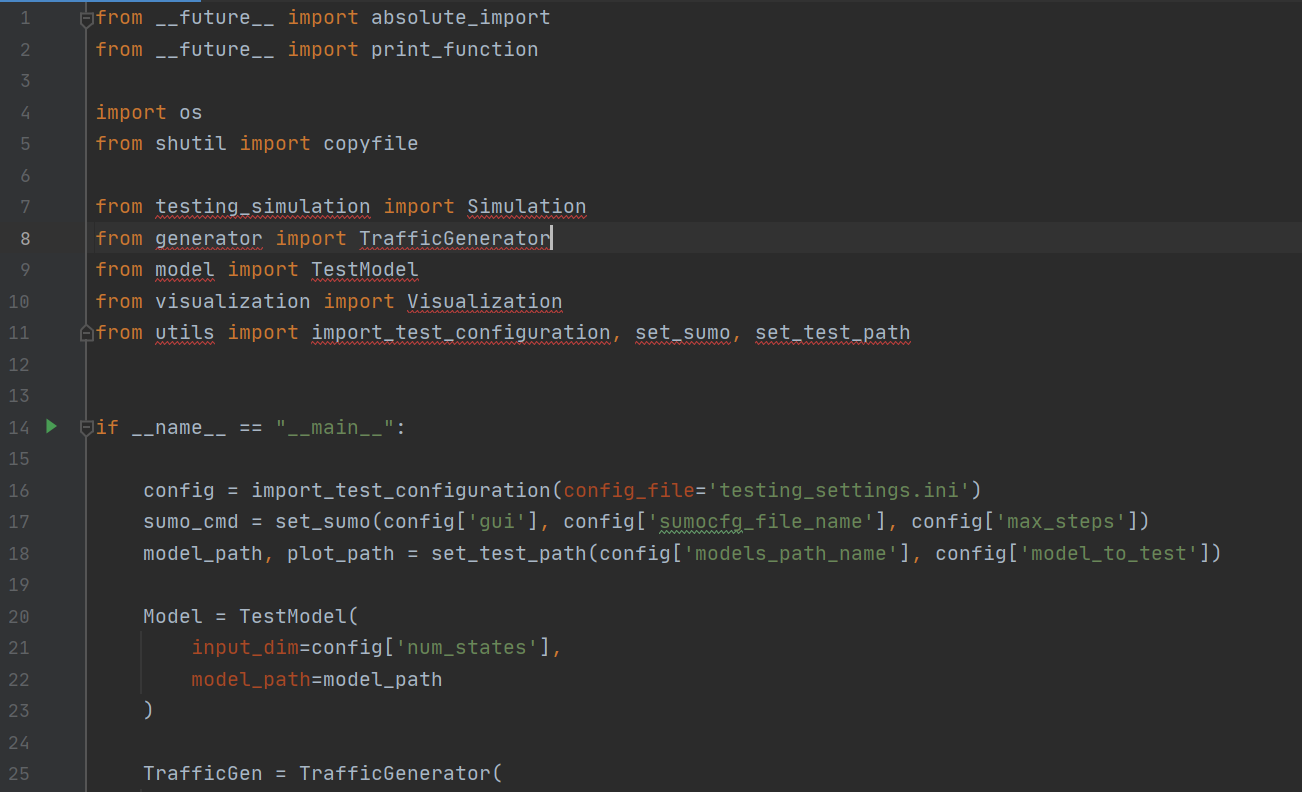
****

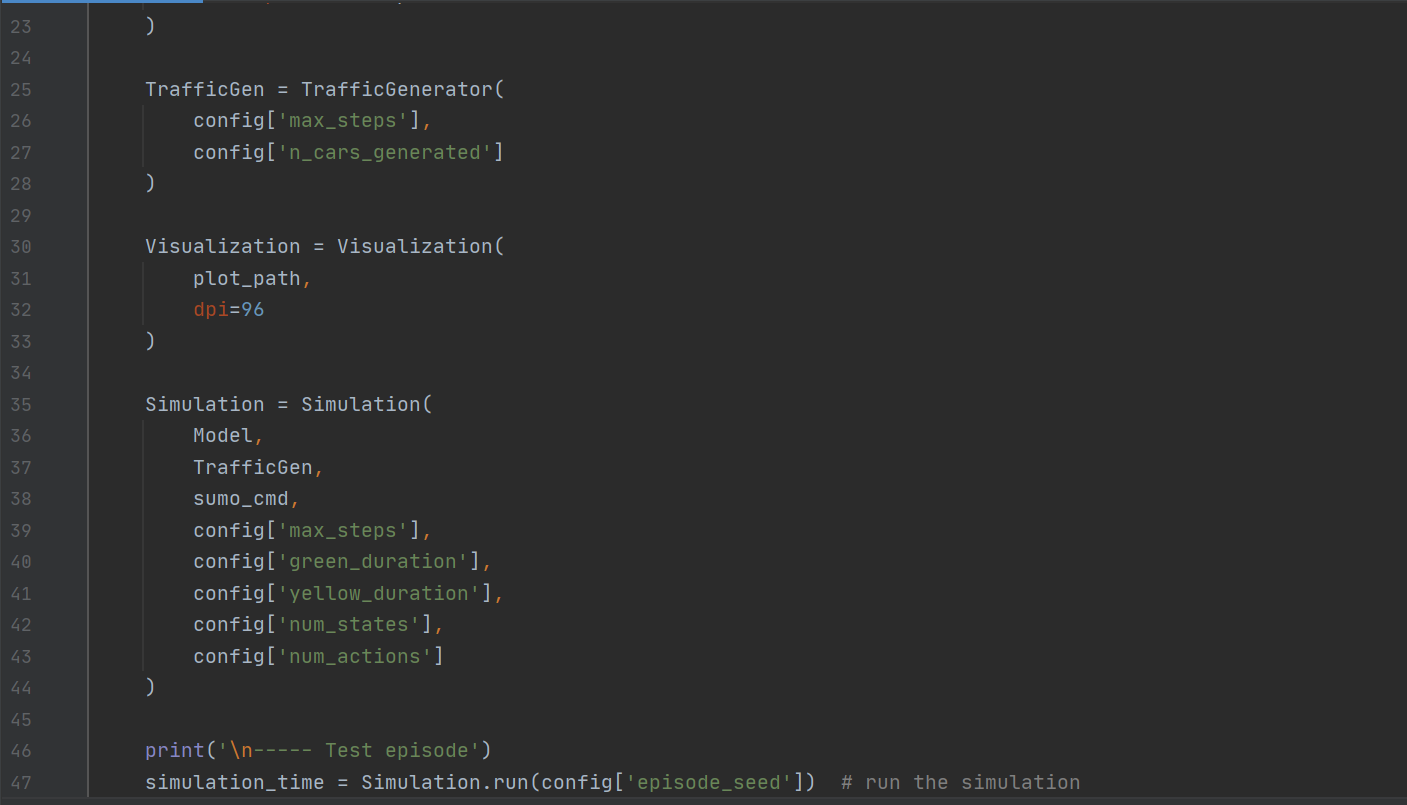






**test\_main.py:**

****

****

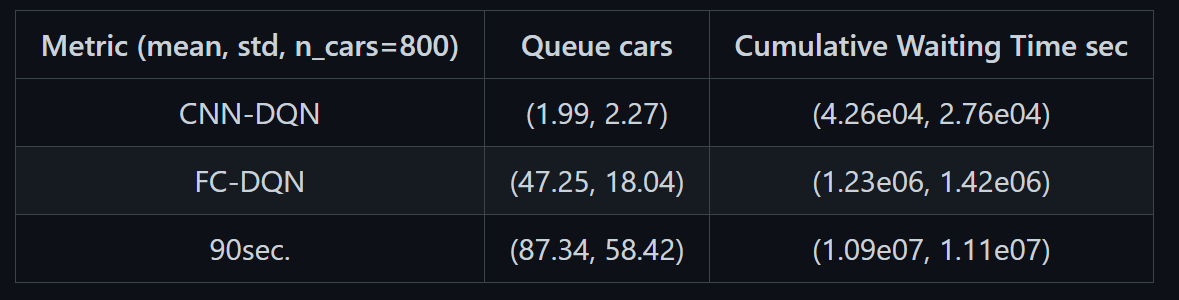
****

Two models are conditioned, one convolutional