Analysis of Human Behaviour in Different Driving Scenarios

Group 1: Aniruddha Chaki, Puya Fard, Ruchi Patel, Sauryadeep Pal, Yasamin Moghaddas

Topics for Discussion

- **Introduction**
  - Problem Statement
  - Goal
- **Methodology**
  - Hardware setup
  - Driving Scenarios
  - Study conducted
  - Data collection methods
  - Case study methods
- **Findings**
  - Data collection
  - Data clustering
  - Case study findings
- **Conclusion**
- **Challenges encountered**
- **Github & Website**
- **Reference**

Introduction

- Road accidents are a major health concern globally.
- Understanding the complex human behaviour in traffic situations is crucial for enhancing safety.

Problem statement

- Create and simulate controlled traffic scenarios using CARLA simulator environment to study driver's behaviour

Goal

- Research and implement traffic scenarios on CARLA simulator.
- Gather data on human biometrics, such as heart rate, along with vehicular metrics like speed and acceleration, to analyze and classify driving behavior.
- Case Study on driving behaviour across various traffic conditions.

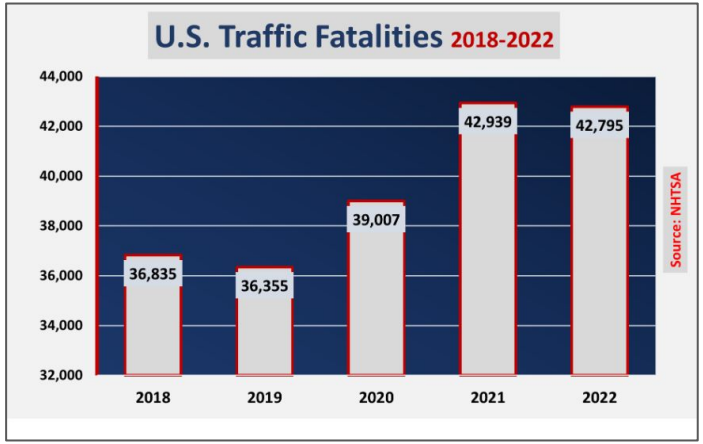


Fig 1: Accidents

https://maisonlaw.com/2023/04/2023-california-car-accident-statistics/ - Source : National Highway Traffic Safety Administration (NHTSA)

Hardware Setup

- G920 Driving wheel
- GTrack Driving seat
- Logitech Driving pedals
- 50’ TV
- PC installed with CarlaUE4
- BioHarness Zephyr Belt

Figure 2: Hardware setup

Methodology

Driving scenarios

- Default Environment
- Driving at Night
- Surrounded Vehicles Overspeeding
- Surrounded Distracted Vehicles
- Congested Driving

Data collection parameters

- Vehicular Data collection
  - Driver Speed timestamps
  - Collison
- Biometric Data Collection
  - Heart rate
  - Breathing rate

Figure 3: CARLA

Methodology

Study conducted

- Total of 10 test drives volunteered by UCI students.
- Age of the drivers was between 21-26 with no known heart conditions.
- Each scenario to be tested for 10 minutes.

Case study & analysis

- Collected data from vehicles and the biometric belt.
- The data was then analysed using clustering algorithms to draw conclusions on human driving behaviour

Figure 4: Zephyr Belt

Code - Aggressive Behavior for Driving Scenarios

- Aggression is defined by:
  - Forceful lane changes
  - Ignoring red lights
  - Ignoring stop signs
  - Overspeeding
- The type of behaviors applied depends on the scenario
- Each car has a 50% chance to have the applied behaviors

```
if scenario == Scenario.overspeeding:
    traffic_gen.set_aggressive_behavior_all(
        en_ignore_light=False,
        en_ignore_signs=False,
        en_overspeed=True,
        en_lane_change=True
    )
elif scenario == Scenario.distracted:
    traffic_gen.set_aggressive_behavior_all(
        en_ignore_light=True,
        en_ignore_signs=True,
        en_overspeed=False,
        en_lane_change=False
    )
```

Code 1: Setting aggressive behavior

Code - Traffic Congestion

- Traffic congestion is simulated by making cars follow specific routes.
- The simulation keeps spawning more cars over time to simulate a gradual traffic buildup.

```python
def spawn_congestion_vehicle(self):
    n_congestion_vehicles = len(self.world.get_actors().filter('*vehicle*'))
    if self.counter == 0 and n_congestion_vehicles < 200:
        spawn_point = self.spawn_points[32] if self.alt else self.spawn_points[149]
        vehicle = self.world.try_spawn_actor(random.choice(self.vehicle_blueprints), spawn_point)
        if vehicle:
            vehicle.set_autopilot(True)
            self.traffic_manager.auto_lane_change(vehicle, True)
            if self.args.aggression:
                self.set_aggressive_behavior(vehicle, True, True, True, True)
            if not self.args.disable_car_lights:
                self.traffic_manager.update_vehicle_lights(vehicle, True)
            path = self.route_1 if self.alt else self.route_2
            self.traffic_manager.set_path(vehicle, path)
            self.alt = not self.alt
            self.vehicles_list.append(vehicle.id)
            logging.info('Spawned congestion vehicle (current: %d vehicles)', n_congestion_vehicles)
            return vehicle
        self.counter = self.spawn_delay
    else:
        self.counter -= 1 if self.counter > 0 else 0
    return None
```

Code 2: Generating traffic congestion

Code - Control Inputs

- This simulation accepts control inputs from the user and applies them to the Ego vehicle in CARLA.
- Control inputs are parsed and applied every server tick in the simulation

```python
def _parse_vehicle_wheel(self):
    """Parse inputs from the steering wheel and pedals."""
    numAxes = self._joystick.get_numaxes()
    jsInputs = [float(self._joystick.get_axis(i)) for i in range(numAxes)]
    # print (jsInputs)
    jsButtons = [float(self._joystick.get_button(i)) for i in
                 range(self._joystick.get_numbuttons())]

    # Custom function to map range of inputs [1, -1] to outputs [0, 1]
    # For the steering, it seems fine as it is
    K1 = 1.0  # 0.55
    steerCmd = K1 * math.tan(1.1 * jsInputs[self._steer_idx])

    K2 = 1.6  # 1.6
    throttleCmd = K2 + (2.05 * math.log10(
        -0.7 * jsInputs[self._throttle_idx] + 1.4) - 1.2) / 0.92
    if throttleCmd <= 0:
        throttleCmd = 0
    elif throttleCmd > 1:
        throttleCmd = 1

    brakeCmd = 1.6 + (2.05 * math.log10(
        -0.7 * jsInputs[self._brake_idx] + 1.4) - 1.2) / 0.92
    if brakeCmd <= 0:
        brakeCmd = 0
    elif brakeCmd > 1:
        brakeCmd = 1

    self._control.steer = steerCmd
    self._control.brake = brakeCmd
    self._control.throttle = throttleCmd
```

