The Effectiveness of Artificial Intelligence in Traffic Optimization

Sahand Khademi & Madhav Roshan Babbar

Vienna International School

Vienna International Science & Engineering Fair 2025

Abstract

This study investigates how artificial intelligence (AI) can enhance traffic system efficiency. A virtual traffic simulation was conducted to compare AI-driven traffic management systems with traditional models, measuring key performance indicators (KPIs) such as waiting times, emergency vehicle response times, fuel consumption, and emissions. Simulation results revealed notable improvements across all areas that included a 73.9% decrease in overall waiting time, a 61.7% decrease in emergency vehicle waiting times, an overall decrease of 41.4% in fuel consumption, and a reduction in emissions produced of approximately 58.9%. These findings suggest that AI-based traffic systems can vastly enhance urban mobility, mitigate environmental damage, and shorten emergency response times. These substantial improvements point to the potential of AI to optimize traffic flow in real-world applications.

Introduction

Every day, millions of drivers sit in traffic, wasting fuel, increasing emissions, and losing valuable time. Each year, the average driver in London spends approximately 101 hours alone trapped in traffic due to congested roads , wasting 116 liters of fuel and costing about, wasting an estimated 116 liters of fuel and costing about €1,793 (£1512) in lost time and productivity (INRIX, 2024). Traditional traffic lights operate at fixed intervals, failing to adapt to real-time traffic conditions. With the growth of cities and increased vehicle usage, these systems often struggle to manage fluctuating traffic demand effectively, leading to longer travel times, increased fuel usage, and greater environmental pollution. A study of European cities revealed that road traffic pollution costs an estimated €166 billion annually in lost well-being, with each city suffering an average of €385 million in damages per year (EPHA, 2020). These damages include direct healthcare expenses (e.g., hospital admissions for respiratory and cardiovascular diseases) and indirect societal costs such as reduced productivity due to illness, premature mortality, and diminished quality of life from chronic conditions like asthma. The environmental impact is equally concerning, with traffic in Paris producing 21 megatonnes of CO_2 each year, of which 2.8 megatonnes (13%) is directly attributed to congestion (Beedham, 2022).

However, artificial intelligence presents the opportunity to optimize traffic management through the dynamic adjustment of signal timings according to real-time data gathered from various sources such as cameras, sensors, and Internet of Things (IoT) devices. Using data analysis, AI-based traffic systems can adjust traffic light timing cycles based on real-time vehicle density, predict potential congestion, and maintain a smoother traffic flow throughout the day.

This study seeks to answer the question: ***To what extent can AI optimize traffic systems compared to traditional traffic lights?***, focusing on its impact on waiting times, fuel consumption, emissions, and emergency vehicle response time.

Hypothesis

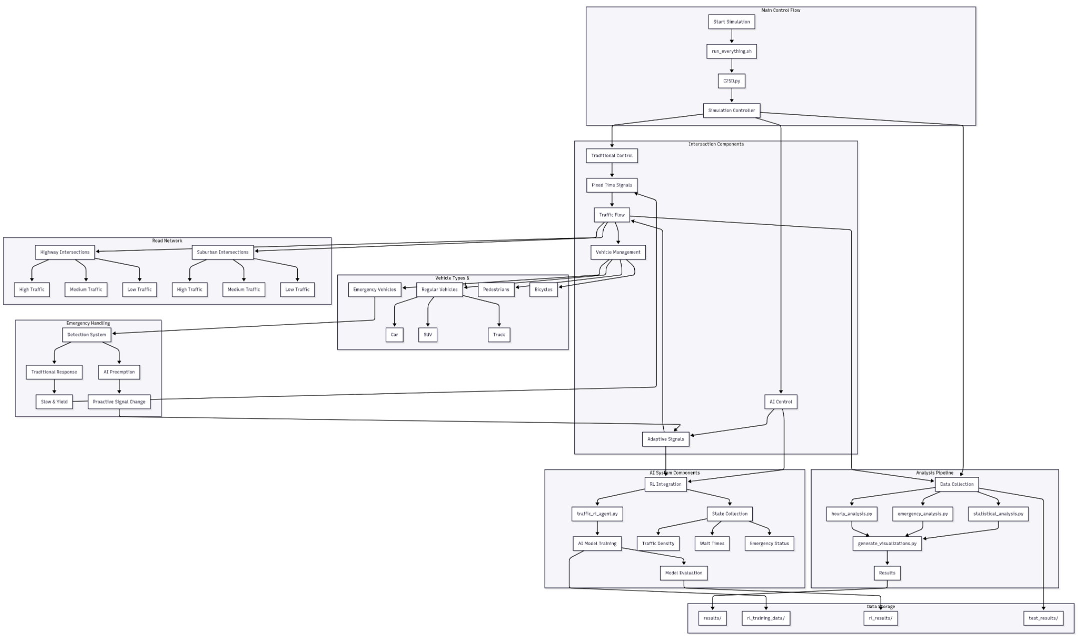
This study hypothesizes that AI-integrated traffic lighting systems will outperform the traditional interval-based system, showing significant improvements in several key areas. First, AI is expected to reduce waiting times by adjusting the traffic light timing according to real-time vehicle density. Second, through the prioritization of emergency vehicles within intersections, it is anticipated to reduce the time for such vehicles' responsiveness. Third, the application of AI should lead to a reduction in fuel consumption, as it can minimize stop-and-go traffic, which accounts for the largest share of fuel wastage. Lastly, through traffic stream optimization and minimizing the amount of time that vehicles spend at a standstill, AI-based traffic systems are likely to result in lower emissions, helping to reduce air pollution in cities.

