We are creating the Environment based on the specifications. The configuration of the highway is shown in Figure 1.

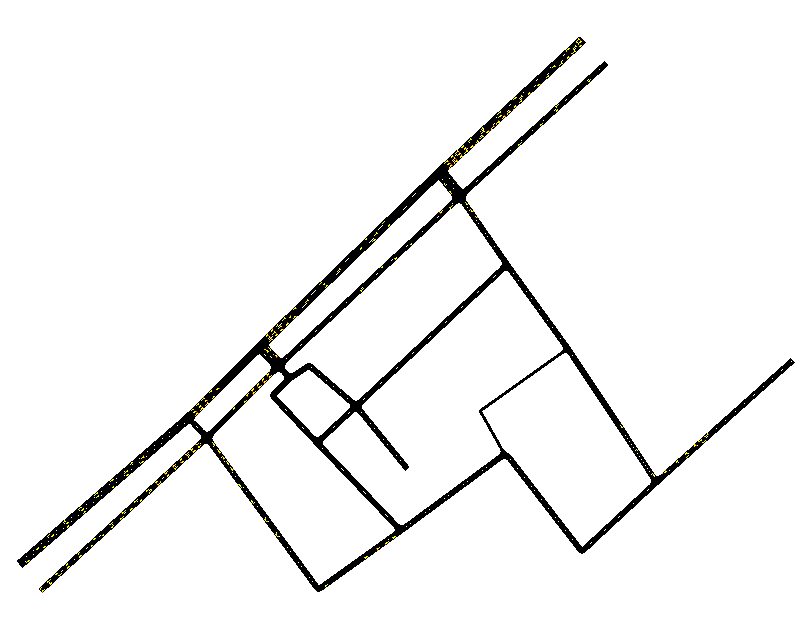


Figure 1. The proposed configuration, resulting from careful planning and consideration, is based on highways and three lanes for each.

Based on the defined version and the specification of the number of cars, it has to be between 1 and 100. The 5G network is crucial in this environment, with 15 information flows considered. The proposed network specification, a crucial part of our project, is defined in the ntust.net.xml file. The number of vehicles in this architecture is 100, and the specifications for these cars are specified in the osm.passenger.trips.xml file. The definition of the care is shown in Figure 2.