and the other completely connected, each with 1600 epochs and 4500 steps.

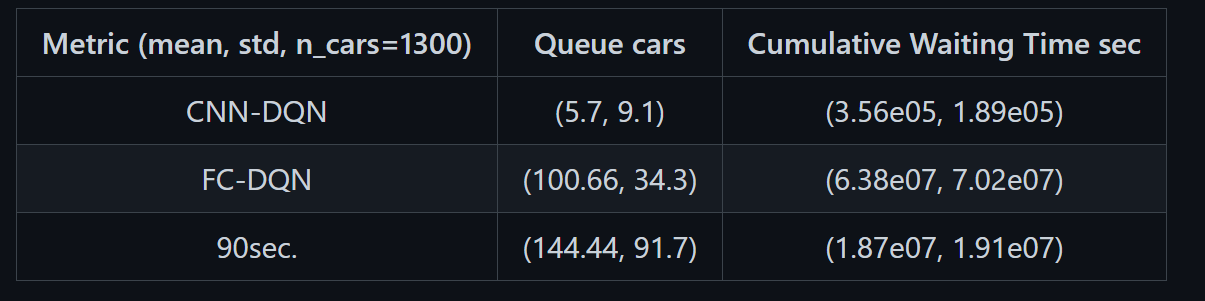
Those models were trained on a cpu, and each one took about 24 hours to complete.

Three scenarios of 800, 1300, and 1800 cars were developed for testing purposes. They reflect three different road states during low, mid, and peak hours, respectively. In the following slide, the results for each are shown in a table.

