



清华大学 深圳国际研究生院  
Tsinghua Shenzhen International Graduate School

结题汇报

# 车联网的仿真模拟、边端通信 模拟及道路质量评估

土木与建筑工程CAE——大作业

小组成员：陈洋、付磊

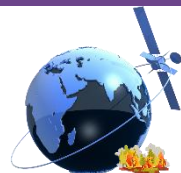
2023.6.9





# 项目调整

## 基于卫星影像和神经网络的森林火灾时序预测



### 项目局限性

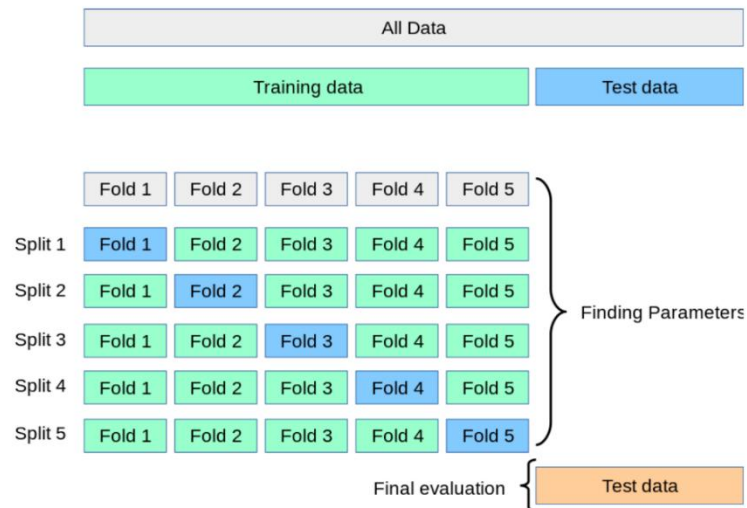
Output (19.5GB / 19.5GB)

```

/kaggle/working
├── S2B_MSIL1C_20200529T004709_N0209_R102_T53HPA_20200529T0210
├── S2B_MSIL1C_20220310T004709_N0400_R102_T53HPA_20220310T0214
├── S2A_MSIL1C_20220315T004711_N0400_R102_T53HPA_20220315T0214
├── S2A_MSIL1C_20170629T004701_N0205_R102_T53HPA_20170629T0053
├── S2A_MSIL1C_20160106T004742_N0201_R102_T53HPA_20160106T00522
├── S2B_MSIL1C_20190912T004709_N0208_R102_T53HPA_20190912T0210
├── S2B_MSIL1C_20171211T004659_N0206_R102_T53HPA_20171211T02034
├── S2A_MSIL1C_20171206T004701_N0206_R102_T53HPA_20171206T03552
├── S2B_MSIL1C_20200519T004709_N0209_R102_T53HPA_20200519T0210
├── S2A_MSIL1C_20200504T004711_N0209_R102_T53HPA_20200504T0227
├── S2B_MSIL1C_20181027T004709_N0206_R102_T53HPA_20181102T12592
├── S2B_MSIL1C_20170823T004659_N0205_R102_T53HPA_20170823T0053
├── S2A_MSIL1C_20200514T004711_N0209_R102_T53HPA_20200514T0230
├── S2B_MSIL1C_20190922T004709_N0208_R102_T53HPA_20190922T0210
├── S2A_MSIL1C_20210628T004711_N0300_R102_T53HPA_20210628T0225
├── S2A_MSIL1C_20190907T004711_N0208_R102_T53HPA_20190907T02210
├── S2B_MSIL1C_20200509T004709_N0209_R102_T53HPA_20200509T0210
├── S2B_MSIL1C_20191022T004709_N0208_R102_T53HPA_20191022T0209
├── S2B_MSIL1C_20201125T004709_N0209_R102_T53HPA_20201125T0206
├── S2A_MSIL1C_20161101T004702_N0204_R102_T53HPA_20161101T00524
├── S2A_MSIL1C_20171226T004701_N0206_R102_T53HPA_20171226T03552
├── S2A_MSIL1C_20190927T004711_N0208_R102_T53HPA_20190927T0222
├── S2A_MSIL1C_20161121T004702_N0204_R102_T53HPA_20161121T00542
├── S2A_MSIL1C_20180405T004711_N0206_R102_T53HPA_20180405T0216
├── S2A_MSIL1C_20210618T004711_N0300_R102_T53HPA_20210618T0228
├── S2A_MSIL1C_20210608T004711_N0300_R102_T53HPA_20210608T0224
├── S2A_MSIL1C_20190917T004711_N0208_R102_T53HPA_20190917T02210
├── S2B_MSIL1C_20220330T004709_N0400_R102_T53HPA_20220330T0213
├── S2B_MSIL1C_20210603T004709_N0300_R102_T53HPA_20210603T0216

```

- 经过筛选约有2015年至今共**323组**数据。
- 预计有**217G**的数据集，**没有合适的服务器**，而kaggle只允许**单个数据集20G**的输入。



- 25个区域的**交叉验证策略**虽然平衡了训练集与验证集，但会使得模型的**泛化能力较差**。
- 对于时间序列预测任务来说，323组数据的**数量相对较少**，可能使模型**过拟合**。

0.053	0.063	1	1	1
0.002	0.002	0.95	0.6	0.58
0.007	0.062	0.05	0.008	0.015
0.016	0.029	0.032	0.007	0.012
0.022	0.017	0.02	0.008	0.92

- 火灾发生次数过少，不适合做预测，应该做**火点识别**和**火灾监测**。
- 由于极轨卫星的特殊性，只能在**未来5天**划分的**25个超大地区范围内**进行**火灾高风险预警**，意义不大。

另寻到能够充分兼顾CAE课程所教授各章节的内容的项目——





# 项目调整

## 车联网：边缘计算典型应用场景



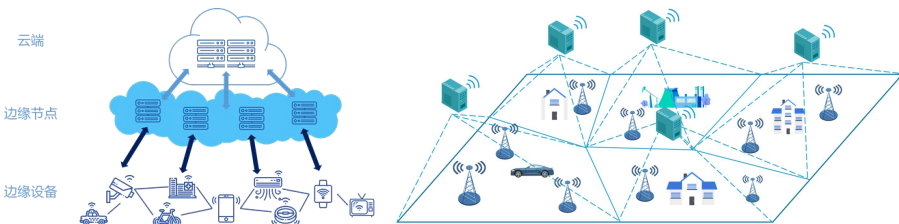
**车联网：**通过车载传感器和无线通信技术，使汽车具备了**数据采集、处理和传输**的能力。

### 实时性要求

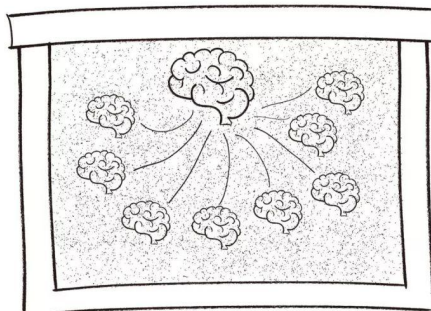
边缘计算通过将计算和数据处理**推向靠近车辆的边缘节点**，实现了**更快的数据传输和处理速度**。