Code 3: Parsing control inputs

Data Collection

Logging vehicle and biometrics data

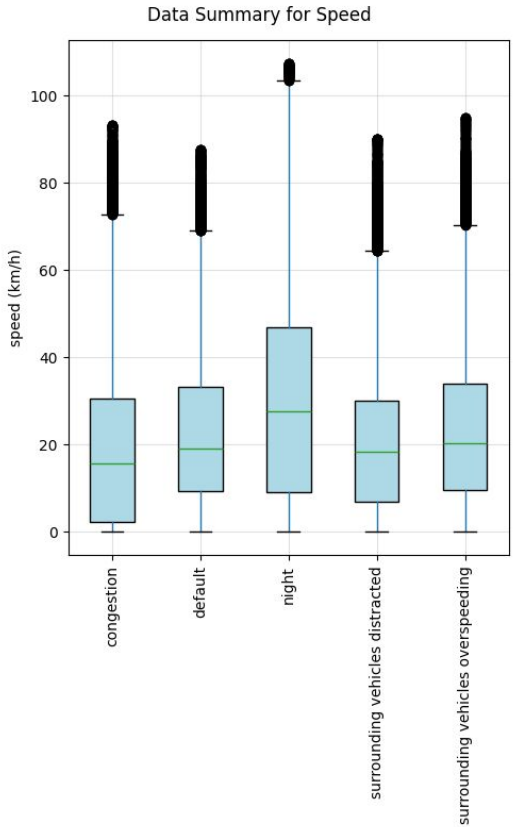
- Data collection has been completed on 10 drivers, with every driver conducting each scenario once.
- Data has been collected in form of .csv file listing parameters such as:
  - Timestamp
  - Vehicle speed
  - Heart rate
  - Breathing rate
- Biometrics data is sent to the simulation server by the zephyr belt at a 1 Hz frequency.
- Data is collected by the simulation server, which logs vehicular and biometrics data every server tick.
- Logging per tick helps prevent biometrics data loss as there are multiple server ticks in one real-time second.

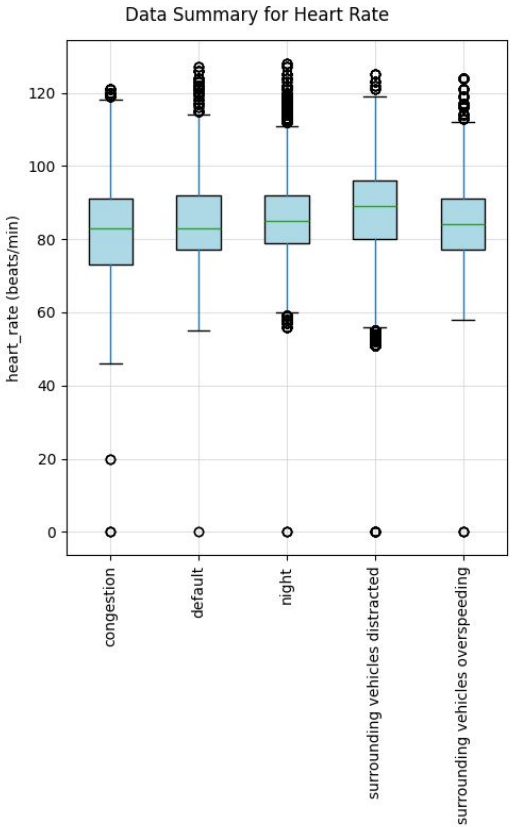
```
with open(self.speed_log, 'a', encoding='utf-8') as speed_f:
    speed_f.write(f'{timestamp.seconds},{timestamp},{speed:.2f},{c.throttle:.2f},{c.brake:.2f},{c.steer:.2f},{self.heart_rate},{self.breathing_rate}\n')
```

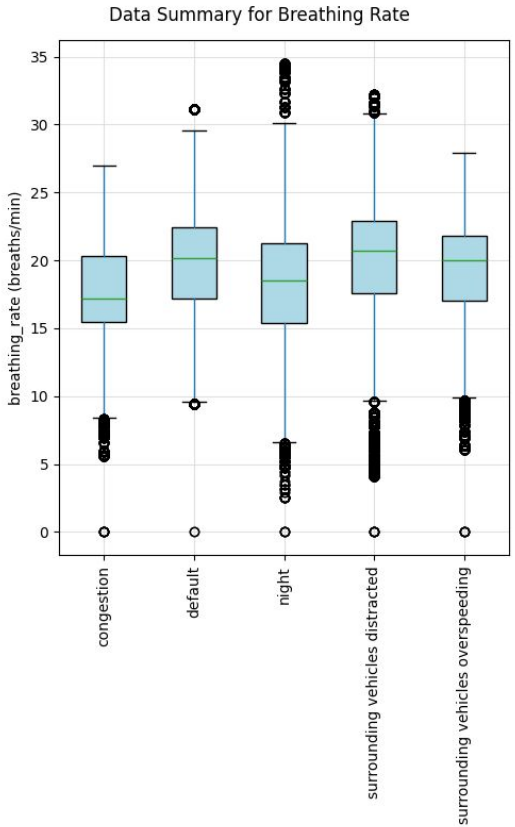
Code 4: Data Collection

Data Summary

Summary of all participant data across all scenarios







Data Collection

Filtering duplicate data

- Collected data has been filtered to remove redundant data caused by CARLA's timekeeping.
- CARLA records data per tick and each second in real-time can have a variable number of ticks. This results in a lot of redundant data.
- We remove duplicate data which is defined as multiple rows of identical data across all columns.

```python
def remove_duplicates_multi(logs_dir):
    """Remove duplicates from all log files in a directory"""

    output_dir = Path(f'{logs_dir}/filtered')
    output_dir.mkdir(parents=True, exist_ok=True)

    for f in Path(logs_dir).glob('*.csv'):
        df = pd.read_csv(f)
        df_new = df.drop_duplicates(keep='first')
        if args.interpolate:
            try:
                interpolate(df_new, 'heart_rate')
                interpolate(df_new, 'breathing_rate')
            except KeyError as e:
                print(f'Error for log "{f}": {e.args[0]}')
        df_new.to_csv(f'{output_dir}/{f.stem}_filtered.csv', index=False)
```

Code 5: Data Filtering

Data Summary

Data clustering

- For behavioral analysis, the filtered data was clustered using a combination of:
  - Vehicle speed
  - Heart-rate
  - Breathing-rate

  Each data point was a 3 element vector [speed, heart_rate, breathing_rate]
  This data was normalized with Z-score normalization.