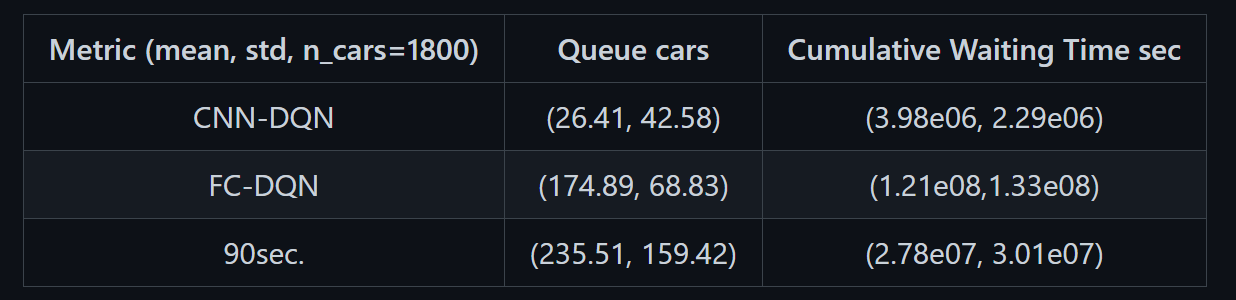
**Comparison of models with 800 cars:**

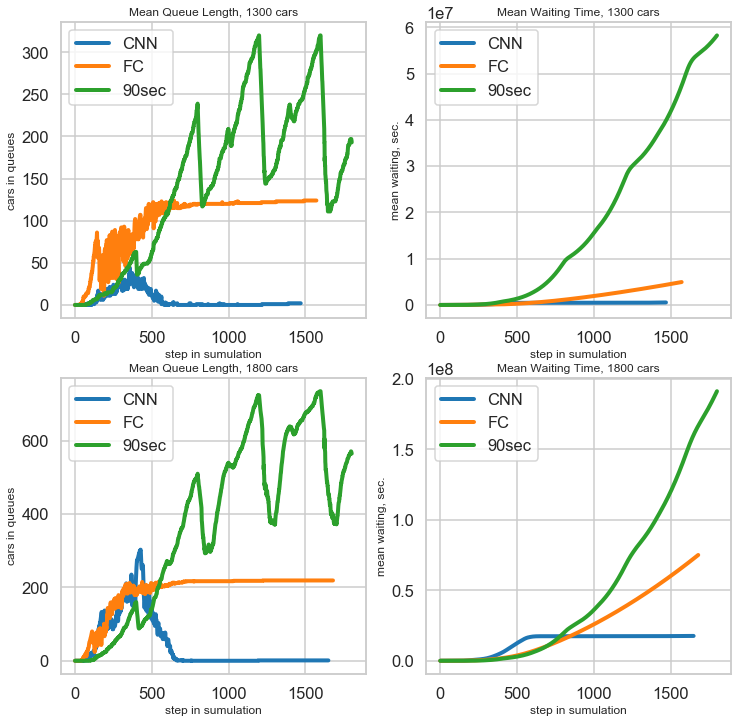


**Comparison of models with 1300 cars:**

****

**Comparison of models with 1800 cars:**

****

****

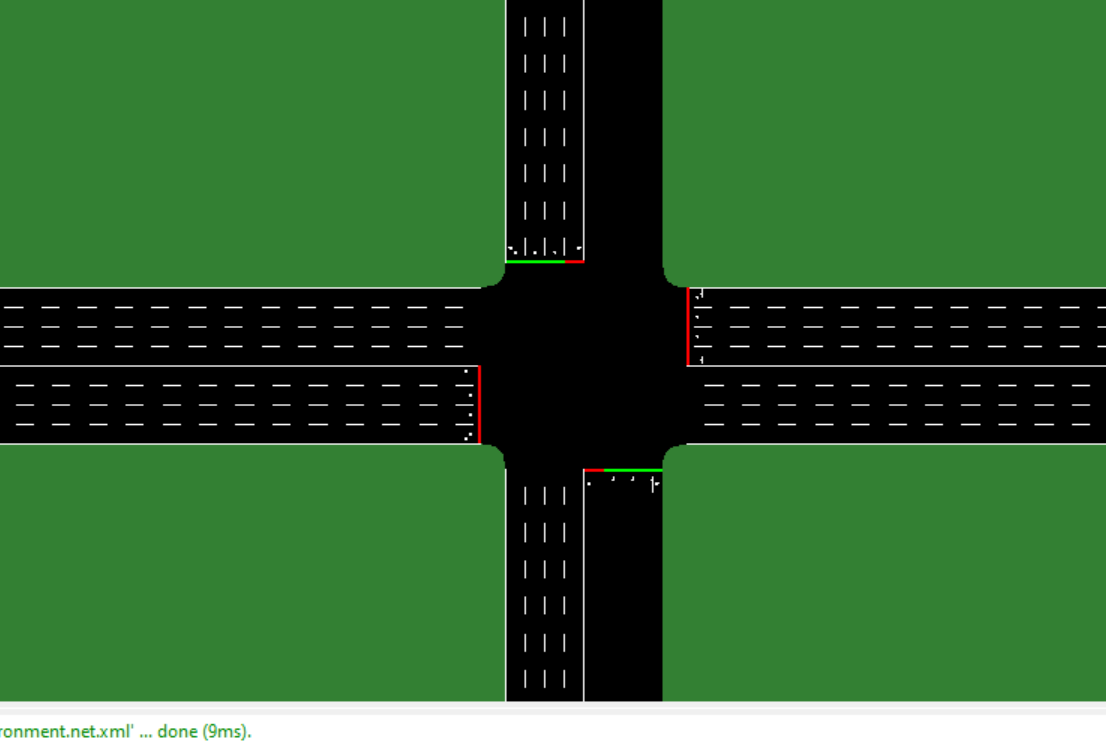
The graph CNN-DQN is the best in the traffic scheme, as you can see here.