### 可靠性要求

边缘计算通过在边缘节点上进行数据处理，**减少了对云服务器**的依赖，从而降低数据传输过程中的安全风险。

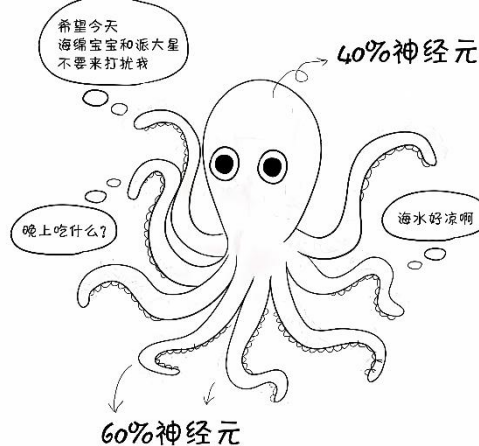


章鱼在捕猎时异常灵巧迅速，腕足之间配合极好，从不会缠绕打结。这得益于它们类似**分布式计算**的“多个小脑+一个大脑”。



边缘计算也属于一种分布式计算：在网络边缘侧的智能网关上就近处理采集到的数据，而不需要将大量数据上传到远端的核心管理平台。

章鱼是用“腿”来思考并就地解决问题的



**边缘计算：**一种分布式计算模型，将计算和数据处理能力**靠近数据源**和最终用户，**减少数据传输延迟**和网络拥塞。

### 实时数据处理和决策

车联网中产生大量的数据，包括车辆状态、传感器数据、驾驶行为等等。通过**在边缘设备上**进行实时数据处理和分析，实现快速响应和实时决策。

### 与基础设施的互联互通

实现车辆与交通基础设施的实时互联互通，通过将计算能力推向边缘设备，车辆可以与周围的**智能交通系统、红绿灯**等进行实时通信和协作。

### 车辆间的协同和通信

通过车辆之间的边缘设备进行实时的车辆间通信（V2V），并用于**交通安全警示、自适应巡航控制、交通流优化**等功能。



# 基于Veins、SUMO及OMNET++的车联网仿真模拟

## 车联网仿真 - 软件介绍

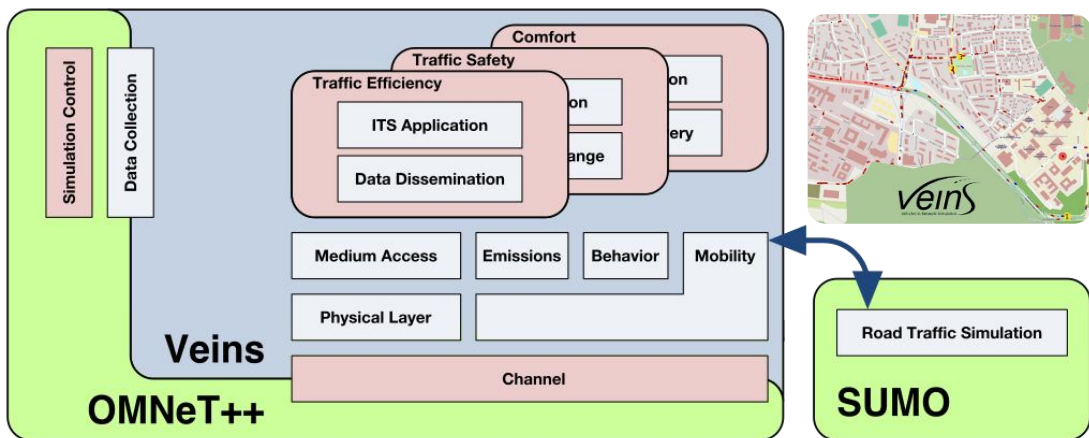
**Veins**是一个基于OMNeT++的开源模拟框架，用于车联网仿真模拟车辆之间的通信和交互。



**SUMO**是一个开源的交通仿真软件，用于模拟城市交通的运行和车辆的移动。



**OMNeT++**是一个开源的离散事件仿真框架，用于模拟各种通信网络和分布式系统。



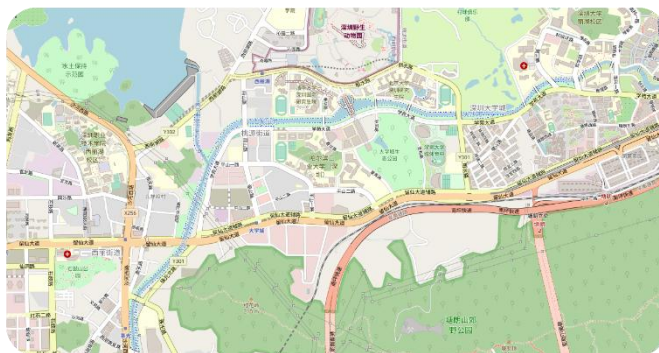
## 协同工作原理

- **SUMO**作为服务器通过TraCI提供通信API,
- **Veins**作为客户端，通过Python脚本与**SUMO**通信。
- **Veins**通过**OMNet++**创建的**车辆实例**通过TraCI由**SUMO**仿真后回传显示。



## 项目对象选取

- 选取北纬22.5977°至22.5845°，东经113.9550°至113.9784°区域
- 从[www.openstreetmap.org](http://www.openstreetmap.org)中下载导出.OSG地图文件
  - OSM (OpenStreetMap) 文件是一种开放源代码的地理数据文件格式，包含地理要素的几何形状和属性信息、交叉口节点、道路、关系、标签等信息。
- 将.OSG文件转化为.net.xml文件以及.poly.xml文件
  - .net.xml文件包含了一个完整的道路网络的拓扑结构、车道、连接关系以及车辆流量等信息
  - .poly.xml文件是用于描述地理区域的多边形边界信息的文件格式，生成建筑障碍



选取大学城区域作为对象



SUMO中加载.net.xml文件



# 基于Veins、SUMO及OMNET++的车联网仿真模拟

## OMNET++主要工作文件

- 通信模型和协议实现：  
消息传输，路由选择
- 车辆行为模拟：  
速度控制，移动方式
- 仿真事件处理：  
响应消息，触发信号灯

Message.cc

OMNET++

RSUScenario.NED

- 网络拓扑结构：  
车辆节点，RSU，基站
- 模块的属性和参数：  
通信范围，缓冲区
- 定义消息和事件：  
消息的结构及类型

omnetpp.ini

模块配置：车辆节点的通信范围、传输速率，路边单元 (RSU) 的位置

仿真参数：仿真起始时间，运行的步长，定义统计信息的收集间隔

通信配置：通信模型和传输协议以及障碍物引起的信号衰落模型

umap.net.xml

路网拓扑结构：道路的几何形状、连接关系和拓扑结构

路口和交通信号灯：用于模拟车辆在路口的交通行为和交通信号的控制

路段属性：道路长度、车道数量、速限、道路类型、车道宽度、车道分隔方式

umap.poly.xml

多边形边界：地理区域的形状和边界线的坐标点

邻接关系：多边形区域之间的连接和接壤关系

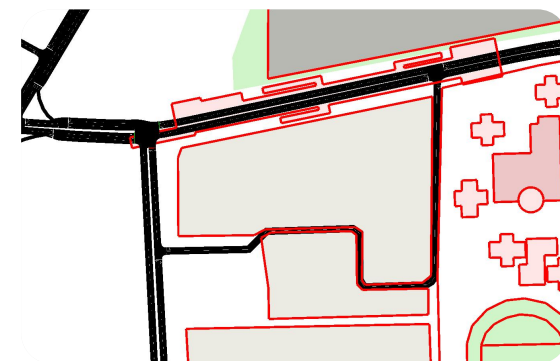
区域属性：区域的名称、标识符、类型、人口密度、土地利用

umap.rou.xml

车辆轨迹：由道路序列指定经过的道路、路口和交叉口等

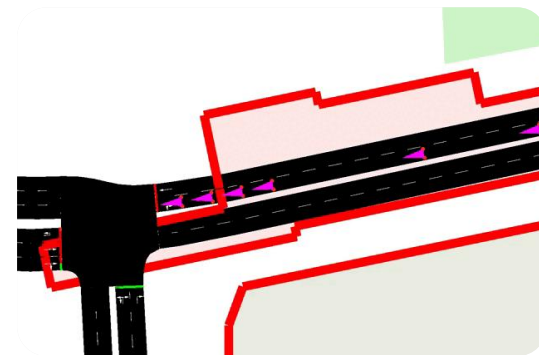
车辆属性：车辆的类型、长度、宽度、最大加速度、减速度、转向角度等

车辆行为：路线选择和行驶行为



umap.sumo.cfg

建立OMNET++与Sumo的实时通信





# 基于Veins、SUMO及OMNET++的车联网仿真模拟

## 部分仿真参数及结果展示

```

▼ NIC-Settings
1 #####
2 #          11p specific parameters          #
3 #          NIC-Settings                    #
4 #####
5 *.connectionManager.sendDirect = true
6 *.connectionManager.maxInterfDist = 2600m
7 *.connectionManager.drawMaxIntfDist = false
8
9 ***.nic.mac1609_4.useServiceChannel = false #MAC层
10 ***.nic.mac1609_4.txPower = 20mW
11 ***.nic.mac1609_4.bitrate = 6Mbps
12 ***.nic.phy80211p.sensitivity = -89dBm
13 ***.nic.phy80211p.useThermalNoise = true
14 ***.nic.phy80211p.thermalNoise = -110dBm
15 ***.nic.phy80211p.decider = xmldoc("config.xml")
16 ***.nic.phy80211p.analogueModels = xmldoc("config.xml")
17 ***.nic.phy80211p.usePropagationDelay = true
18 ***.nic.phy80211p.antenna = xmldoc("antenna.xml", "/root/Antenna[@id='monopole']")