- We evaluated two clustering algorithms:
  - KMeans++
  - (Density Based SCANning) DBSCAN

Data Summary

Data clustering

- DBSCAN was used for clustering
  - DBSCAN is robust to noise and doesn't require a predefined number of clusters.
  - KMeans++ was also attempted but silhouette analysis consistently favored high cluster numbers (>7) which didn't seem realistic for a dataset consisting of data from only 10 people.

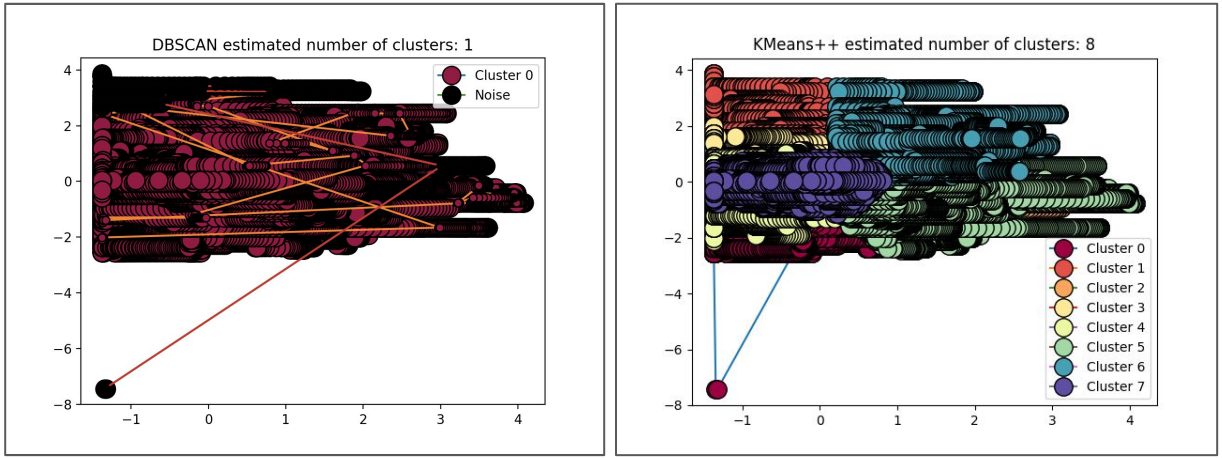
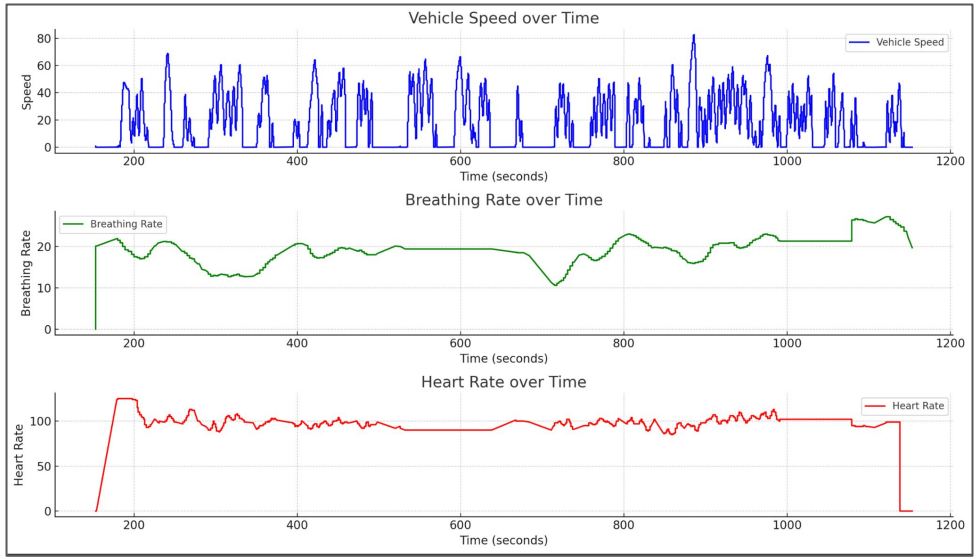


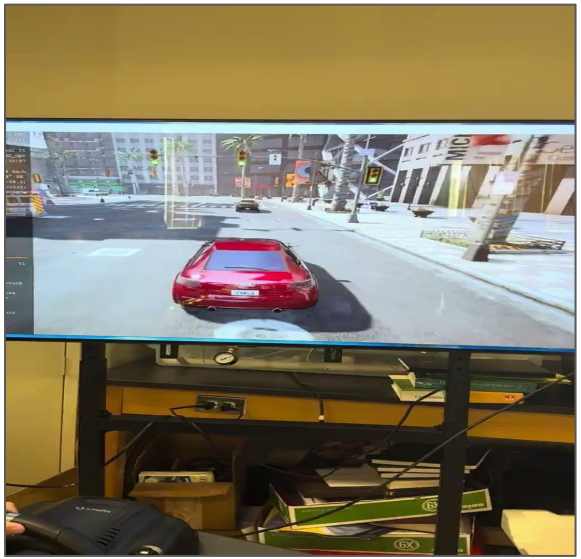
Figure 5: Data Clustering with DBSCAN vs KMeans++ on the same dataset