When using FC (fully connected) and a 90-second loop, there will be more waiting time and the traffic will be longer.

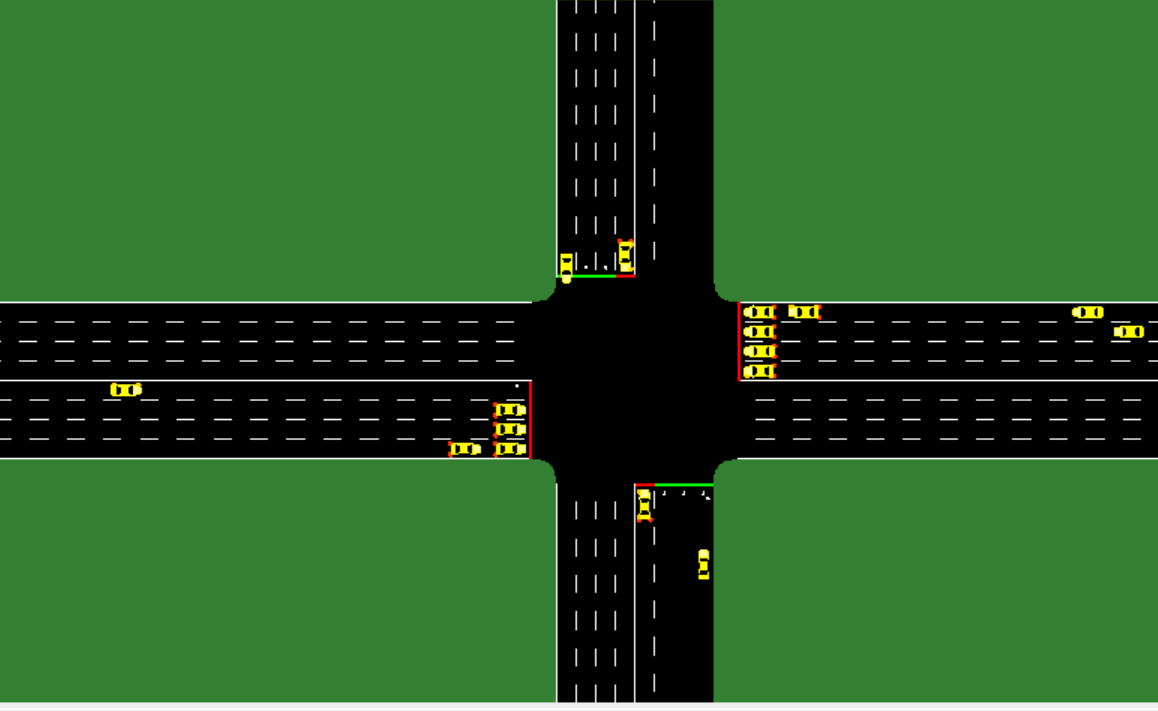
When the load is at its highest, a vehicle stands in line for an average of 117 seconds (CNN). That's 20% better than a constant 90-second loop. As can be shown, FC-DQN is incapable of handling medium road loads, while CNN-DQN is perfectly capable of handling large loads.