```

对RSU(路侧单元)的MAC层ned(nic), 以及信号传播范围的影响。

```

▼ Mobility
1 #####
2 #          Mobility                        #
3 #####
4 *.node[*].veinsmobilityType.debug = true
5 *.node[*].veinsmobility.x = 0
6 *.node[*].veinsmobility.y = 0
7 *.node[*].veinsmobility.z = 1.895
8 *.node[*0].veinsmobility.accidentCount = 1 #发生事故次数
9 *.node[*0].veinsmobility.accidentStart = 75s #发生事故时间
10 *.node[*0].veinsmobility.accidentDuration = 50s #事故持续时间

▼ WaveApplayer
1 #####
2 #          WaveApplayer                  #
3 #####
4 *.node[*].applType = "TraCIDemo11p"
5 *.node[*].appl.headerLength = 80 bit
6 *.node[*].appl.sendBeacons = true
7 *.node[*].appl.dataOnSch = false
8 *.node[*].appl.beaconInterval = 1s

```

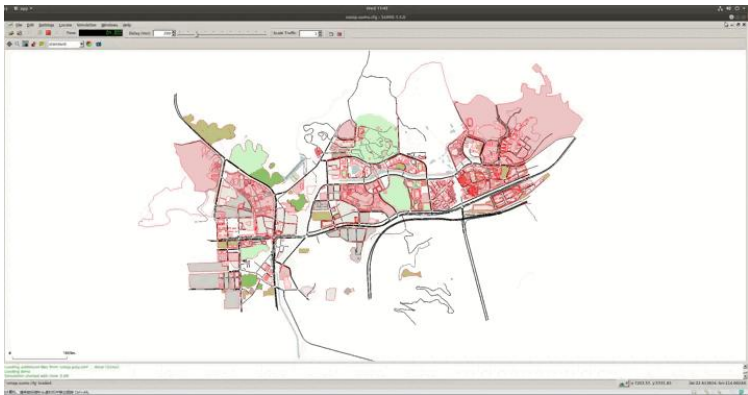
移动节点移动模块的设置, 事故发生次数以及时间, 应用层的信标间歇等相关设置。

```

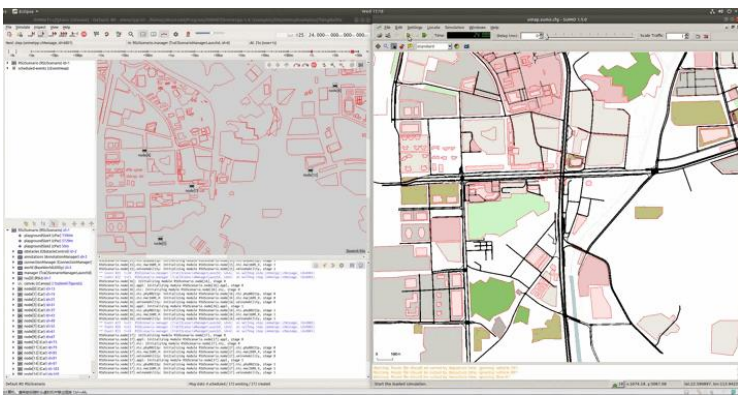
▼ trip-0.rou.xml
1 <!-- Routes + associated vehicles & flows -->
2 <route id="trip-0" color="Orange" edges="1158676499#4 772812917#6 772812917#11 772812917#12 772812917#13 772812917#14 772812917#15 1050942980#0 1050942980#1 529070129#0 529070129#2 529070129#3 529070129#5 1050942979-AddedOnRampEdge 1050942979 1052075121#0 772812900 772812898 772812897#0 772812897#1 636236886#0 636236886#1 636236887#0-AddedOnRampEdge 636236887#0 636236887#1 636236887#1-AddedOffRampEdge 636237032 1051539528#0-AddedOnRampEdge 1051539528#0 1051539528#1 1051539528#1 1051539549#1 1051539549#1-AddedOffRampEdge 1051539549#2 10518178 16 1051539531#0 1051539531#2 1051539531#3 1051539536 1051539536#0 534908412#0 534908412#2 -249695488#0 249695488#1"/>
3 <vehicle id="vehicle-0" type="passenger" color="Magenta" depart="0.0" route="trip-0"/>
4 <vehicle id="vehicle-10" type="passenger" color="Magenta" depart="1.0" route="trip-0"/>
5 <vehicle id="vehicle-20" type="passenger" color="Magenta" depart="2.0" route="trip-0"/>
6 <vehicle id="vehicle-30" type="passenger" color="Magenta" depart="3.0" route="trip-0"/>
7 <vehicle id="vehicle-40" type="passenger" color="Magenta" depart="4.0" route="trip-0"/>
8 <vehicle id="vehicle-50" type="passenger" color="Magenta" depart="5.0" route="trip-0"/>
9 <vehicle id="vehicle-60" type="passenger" color="Magenta" depart="6.0" route="trip-0"/>
10 <vehicle id="vehicle-70" type="passenger" color="Magenta" depart="7.0" route="trip-0"/>
11 <vehicle id="vehicle-80" type="passenger" color="Magenta" depart="8.0" route="trip-0"/>
12 <vehicle id="vehicle-90" type="passenger" color="Magenta" depart="9.0" route="trip-0"/>
13 <flow id="flow-0" type="ignoring" color="Yellow" number="1" begin="0.0" end="1.0" route="trip-0"/>