Data collection - Default environment

Parameters

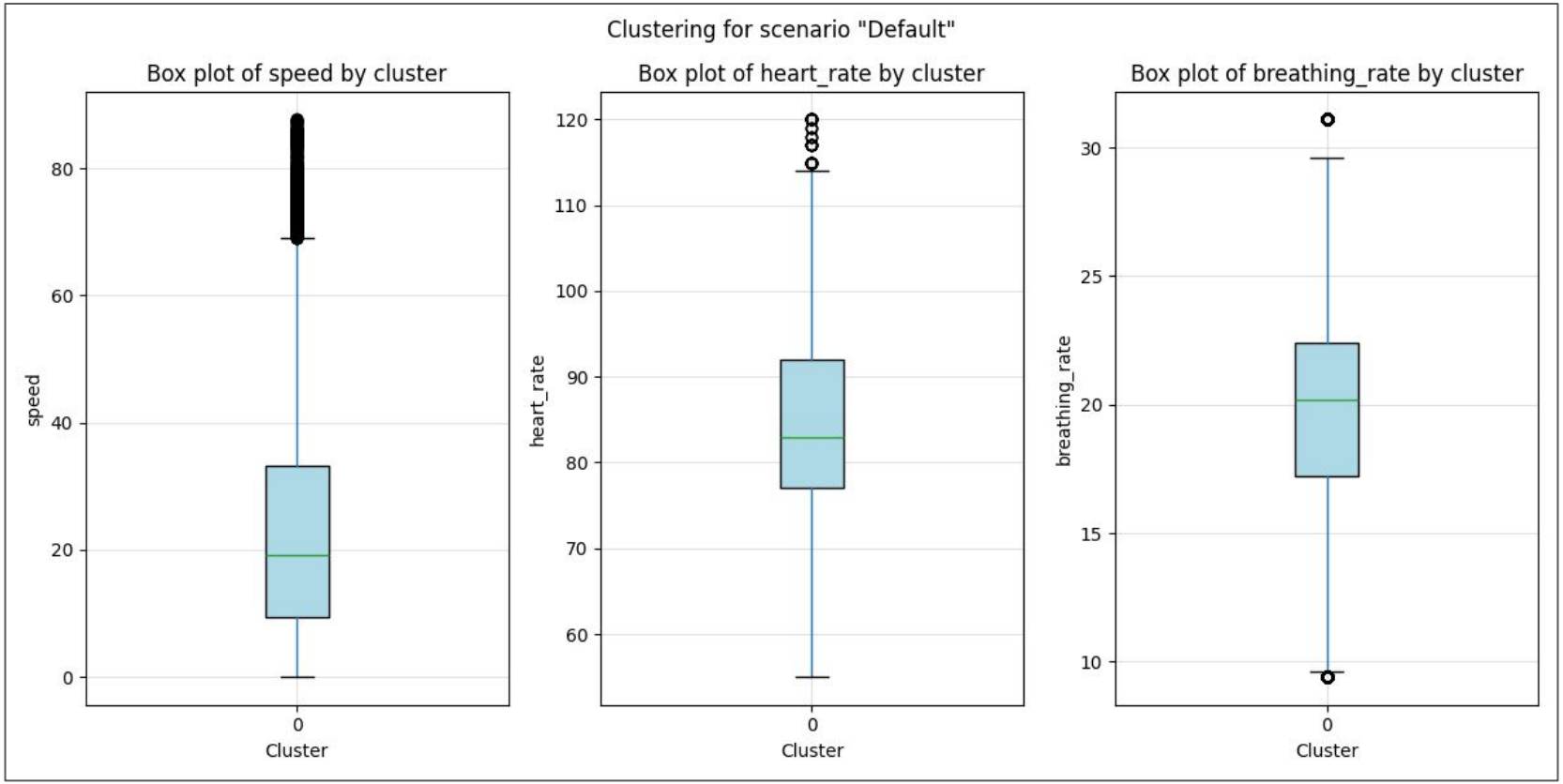
- Clear sunny weather
- Default traffic
- Default driving behavior.





