IoT Traffic Simulation: Synthetic Modelling  
Rishabh Jain, Prof. Robert Brunner

Abstract

With the advent of self driving cars and other electric cars, technology has already made its way in modern day traffic. With technology controlling how vehicles would move on the road, there is increased need for safety and risk handling. IoT Traffic Simulation proposes a network of sensors laid out on the roads and on vehicles for better vehicle management. By correlating weather and traffic, the prediction model will predict useful vehicle patterns.

**Keywords:** synthetic, data modelling, prediction

# Introduction

This paper explores the realm of synthetic modelling. Due to lack of significant data of weather and vehicle patterns for different coordinates, we created our own data models. The first one simulates weather conditions such as temperature, wind speed, precipitation and humidity based on a specific coordinates and the second one aims to use this information to create pseudo vehicle trajectories that can be correlated to the conditions around the vehicle.

The aim is to create a enough data with different situations so that based on a system of vehicles so that the prediction model can both train and test on the given data. All the data generated was pushed to Kafka queues.

# Data Generation Overview

The synthetic data generation effort was broken in parts and different cases were considered.

## Straight Highway (Single Vehicle)

The most simple and beginning goal was to create a model that generates coordinates for a vehicle travelling along a straight highway. We had to pick a random coordinate which could be the starting point and then look for the next possible coordinate, which was **x** distance apart and in a certain direction (**theta)**. The direction was kept constant throughout this task. At first, we took the speed of the car to be constant so the **x** was also constant. In later iterations, we fluctuated the speed by a small amounts keeping the overall speed in a certain acceptable driving velocity range. In this task there was no influence of weather conditions affecting the velocity of the vehicle.

## Straight Highway (Multiple Vehicles)

An extension of 2.1 was a model that simulated a single vehicle along a straight highway and then ran multiple vehicles (as asked) along the same set of coordinates generated. To maintain the randomness of vehicle movement, the other vehicles were simulated with different speeds at different points of time. Again, there were fixed thresholds (both minimum and maximum) that kept the vehicle at a reasonable velocity.

## Straight Highway with Sudden Velocity Dip

Before bringing in the weather conditions into play, we created a model that simulated the velocities of the vehicles such that after a breif amount of time all the vehicles would have a sudden dip in their velocity trend. This was a more generalized version of handling situations such as diversions, road blocks, stopped vehicle in the middle of the road. This reused the implementation from sections 2.1 and 2.2 and introduced a random variable that would cause the sudden dip between 1/3rd and 2/3rd of the total data.

## Accidents

Accidents are a major issue that we aim to resolve through the approach of Traffic Simulation, hence we created a model that could induce random accidents along the road where two cars that are in close proximity. The accident would cause both the cars to come to complete rest and the other cars to slow down the velocity. The limitation of the model is that all the cars slow down as a result of an accident before the point of accident, which might be a little different from a real world scenario.

## Turns

We needed to incorporate turns in our data sets so that we could harness more real life situations. The 90 degree left and right turns were included as the starting point. As of now, all the cars along the road turn as it is being assumed the there was a dead-end at the end of the road. There is a need to generate data such that only a fraction of the cars take the turn and others keep on continuing on the same road.

## Weather Data

The weather data did not play any role in determining the velocity of the vehicle till now. We integrated the temperature so that it affects the vehicle velocity. Right now, the relation is weak as it only depends on the temperature, which does not give us much information about the road conditions.

## Weather Changes – (Temperature, Snow, Wind Speed)

Need to take other factors such as temperature, snow and wind speed into account. In the table below 0 represents a decrease in that quantity and 1 represents an increase in the quantity. The last two cases of the truth table namely, increase in temperature and snow and increase in temperature, snow and wind speed yield not applicable since the assumption is that increase in snow will not increase the temperature. If this case happens it would bring no change to the speed of the vehicle. However, certain thresholds were kept to ensure that minute changes are not considered. These changes played a role in determining the change in the vehicle speeds.

|  |  |  |  |
| --- | --- | --- | --- |
| **Temperature** | **Snow** | **Wind Speed** | **Change in Vehicle Speed** |
| 0 | 0 | 0 | No change |
| 0 | 0 | 1 | Slower by 1x |
| 0 | 1 | 0 | Slower by 2x |
| 0 | 1 | 1 | Slower by 3x |
| 1 | 0 | 0 | No change |
| 1 | 0 | 1 | Slower by 1x |
| 1 | 1 | 0/1 | Not Applicable |

## Turns (Left and Right)

The vehicles can randomly turn as there is a random variable introduced for turns as well. The model has a forty percent probability of deviating from the straight line right now. The theta was changed accordingly to 180 and 0.

## U-Turn

The U-Turn was added by changing the theta to 270 as we started with 90 degrees as our seed directions. The limitiation about turns is detecting accidents while turning. We need to maintain a set of current points for all the cars and check while turning if any two cars are colliding.

## Introduce different types of vehicles

We introduced different types of vehicles, mainly in three categories: light motor vehicle, heavy motor vehicle and two-wheelers. All the vehicles have unique id’s attached to them that can be used to identify the vehicle. The obvious choice was Python’s UUID library. The module provides immutable UUID objects for generating version 1, 3, 4 UUIDs as specified in RFC4122. [c] We used uuid4() to generate the unique id’s as the chance of collision is very small.

References

[c] [https://docs.python.org/2/library/uuid.html]