```

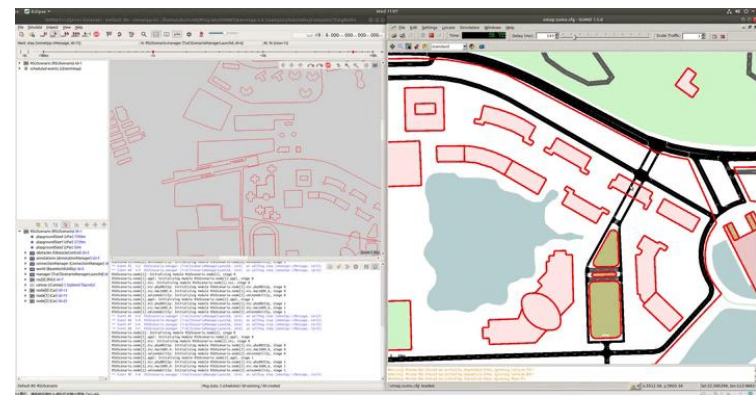
依据道路ID进行车流轨迹的制定, 以及车辆相关信息, 选择轨迹等



信号传播



仿真结果

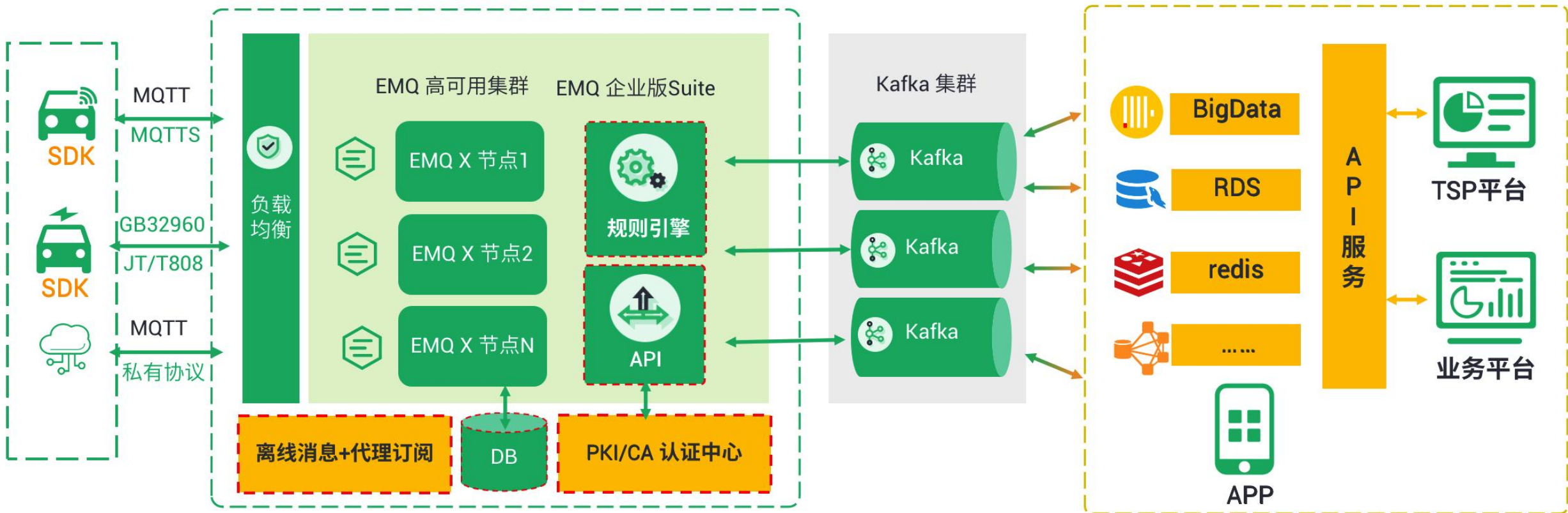


车流仿真



# 车联网边端通信模拟

## 车联网架构分析 —— 端边通信

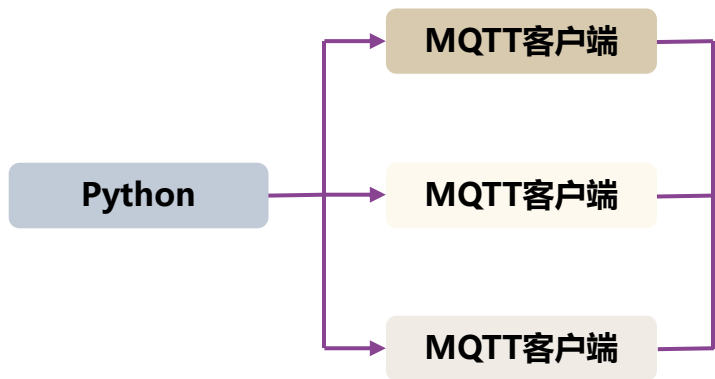


**模拟流程:** 使用Python语言构造一个数据模拟器用来产生汽车传感器数据。使用MQTT作为传输协议，确立EMQ作为MQTT Broker，在EMQ中构建桥接Kafka的插件和规则引擎，实现将数据被预处理后传输给Kafka。

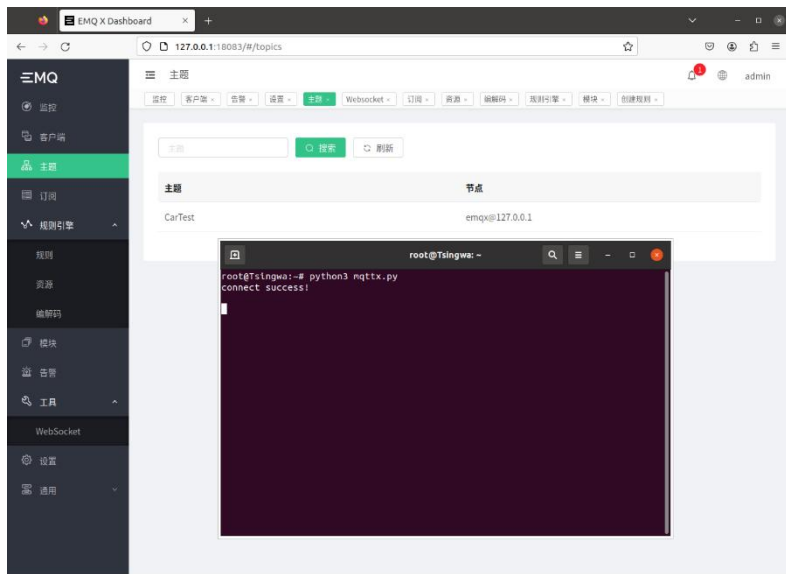
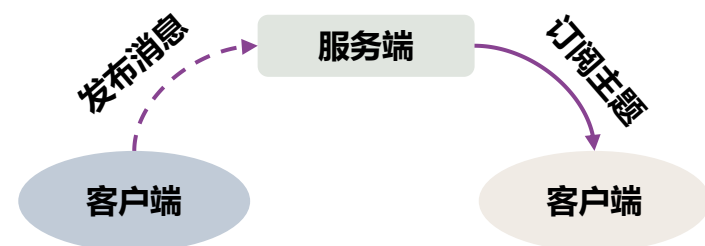
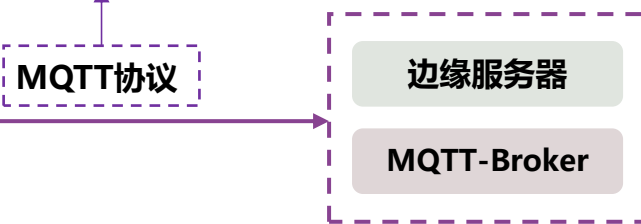


# 车联网边端通信模拟

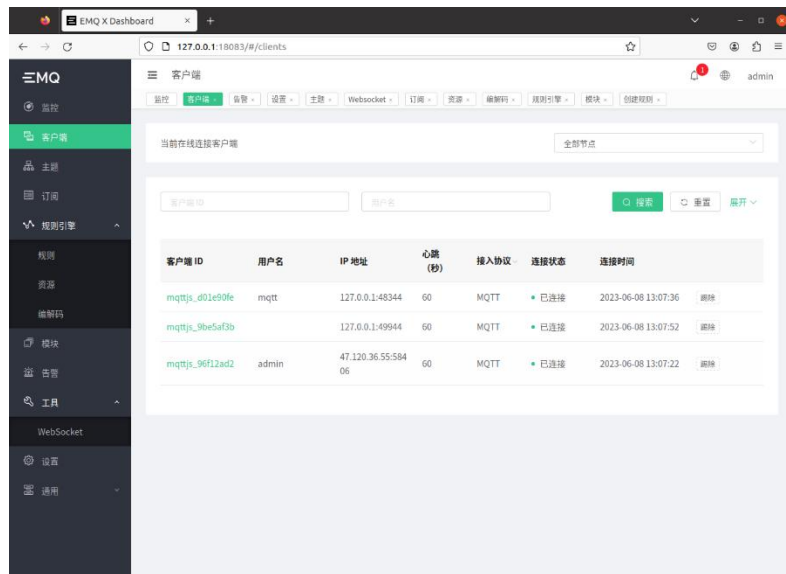
## 数据产生及MQTT协议通信



**MQTT:** MQTT 的运行模式为主从栈模式，主栈是一个代理服务器 (broker)，从栈是各个客户端 (client)，各个客户端 (client) 仅与broker 进行通信。



Python实现发布主题



客户端订阅连接

- 信息的传输是通过**主题 (topic)** 管理的。发布者有需要分发的数据时，其向连接的消息代理发送携带有数据的控制消息。
- 代理会向**订阅此主题的客户端**分发此数据。发布者不需要知道订阅者的数据和具体位置；同样，订阅者不需要配置发布者的相关信息。
- 如果消息代理接收到某个主题上的消息，且这个主题**没有任何订阅**，那么代理就会**丢弃之**，除非发布者将其标记为保留消息 (retained message)。