Video 1: Default scenario

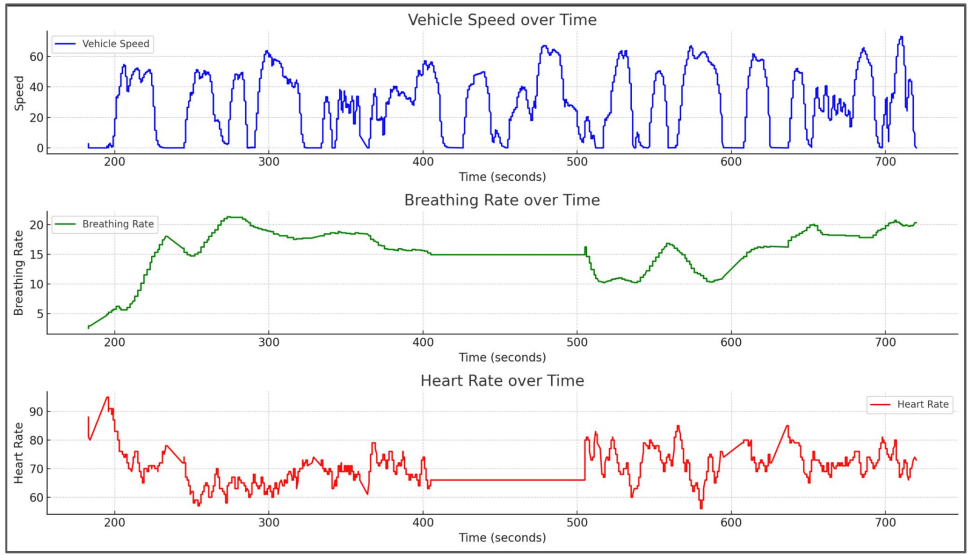
Data clustering - Default environment

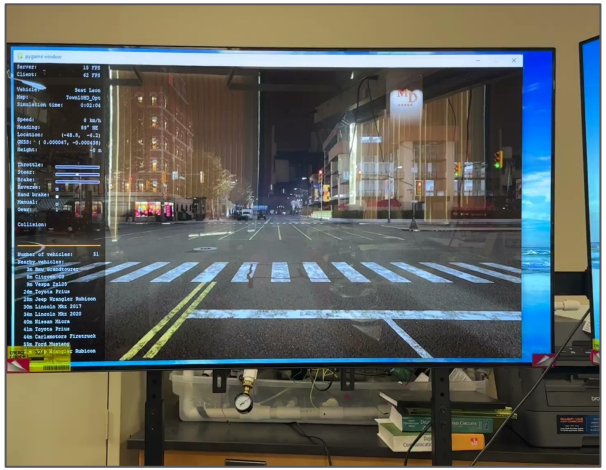


Data collection - Driving at night

Parameters

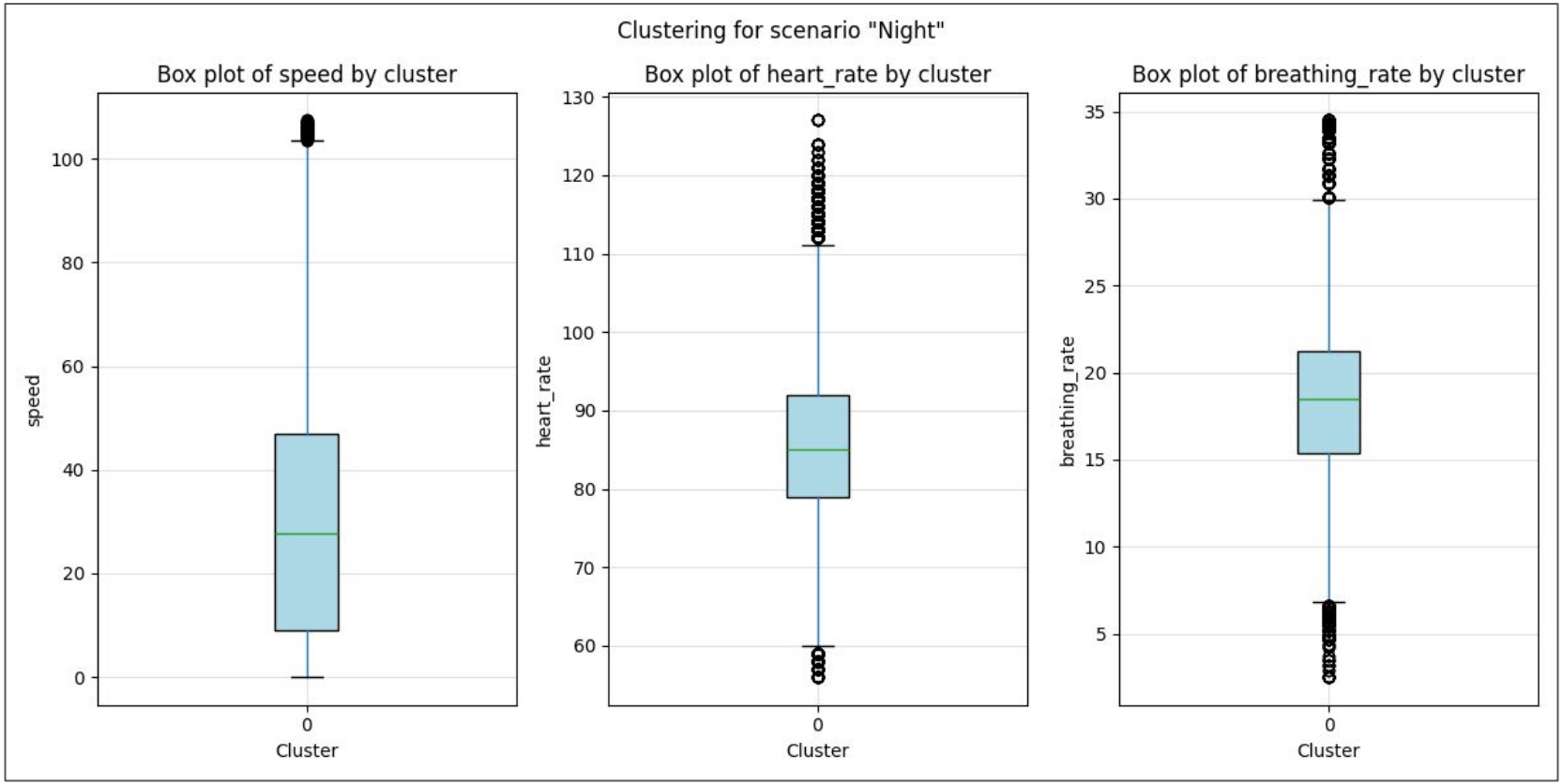
- Clear night weather
- Default driving behavior