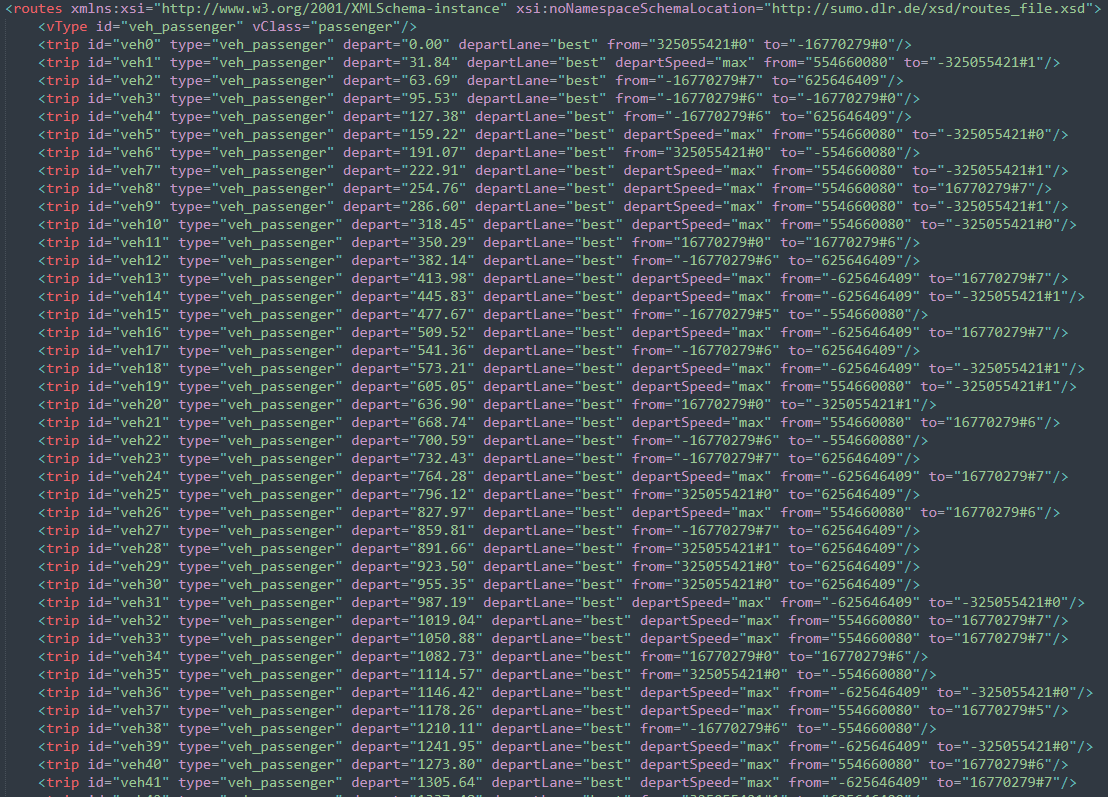


Figure 2. The specification for the road detail specification.

To follow the simulation process, which specifies working with 200 to 300 seconds, the proposed model is simulated for 3300 milliseconds to reach the simulated environment with a 70-millisecond delay. The vehicle's density is 1000. Based on the response time for the message transfer, the BSM message period is between 100 and 1000. The part of the code for message duration is shown in Figure 3. The number of processed edges here is 32. There are 24 junctions in the provided architecture.

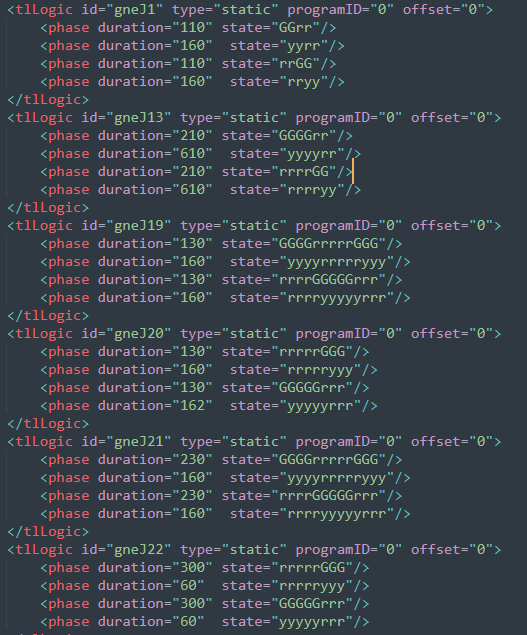


Figure 3. The duration of the transmitted message.

The rest of the proposed signal is defined inside the code processor sumo engine. The architecture for the proposed environment is based on a real-world environment. The final environment is shown in Figure 4.



Figure 4. The environment in the real world.

The next step is to obtain all the available information from the network. To do that, the proposed method extracts every possible state and action from the simulation. To extract information, use Python code from the sumo\_run.py program. The code for running the code and reading the dataset is shown in Figure 5.



Figure 4. The code is to read and load the XML file simulation.

As shown in Figure 4, the proposed solution uses the information from the simulation to extract the headline from the edge and node structures. Checking the code for each head part of the simulation can be seen in Figure 5.

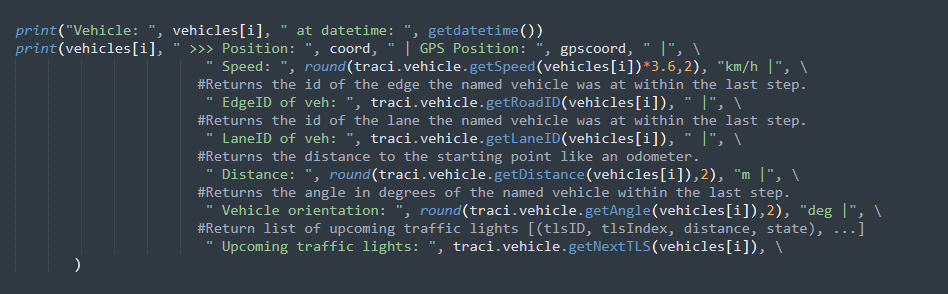


Figure 4. The extracted features from the simulation.

As shown in Figure 5, the extracted features are: 'date and time,' 'vehicle id,' 'coord,' 'GPS coord,' 'speed,' 'edge,' 'lane,' 'displacement,' 'turning Angle,' 'Next TLS,' 'tf light,' 'tl state,' 'tl phase duration,' 'tl lanes controlled,' 'tl program,' 'tl next switch'. This information will be used further to define the state, action, reward, and architecture of the DRL model. The speed and displacement of the vehicle are shown in Figures 5 and 6 to get deeper into the extracted features.