**11.Result And Analysis:**

**Output without Traffic:**

****

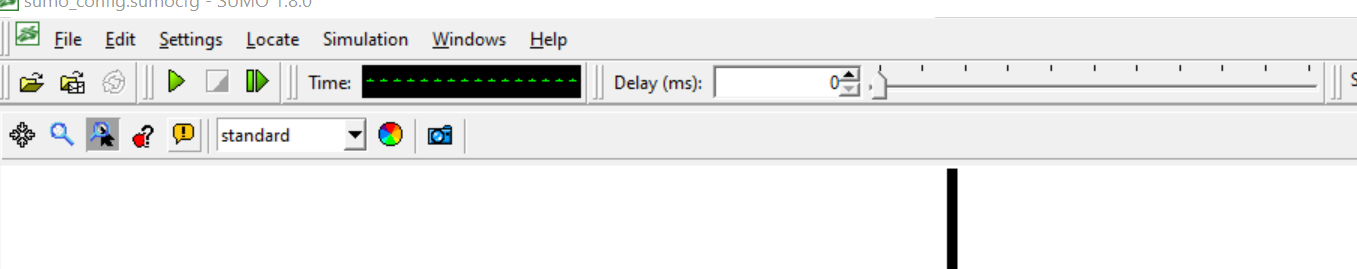
**Output With Less Traffic:**

****

**Output With High Traffic:**

**Output:**

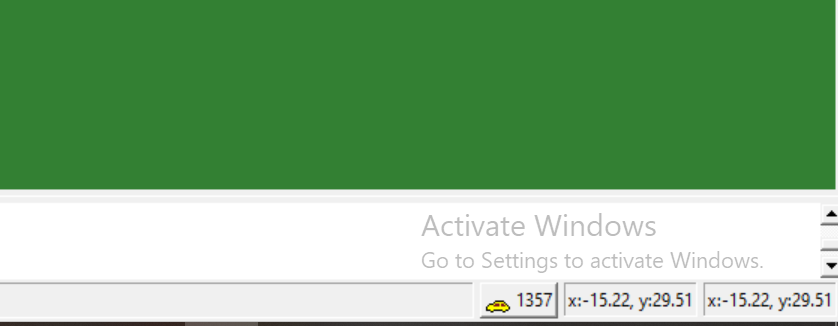
****

****

It is part of the sumo-simulator. We have wide ranges of options for the output.

Standard/Real world/Faster Standard/Rail /Selection.

In the Time column we can see the time taken for the vehicle to all vehicles to cross.



We can know the number of cars present at a particular time. And the position of the cursor.



Using that nozzle we can increase the speed and decrease the speed of the traffic in the traffic.

In the Scale traffic we can increase the Intensity of the traffic. Using that we can increase the traffic in terms of cars.

**12.Conclusion:**

Multiple features of machine learning have been used to build the Smart Traffic System. Machine Learning is used to achieve traffic optimization by allocating varying time to all traffic signals based on the number of vehicles in the road direction. The Smart Traffic System is used to effectively deal with traffic congestion and conduct re-routing at road intersections.

In our project “SMART TRAFFIC SYSTEM” we got best results using CNN(Convolutional Neural Networks)

**13.References:**

1.J. Lloret, S. H. Ahmed, D. B. Rawat, W. Ejaz, and W. Yu, “Editorial on wireless networking technologies for smart cities,” Wireless Communications and Mobile Computing, vol. 2018, Article ID 1865908, 3 pages, 2018.

2. "Internet of things for smart cities," IEEE Internet of Things Journal, vol. 1, no. 1, 2014. A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "IEEE Internet of Things Journal, vol. 1, no. 1, "Internet of Things for Smart Cities.", 2014.

3.Genders Razavi, Wade, and Genders "Traffic signal management using a deep reinforcement learning agent." arXiv:1611.01142 arXiv:1611.01142 arXiv:1611.01142 arXiv:1611.01142 ar (2017).

4..Wade Genders & Saiedeh Razavi (2019): Asynchronous n-step Q- learning adaptive traffic signal control, Journal of Intelligent Transportation Systems, DOI: 10.1080/15472450.2018.1491003