Video 2: Night scenario

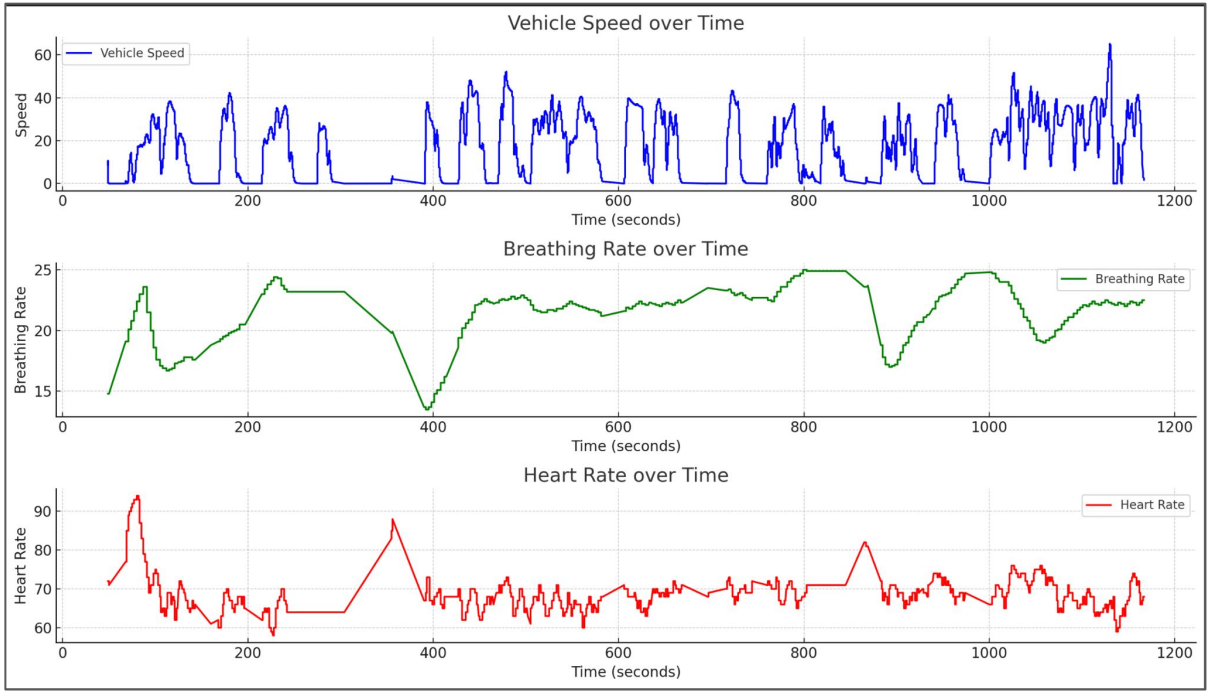
Data clustering - Driving at night



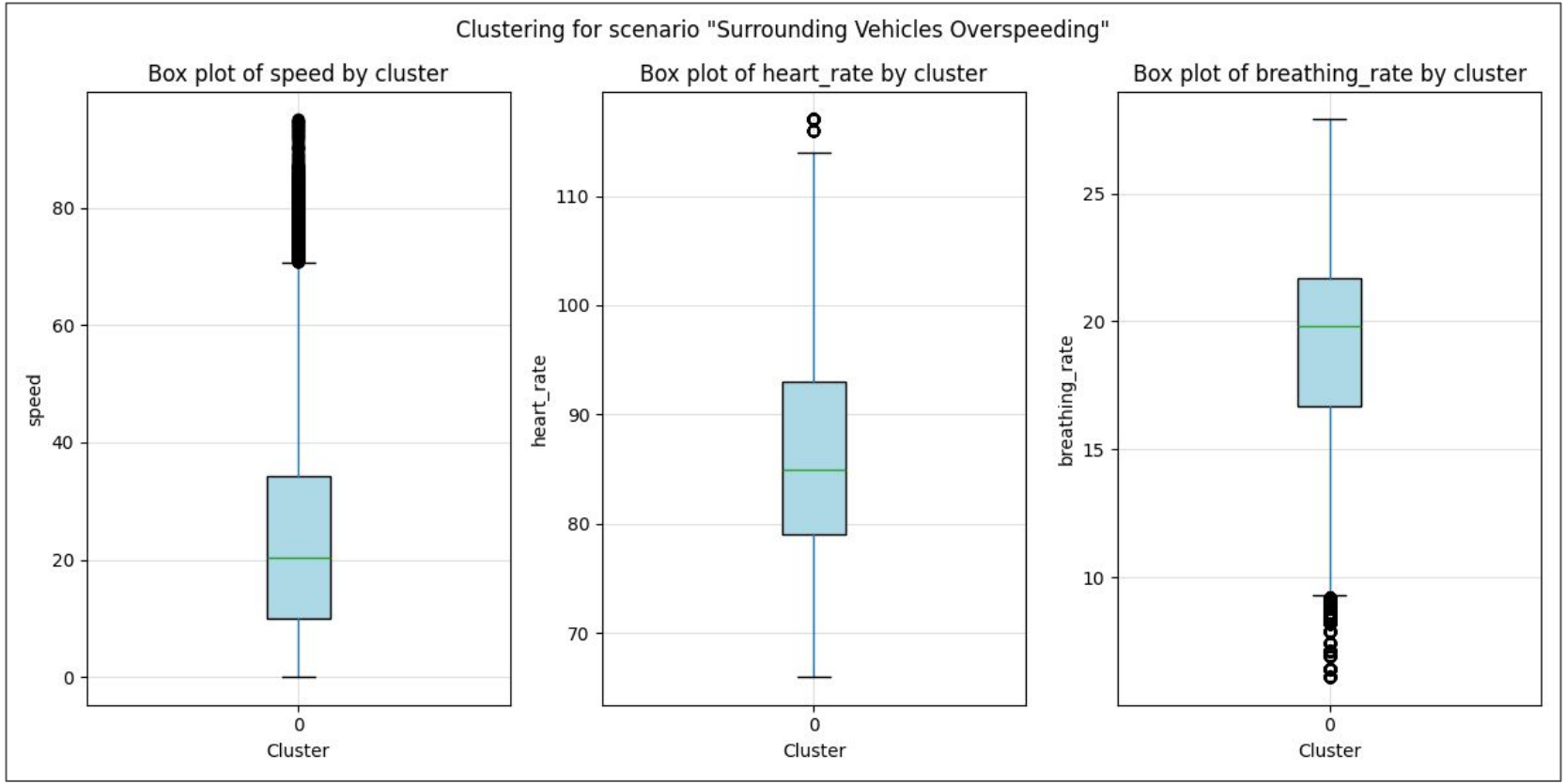
Data collection - Surrounding vehicles overspeeding

Parameters

- Clear sunny weather
- Modified traffic
  - Increased speed
  - Increased lane change freq