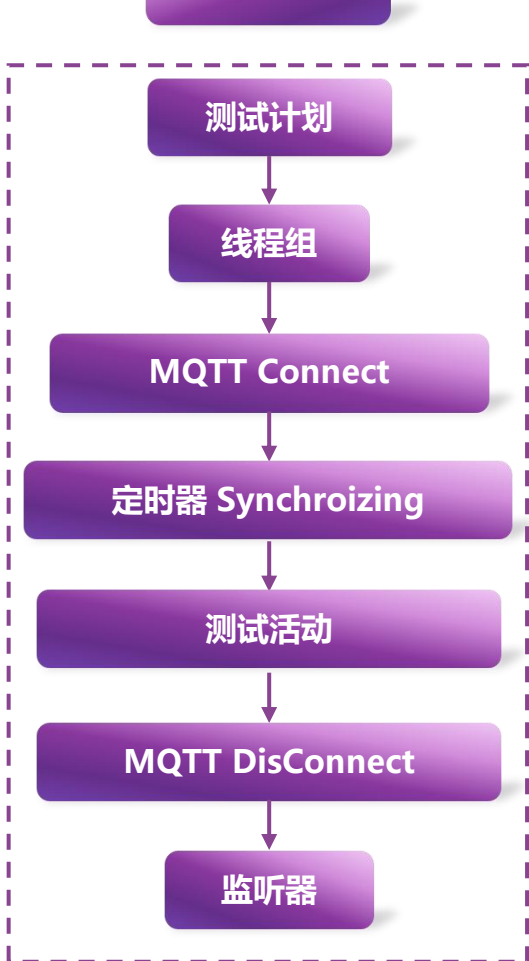
# 车联网边端通信模拟

## 万级终端模拟测试

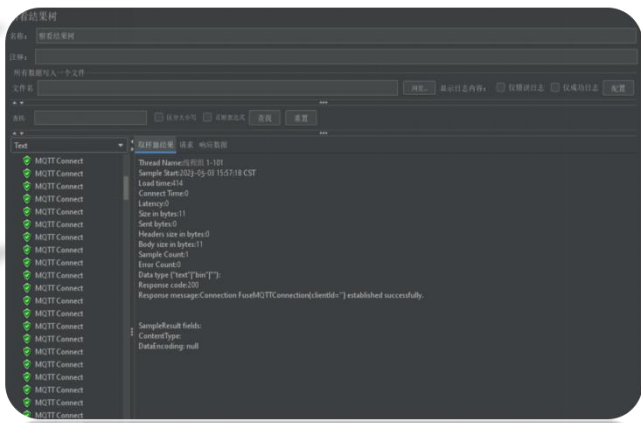
## JMeter模拟万级数据进行测试

### 测试结果

### 测试流程



### 结果树



### EMQX Dashboard

客户端 ID	用户名	IP 地址	心跳 (秒)	会话过期间隔 (秒)	当前订阅数	连接状态
conn_5823514bd07...	gatewayse...	60...	300	0	0	已连接
conn_Sec12a9da94a...	gatewayse...	60...	300	0	0	已连接
conn_29b862d206f...	gatewayse...	60...	300	0	0	已连接
conn_2ff40de07e7c4...	gatewayse...	60...	300	0	0	已连接
conn_07e23b5356f1...	gatewayse...	60...	300	0	0	已连接
conn_110983fc1fb44...	gatewayse...	60...	300	0	0	已连接
conn_a081f369e985...	gatewayse...	60...	300	0	0	已连接
conn_1994e741567e...	gatewayse...	60...	300	0	0	已连接
conn_1515f0a29683...	gatewayse...	60...	300	0	0	已连接
conn_Sec77b33fb78...	gatewayse...	60...	300	0	0	已连接

### 聚合报告

#### Aggregate Report

Name: Aggregate Report

Comments:

Write results to file / Read from file

Filename:   Log/Display Only:  Errors  Successes

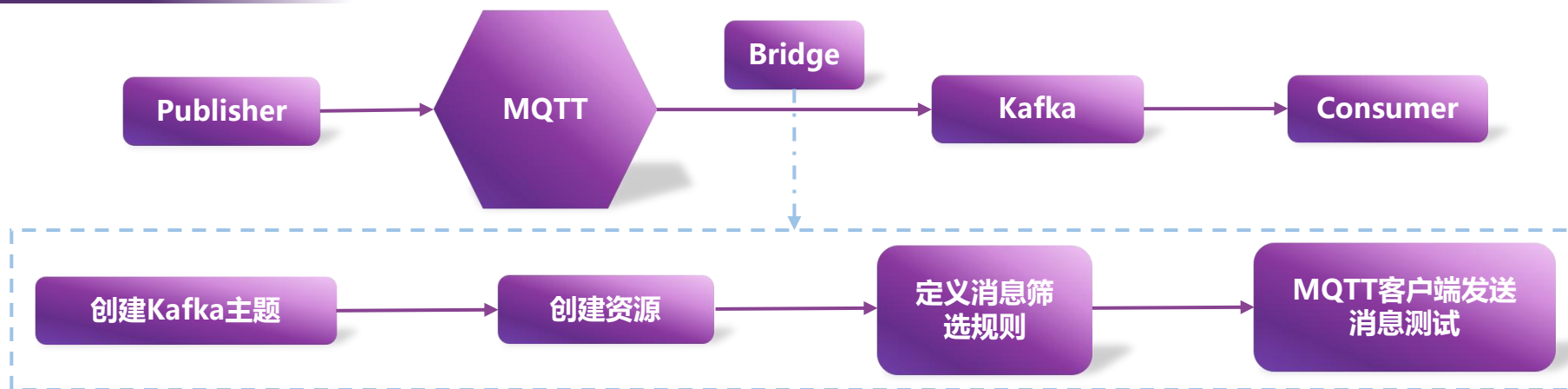
Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB...	Sent KB/sec
MQTT Con...	2000	610	525	842	1149	1534	44	2358	0.00%	113.2/sec	1.22	0.00
MQTT Pub ...	4000	5	5	12	14	19	0	23	0.00%	202.1/sec	3.95	1844.89
MQTT DisC...	2000	352	99	922	1396	2905	16	4630	0.00%	93.6/sec	1.01	0.00
TOTAL	8000	243	21	771	859	1676	0	4630	0.00%	255.0/sec	3.86	1164.27