Methodology

To evaluate the effectiveness of AI-driven traffic light systems, this study utilized a simulation-based approach developed with Python. The simulation compared two scenarios. Scenario one operated using traditional traffic light systems, while scenario two used AI-controlled adaptive systems. To ensure consistent experiment conditions that would allow for a fair comparison, the simulation ran both scenarios under the same precisely controlled conditions (variables), meaning they were tested under the same road infrastructure, intersection design, number of vehicles and traffic flow input. *Figure 1* represents the flowchart of the traffic simulation system, which provides a visual overview of the five main components: the simulation controller, intersection management, vehicle handling, AI system integration, and analysis pipeline.

Figure 1:

System Architecture and Data Flow of the Simulation



Note: C25D.py serves as the primary simulation file containing the core implementation of the traffic control system. See here for an interactive version of this flowchart, where you can zoom and navigate freely.

1. Intersection Modeling and Traffic Control

The heart of the simulation was the Intersection class, which integrated all components of the traffic system and implemented both traditional and AI-based control strategies:

```python
class Intersection:
    def __init__(self,
                 intersection_id,
                 control_type='traditional',# Type of traffic control
        ('traditional' or 'ai')
                 avg_traffic_volume=350, # per hour
                 traditional_cycle_time=120,
                 min_green_time=15,
                 max_green_time=180,
                 emergency_frequency=0.05,
                 road_type="suburban", # Type of road ('suburban' or
        'highway')
                 congestion_level="medium",# Traffic congestion ('low',
'medium', 'high')
                 road_length=100.0, # In meters
                 num_lanes=2 # Number of lanes in each approach
                ):
```

Each intersection parameter was carefully selected based on empirical research:

- The default *avg_traffic_volume* of 250 vehicles per hour is calculated based on an average daily traffic (ADT) of 8,500 vehicles (Kim et al., 2008) distributed across hourly intervals.

- The average cycle time of 120 seconds aligns with findings from NACTO (National Association of City Transportation Officials), which documents typical cycle lengths for urban intersections at this time.
- The *min_green_time* of 15 seconds was based on Federal Highway Administration (FHWA, 2021) recommendations for minimum green phases that allow adequate service for typical vehicle queues.
- The *max_green_time* of 180 seconds followed upper limits suggested by (FHWA, 2021) in their Traffic Signal Timing Manual to prevent excessive delays on cross streets.
- The *emergency_frequency* of 0.05 (5%) was derived from emergency response data analyzed by Pons and Markovchick (2002), who found that emergency vehicles constituted approximately 3-7% of urban traffic during peak hours.

The intersection class implemented two distinct traffic control approaches:

- Traditional fixed-time control: This used predetermined cycle times and phase durations based on historical traffic patterns
- AI-based adaptive control: This utilized a neural network to dynamically adjust signal timing based on real-time traffic conditions

2. Road Network and Infrastructure

The physical infrastructure of the traffic system was modeled through a network of interconnected components:

```python
class RoadSegment:
    def __init__(self,
                 segment_id: str,
                 length: float,  # in meters
```

```
                    num_lanes: int,
                    lane_width: float = 3.7):  # standard lane width in
meters
```

The *lane_width* parameter of 12 feet, or approximately 3.7 meters, represented the standard width recommended by the American Association of State Highway and Transportation Officials (AASHTO, 2018, 196) for urban arterial roads. This value was chosen as it accommodates typical urban traffic while maintaining safety margins for passing vehicles.

The road network was organized through an IntersectionNetwork class that managed the spatial relationships and coordination between multiple intersections:

```python
class IntersectionNetwork:
    def __init__(self):
        self.intersections = {}  # Store intersections by ID
        self.connections = {}    # Store intersection connections
        self.intersection_groups = {}  # For coordinated signal
control
        self.traffic_rules = TrafficRules()  # Common traffic rules
for the network
```

This network structure enabled the modeling of coordinated traffic control across multiple intersections. Moreover, the traffic rules system implemented a complex conflict matrix that managed right-of-way decisions and turning movements. It also featured a sophisticated coordination system between intersections through the IntersectionNetwork class, which managed signal synchronization. Additionally, pedestrian and bicycle integration was handled through dedicated classes (*PedestrianCrossing* and *BicycleLane*).