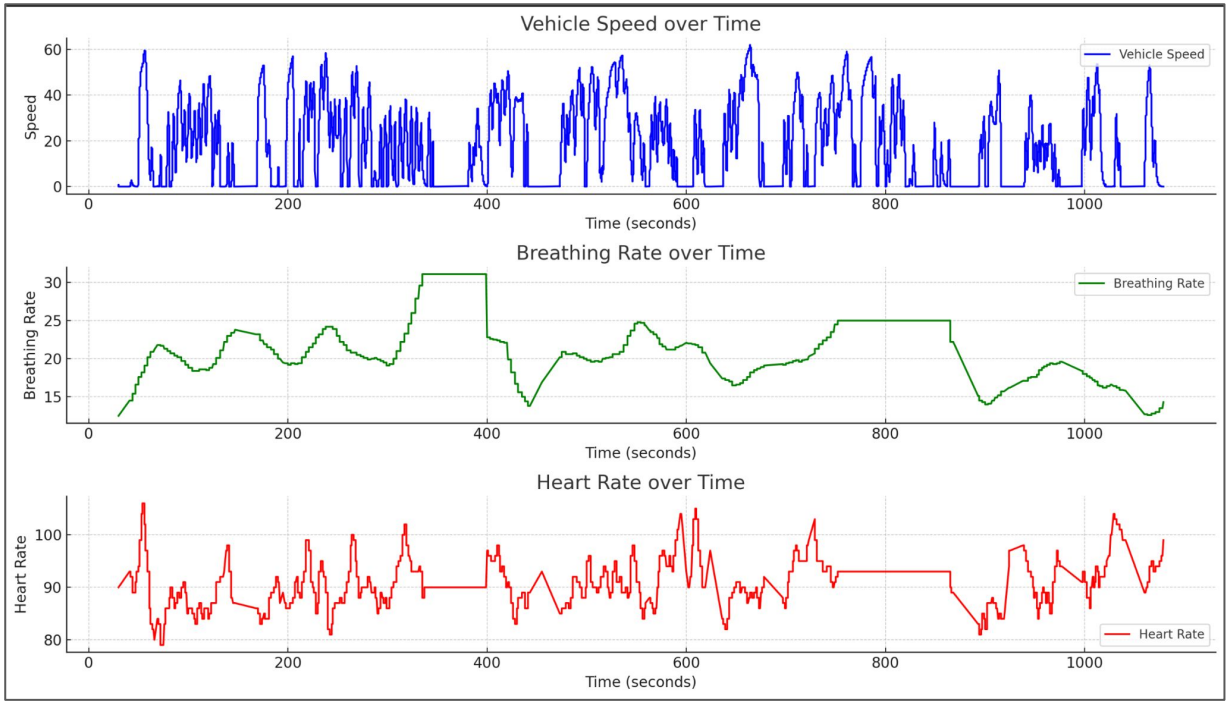
Data clustering - Surrounding vehicles overspeeding



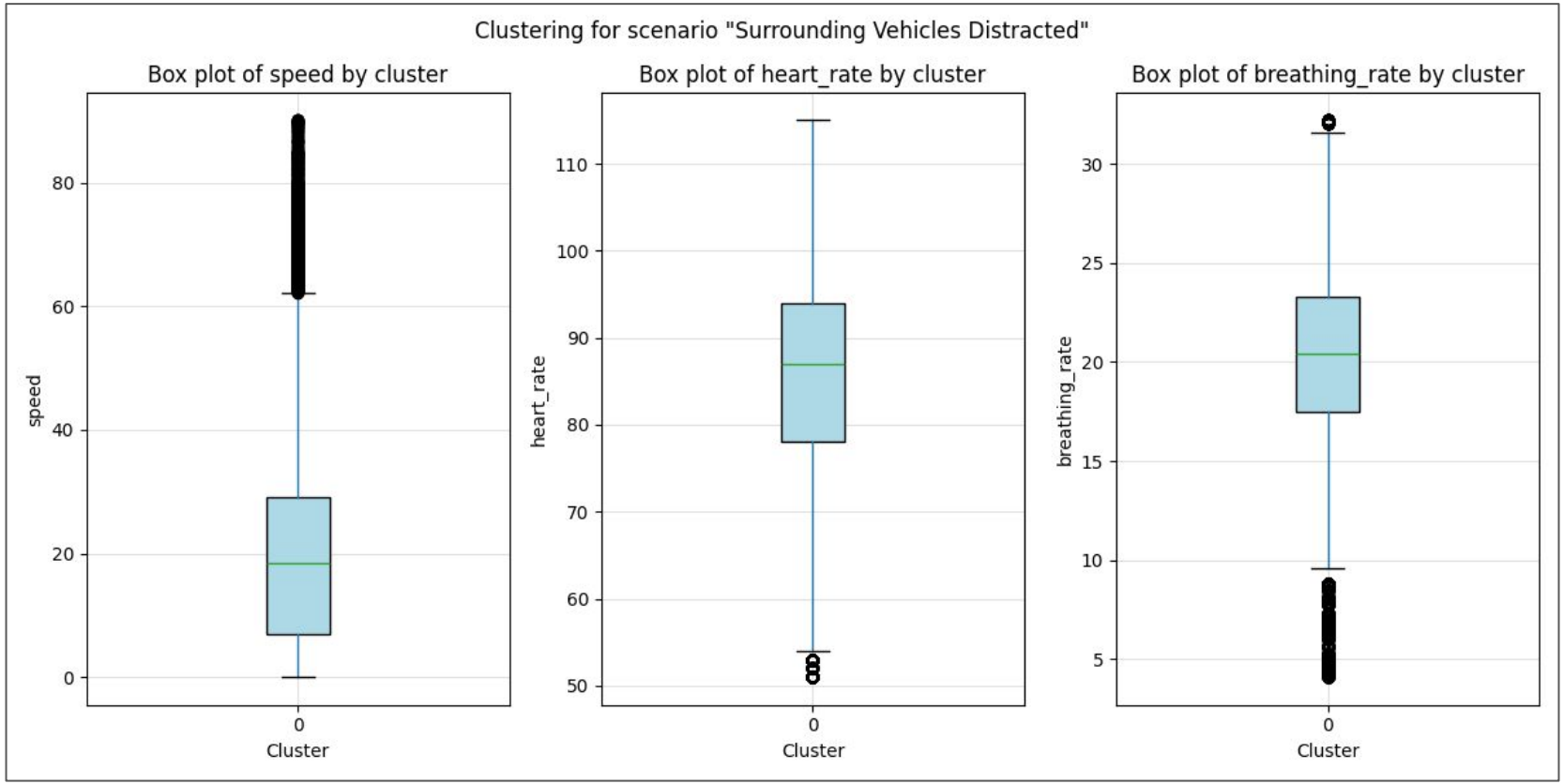
Data collection - Surrounding vehicles distracted

Parameters

- Clear sunny weather
- Modified traffic
  - Ignoring Traffic Lights
  - Ignoring Stop Signs