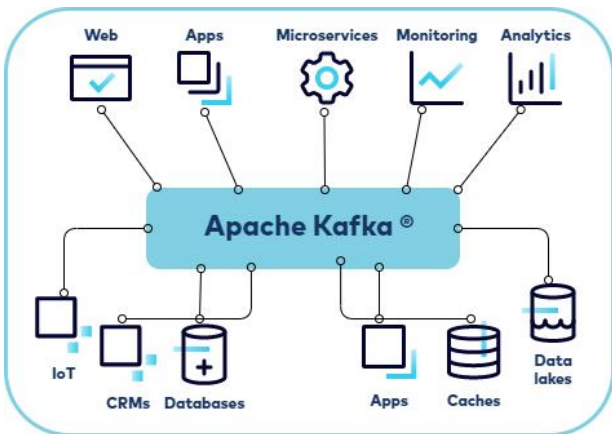
# 车联网边端通信模拟

## MQTT桥接Kafka

## MQTT桥接Kafka



**Kafka:** 一种高吞吐量、持久性、分布式的发布订阅的消息队列系统，主要用于处理消费者规模网站中的所有动作流数据。



**创建MQTT消费者:** 通过使用MQTT客户端库和订阅特定主题 (Topic) 创建一个消费者，用于接收来自MQTT代理的消息。

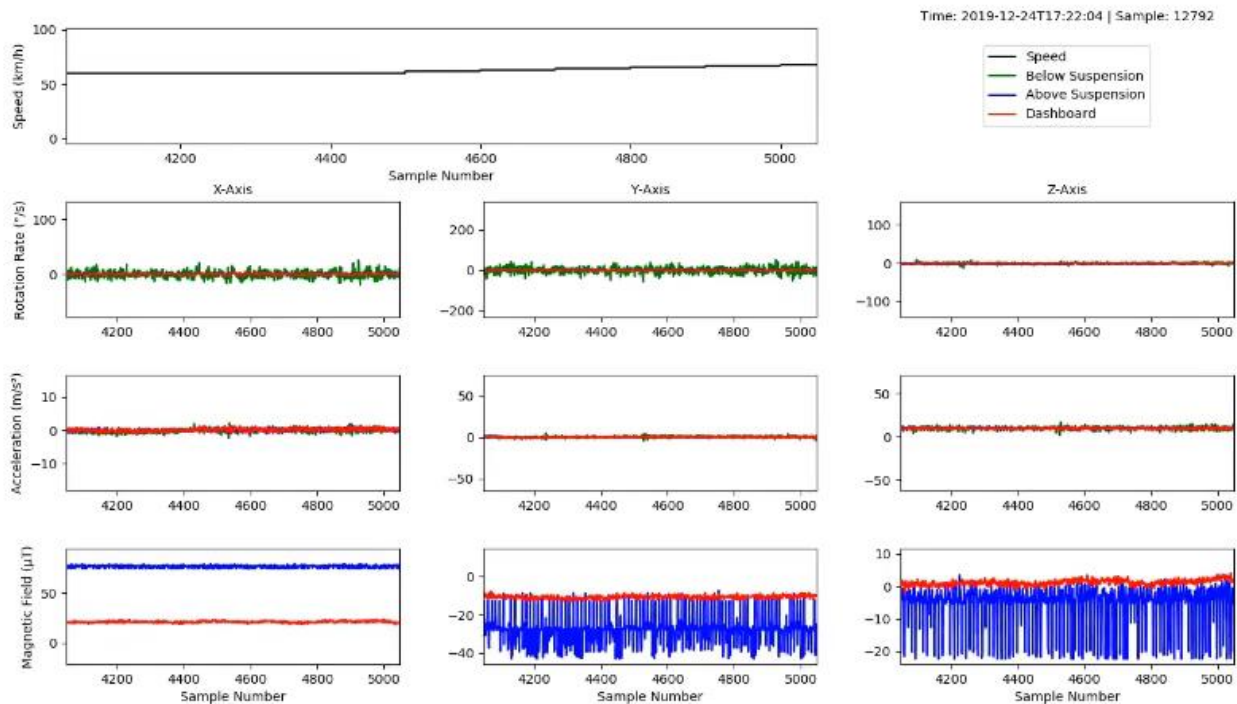
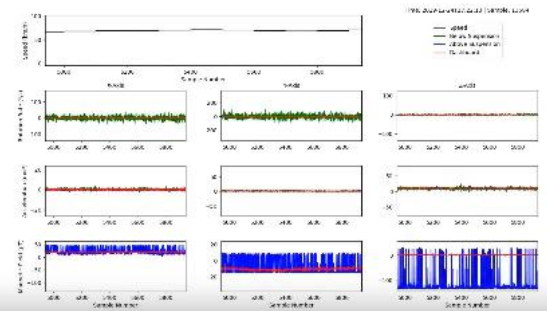
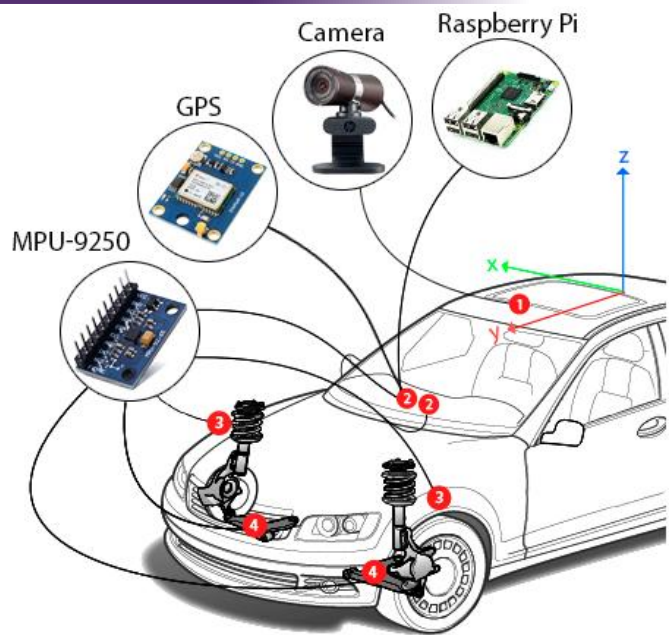
**创建Kafka生产者:** 在Kafka端，创建一个Kafka生产者来将接收到的MQTT消息发送到Kafka集群。

**消息桥接和路由:** 将接收到的MQTT消息通过Kafka生产者发送到Kafka集群中的指定主题，实现消息的桥接和路由。



# 道路质量评估

## 车辆传感器数据



#	文件夹	车辆	驾驶员	场景	距离
#PVS 1		VW Saveiro	Driver 1	Scenario 1	13.81 km
#PVS 2		VW Saveiro	Driver 1	Scenario 2	11.62 km
#PVS 3		VW Saveiro	Driver 1	Scenario 3	10.72 km
#PVS 4		Fiat Bravo	Driver 2	Scenario 1	13.81 km
#PVS 5		Fiat Bravo	Driver 2	Scenario 2	11.63 km
#PVS 6		Fiat Bravo	Driver 2	Scenario 3	10.73 km
#PVS 7		Fiat Palio	Driver 3	Scenario 1	13.78 km
#PVS 8		Fiat Palio	Driver 3	Scenario 2	11.63 km
#PVS 9		Fiat Palio	Driver 3	Scenario 3	10.74 km



# 道路质量评估

## 数据集处理

### 密度聚类算法

数据集

道路分类

**概念:** DBSCAN (Density-based spatial clustering of applications with noise), 基于密度的聚类算法, 通过计算样本点周围的密度来确定样本点所属的簇。

找到核心点

只考虑核心点

划为聚类

遍历非核心点

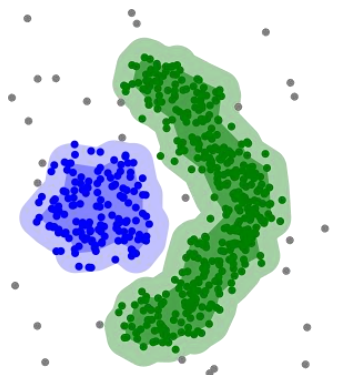
是否在临域内

是

划为聚类

否

划为噪点



**优势:**

1. 相比于K-means, 它不需要指定聚类数量K;
2. DBSCAN可以找出任何形状的聚类;
3. 可以分辨噪点。

**缺点:**

1. 不是完全决定性的: 在两个聚类交界边缘的点属于哪个聚类取决于它在数据库的次序;
2. 两个超参数有时很难确定, 并且最终结果对于超参数非常敏感。



# 道路质量评估

## 数据集处理

数据集

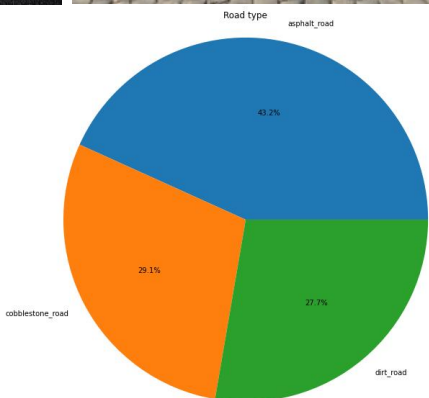
密度聚类算法

道路分类

沥青路面

石砖路面

泥路



### 不同道路类型的道路质量

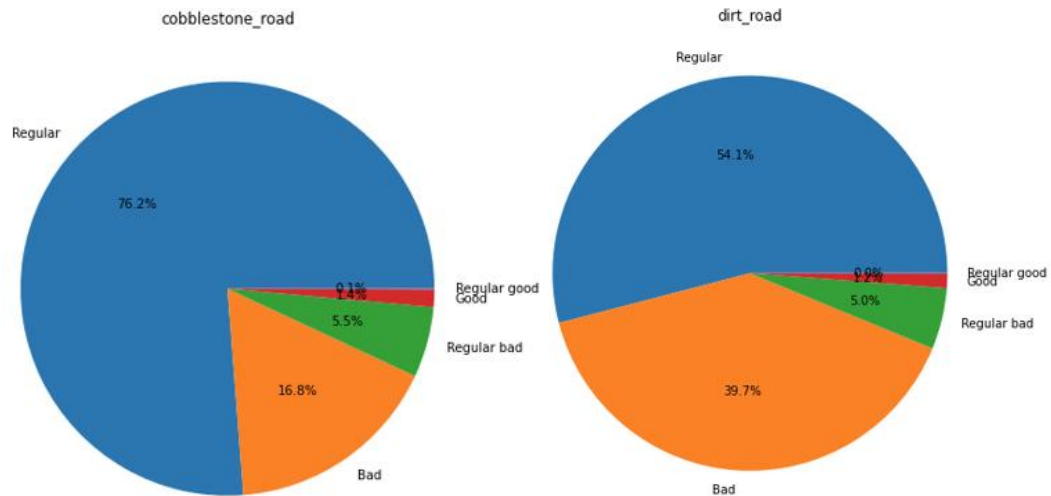
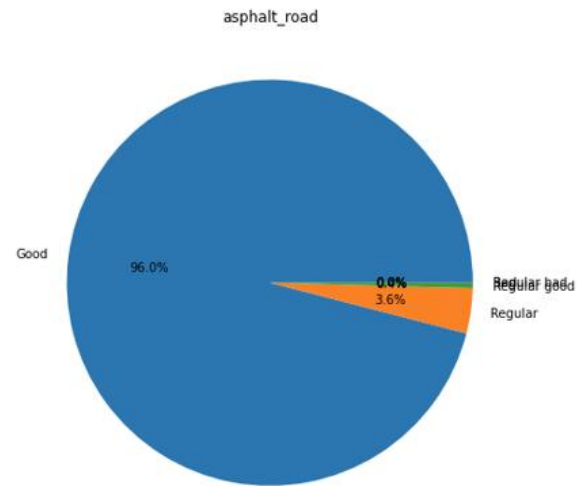
Good