Figure 5. The change in the speed after 1000-time steps.

Figure 6. The change in the car displacements after 1000-time steps.

This is shown by checking the progress in the communication between each car and the speed it takes to go from one location to another in the simulation and progress. The distance that is brought to go from one location to another increases over time. The behavior of each car can be tracked and checked throughout the road. The behavior of the traffic lights is recorded as well. The situation and duration of each light is saved in the following formats:

1. (('gneJ3', 4, 223.96, 'r'), ('gneJ19', 5, 327.83000000000004, 'r'), ('gneJ21', 5, 586.85, 'r'))
2. (('gneJ3', 4, 210.51913074060195, 'r'), ('gneJ19', 5, 314.38913074060196, 'r'), ('gneJ21', 5, 573.409130740602, 'r'))
3. (('gneJ22', 0, 191.01000000000002, 'r'), ('gneJ20', 0, 452.40000000000003, 'r'), ('gneJ5', 0, 562.27, 'r'))

These three are samples of the green and red lights and, on each episode, how many of the lights have occurred in each junction. Another essential piece of information is the light information on each junction road ahead of us through the path that can be taken by the vehicle. A sample of the information for the next lights is shown as follows:

1. G G G r r r G G G.
2. G G G r r r G G G.
3. r r r r r G G G.

The tuning angle on each road shows the model's actions. It also shows the best route to take and the best lane to follow to reach the destination. Figure 7 shows a sample of the turning angle information.

Figure 7. Angle of tuning for the first 1000-time steps.

Figure 7 depicts information on seven types of actions and how to use the proposed method for the action state sequence. The junction information is also shown in Figure 8.

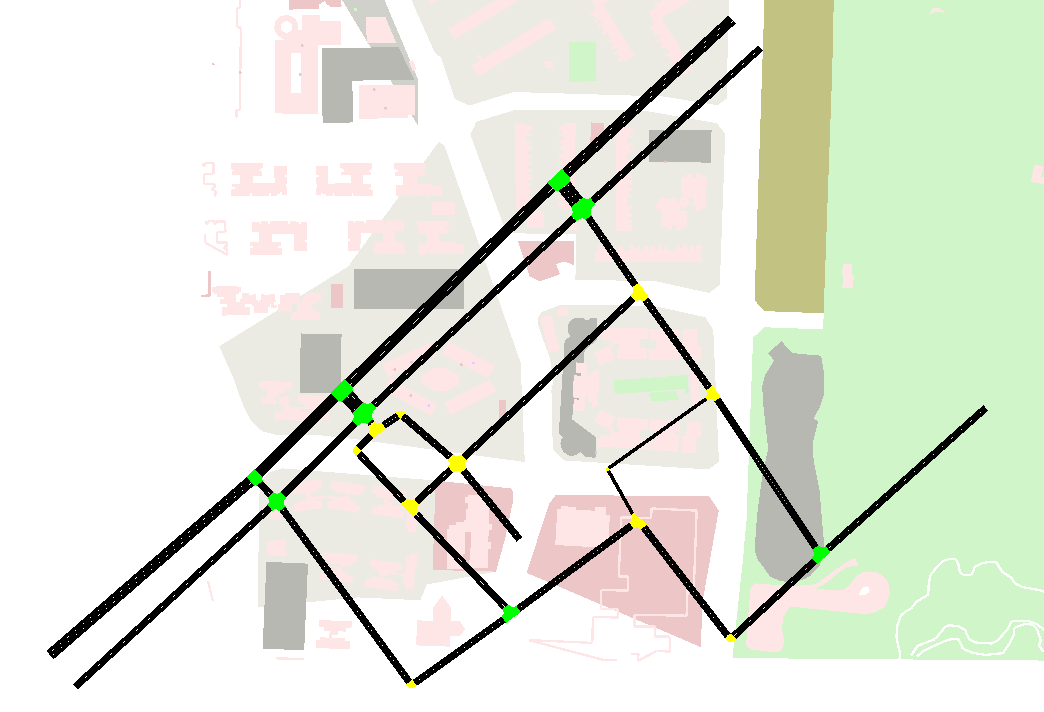


Figure 8. Number of junctions on the simulation environment.