Data clustering - Surrounding vehicles distracted

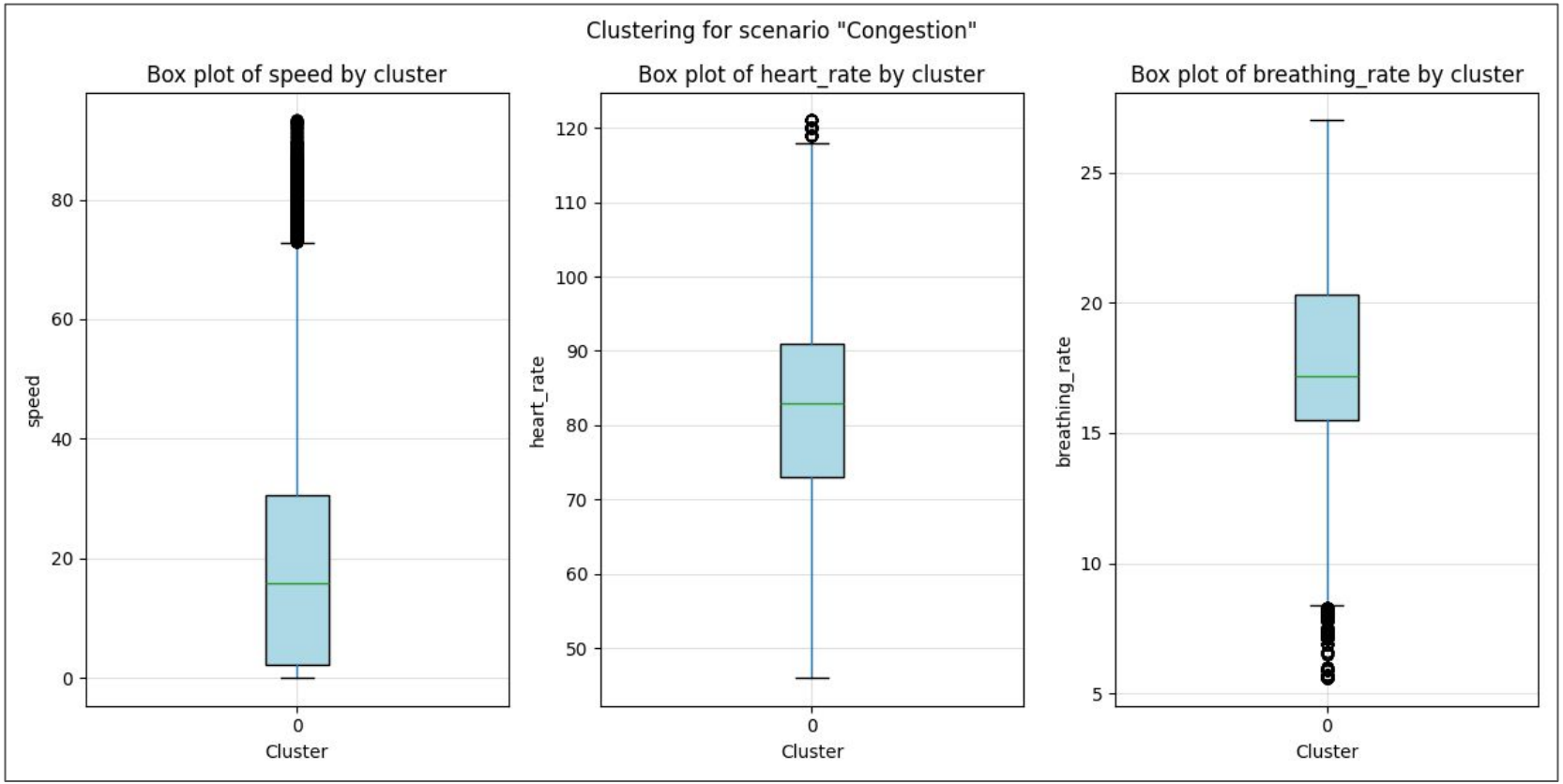


Data collection - Congested driving

Parameters

- Clear sunny weather
- Modified traffic
  - Increased number of cars
  - Congested roads
- Modified driving behavior
  - Frequent lane changing vehicles

Data clustering - Congested driving



Case Study - Common patterns

Common Patterns

Default vs Surrounding vehicles overspeeding driving scenario

- Default scenario outputs average of **82 bpm**, surrounding vehicles overspeeding scenario average of outputs **86 bpm.**

Default vs Surrounding vehicles distracted driving scenario

- Default scenario outputs average of **82 bpm**, surrounding vehicles distracted scenario outputs average of **88 bpm.**

Default vs Congestion driving scenario

- Congestion driving tends to have a lower average vehicle speed of **17 km/h** compared to **20 km/h.**

Default vs Night driving

- Night driving scenarios tend to cause drivers to overspeed based on the output of the clustered data by over **40%.**
- Tends to have higher average heart rate 85 bpm and lower average breathing rate **17 breath per min** compared to **22 breaths per minutes** on default scenario.

Case Study - Data That Stands Out

- The default heart rate is steady, while surrounding vehicles distracted and overspeeding scenarios show spikes in heart rate due to driver stress and anxiety, raising the average from 82 bpm to 86 bpm and 88 bpm, respectively.

- Night driving leads to speeds over 40% faster than in default conditions associated with lower breathing rates.

Conclusion

- Speeds are lower in congestion and higher speeds when surrounding vehicles are overspeeding.

- Heart rate and breathing rate are less sensitive to changes in driving conditions.

- Overall, no particular association was found between driving conditions and changes in drivers' physiological responses, since, while there were changes, they were not noteworthy so it is difficult to draw a general conclusion.

Challenges

- Simulating real-world driving environment for the participants is challenging
- Drawing conclusions from the limited dataset of participants
- Using the BioHarness Belt for biometric was challenging

Future Work

- Conduct research in large scale with participants from different background
- Study on more driving scenarios to mimic the real world driving environment
- Implement a Virtual Environment for Immersive experience for the research participants

Github & Website

- Github: Saurya0401/CBSvC: Crowd Behavior Study via CARLA (github.com)
- Website: Crowd Behavior Study via CARLA | CBSvC (saurya0401.github.io)

References

1. Guofa Li, Weijian Lai, Xiaoxuan Sui, Xiaohang Li, Xingda Qu, Tingru Zhang, Yuezhi Li, Influence of traffic congestion on driver behavior in post-congestion driving, Accident Analysis & Prevention, Volume 141, 2020, 105508, ISSN 0001-4575, https://doi.org/10.1016/j.aap.2020.105508.
2. Choi, Eun-Ha. Crash factors in intersection-related crashes an on-scene perspective. [2010]. Retrieved from the Digital Public Library of America,
3. Team, G. (2022, January 7). Causes of aggressive driving and how to solve it. Geotab. https://www.geotab.com/blog/causes-of-aggressive-driving/
4. Team, G. (2023, December 28). The psychology of driving. Geotab. https://www.geotab.com/blog/psychology-of-driving/