Regular good

Regular

Regular bad

Bad

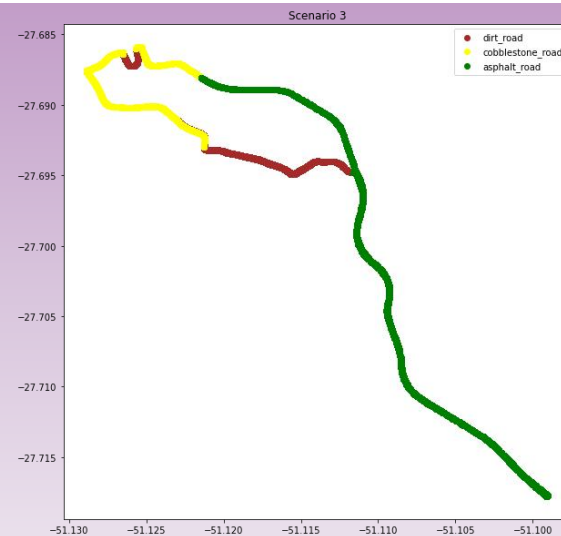
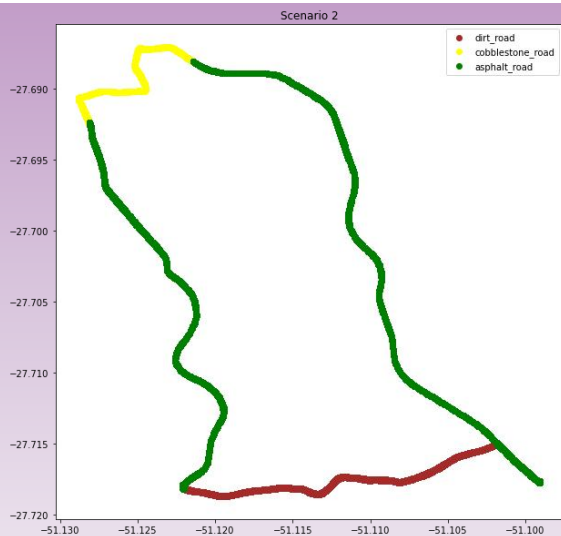
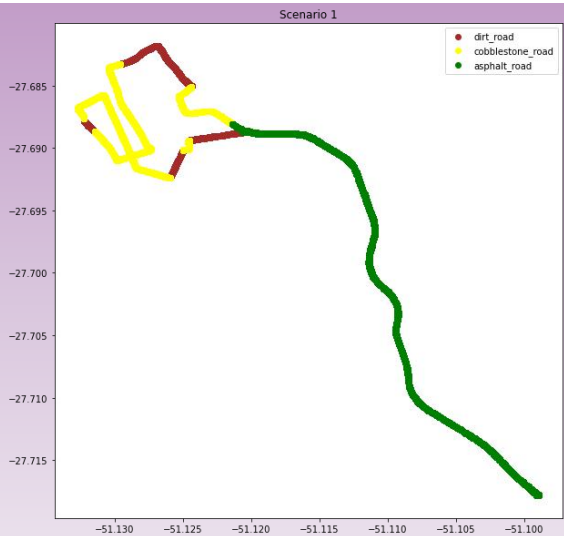




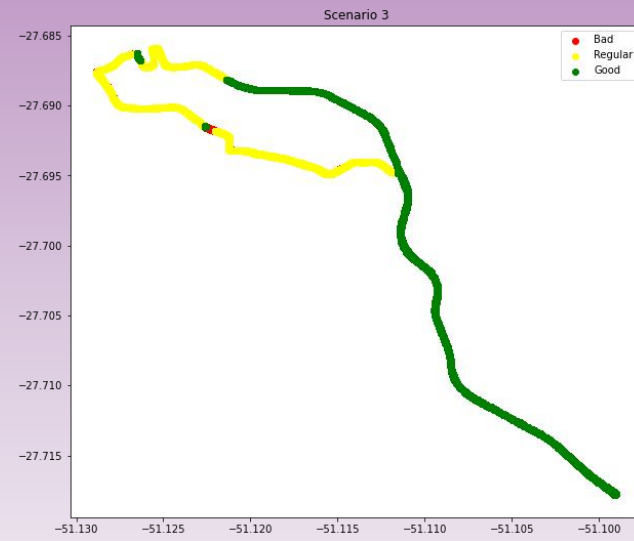
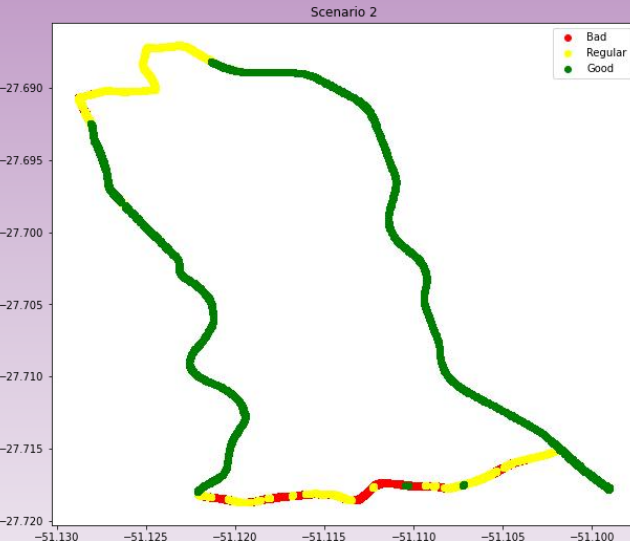
# 道路质量评估

## 道路可视化

道路种类可视化



道路质量可视化

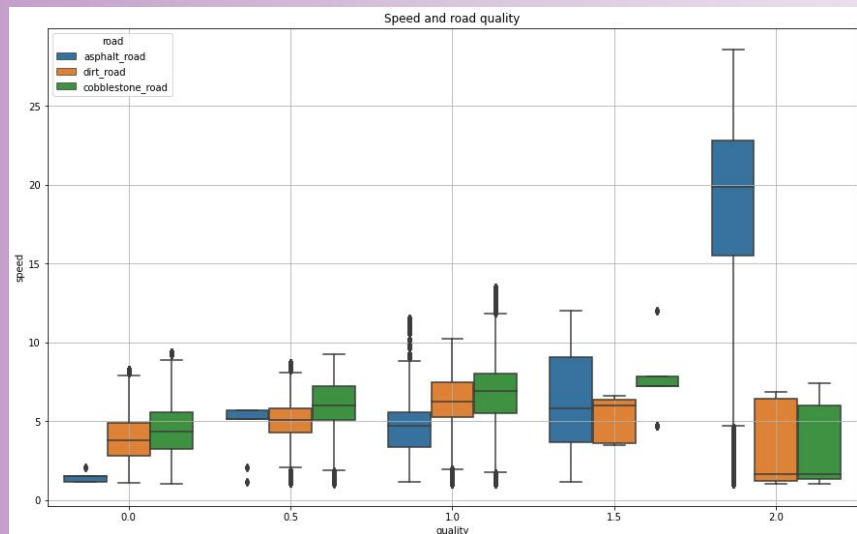




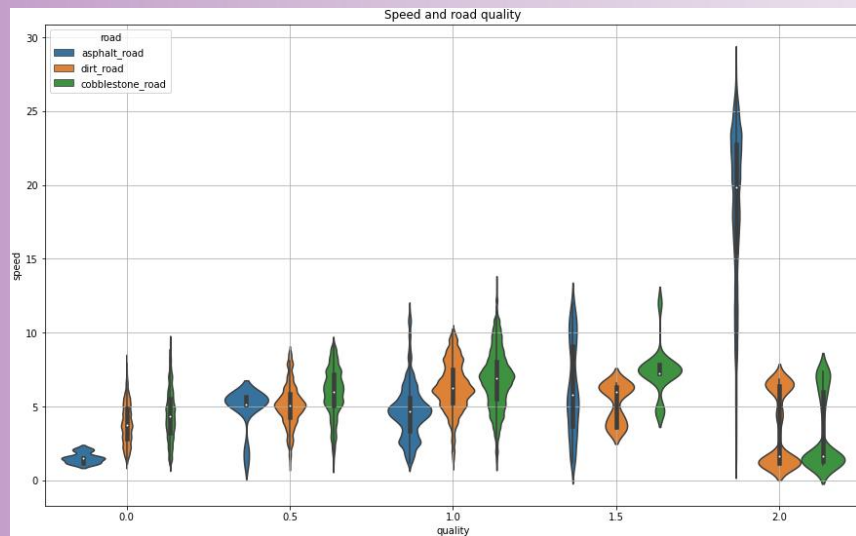
# 道路质量评估

## 速度与道路质量关系探究

绘制不同道路类型下的速度与道路质量之间的盒图和小提琴图。通过观察图表可以观察到：速度在沥青路上更高，对于道路质量更好的石棉路和泥路，速度也更高，与现实情况相符合。



盒图



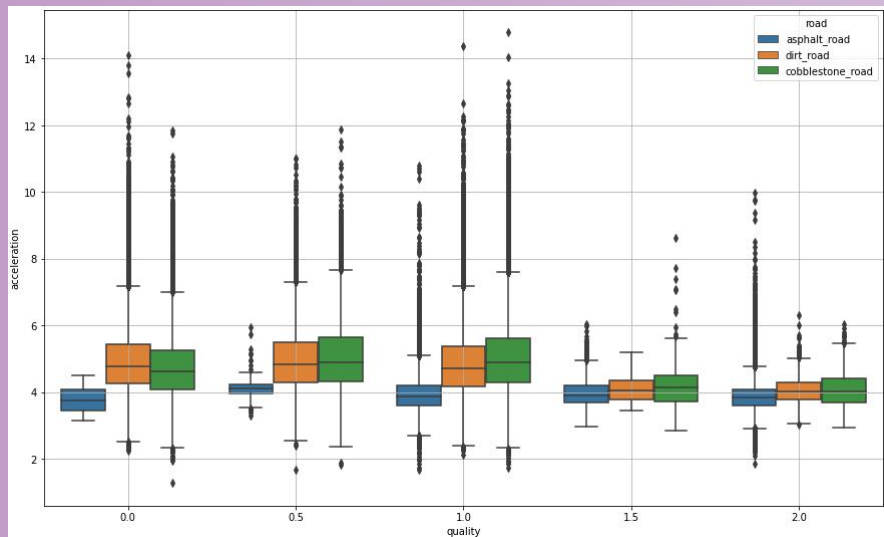
小提琴图



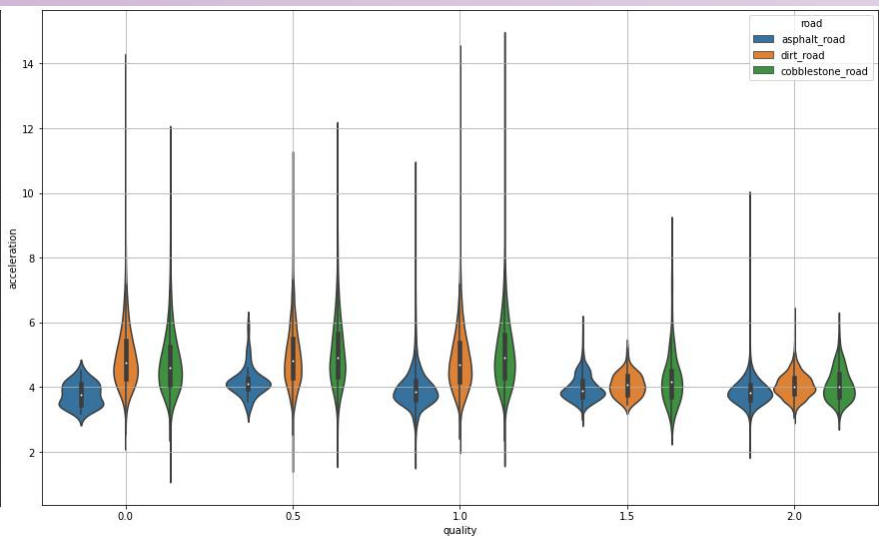
# 道路质量评估

## 加速度与道路质量关系探究

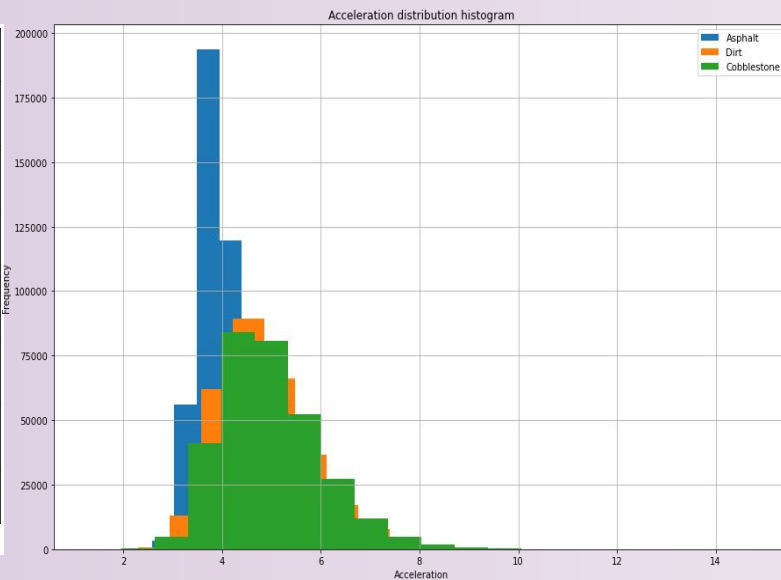
绘制不同道路类型下的加速度与道路质量之间的盒图、小提琴图以及直方图。通过观察图表可以观察到：在鹅卵石和土路上，加速度值的振幅比沥青路大，与现实情况相符合。



盒图



小提琴图



直方图





## 工作总结

- 1.车联网仿真模拟：**使用Veins、SUMO及OMNET++的车联网仿真模拟，模拟事故发生以及实时消息处理。
- 2.边端通信模拟：**使用Python语言构造了一个数据模拟器用来产生汽车传感器数据。使用MQTT作为传输协议，确立EMQ作为MQTT Broker，在EMQ中构建桥接Kafka的插件和规则引擎，实现将数据被预处理后传输给Kafka。
- 3.道路质量评估：**通过**密度聚类**等算法对车联网数据进行分析计算，加入路面种类及质量可视化处理，结合加速度与速度对路面质量进行分析验证模型准确性。

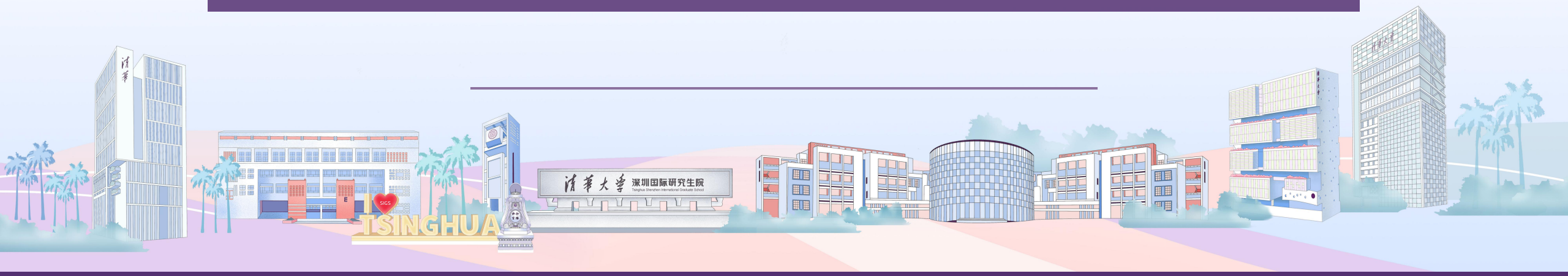
## 项目展望

- 1.车联网仿真模拟：**进一步扩展仿真模型，包括更多车辆行为、路况变化、交通规则等因素，更真实地模拟车联网环境。
- 2.边端通信模拟：**进一步研究和开发边缘通信模拟工具，模拟不同通信协议、网络条件和延迟等因素对系统性能的影响。
- 3.区域细分方面：**可以结合车联网数据和实时传感器信息，为驾驶员和交通管理者提供准确的道路信息和决策支持。



清华大学 深圳国际研究生院  
Tsinghua Shenzhen International Graduate School

请批评指正！



小组成员：陈洋、付磊