3. Parameter Adjustments

To account for the impact of road type and congestion level on traffic dynamics, the model incorporated real-world traffic variability through the *_adjust_parameters* method :

```python
def _adjust_parameters(self):
        # Road type adjustments
        if self.road_type == "highway":
            self.avg_traffic_volume *= 15.88
            self.traditional_cycle_time = 150
        elif self.road_type == "suburban":
            self.avg_traffic_volume *= 1.0
            self.traditional_cycle_time = 100
        self.peak_factor = 1.10
        # Congestion level adjustments
        congestion_multipliers = {"low": 0.7, "medium": 1.0, "high":
1.3}
        self.avg_traffic_volume *=
congestion_multipliers[self.congestion_level]
```

This method adjusted the average traffic volume and traditional cycle time based on the road type and congestion level. The adjustments were based on empirical studies:

- For highways, the model increased the traffic volume by 15.88 times, derived from the ratio of average daily traffic (ADT) on highways (135,000 vehicles) to suburban roads (8,500 vehicles) (Kim et al., 2008).

4. Vehicle Modeling

The traffic simulation employs a comprehensive vehicle modeling approach that captures both the physical characteristics and behavioral dynamics of different vehicle types. The methodology classifies vehicles into three distinct categories—cars, SUVs, and trucks—each represented with realistic physical dimensions and performance parameters derived from empirical data.

Each vehicle is modeled with precise physical attributes, including length, width, maximum speed, and turning radius. These parameters reflect real-world variations: Cars are modeled as compact and nimble (4.5 meters length), SUVs as midsize (5.0 meters), and trucks as substantially larger (7.0 meters) with wider turning requirements (Tapani, 2008). These physical dimensions directly influence the vehicle's interaction with road infrastructure and other traffic participants, affecting their space requirements and maneuverability characteristics.

The vehicle model incorporates sophisticated traffic behavior, including turning intentions based on realistic probability distributions: 50% of vehicles proceed straight, 30% turn right, and 20% turn left at intersections (Chen et al., 2023). The model also accounts for vehicle-specific turning dynamics, where larger vehicles like trucks require wider turning radii (150% larger than standard), and all vehicles automatically reduce speed during turns to match realistic driving behavior (Tapani, 2008).

The intricacies of vehicle movement are captured through continuous updates that consider negative and positive acceleration, and interactions with traffic signals and other vehicles. Emergency vehicles are prioritized to minimize response times and receive a 20% higher maximum speed allowance and prioritization as per traffic rules. The program automatically records time spent waiting at traffic lights not only to assess potential reductions in wait times, but to automatically estimate CO_2 emissions due to idling.

5. Traditional Traffic Control System

The cycle time for traditional traffic control was determined based on pre-set schedules designed to mimic real-world, non-adaptive traffic systems. These systems operate with fixed cycle times for different times of the day but do not adjust in real time using live traffic data. A base cycle time was established, and this was then modified by predetermined factors to account for typical traffic patterns during morning and evening peak hours and different road types (e.g., highway versus suburban road). The cycle time was then fixed with those parameters. These adjustments ensured that the green light duration reflected expected traffic loads without any real-time responsiveness to traffic fluctuations. The adjusted cycle time was constrained within practical limits based on standard traffic engineering guidelines, and green times were calculated considering the durations of yellow and all-red clearance phases to maintain safety and efficiency.

6. AI-Based Traffic Control System

The adaptive AI-based traffic control system utilized a neural network approach implemented using PyTorch to dynamically adjust signal timing based on real-time traffic conditions. This approach should–in theory–engender significant improvements in wait time, as it responds dynamically to changing traffic patterns as opposed to repeating intervals based on historical patterns.

The AI model has access to a comprehensive set of real-time traffic metrics in the simulation to determine optimal signal timing, including traffic volume, waiting times, time of day, presence of emergency vehicles, pedestrian and bicycle counts, congestion level, and average vehicle speeds. This gave the model access to a wide range of data that it was able to use to make effective decisions regarding interval timing.

Reinforcement learning (RL) was integrated into the AI system to guide its learning process. Utilizing a deep neural network architecture with three hidden layers–each containing 64 neurons, it enabled the AI to improve its decisions over time and translate human goals into mathematical optimization targets. It quantified how well the traffic management decisions performed by calculating a numerical value where higher scores indicated better performance. It was designed to balance multiple objectives: minimizing wait time, maximizing throughput, maximizing speed, minimizing queue length, and minimizing emergency vehicle latency to name a few.

```python
    def calculate_reward(self):
        """Calculate reward based on intersection performance"""
        # Get current metrics
        vehicle_count = self.get_vehicle_count()
        wait_time = self.calculate_wait_time(vehicle_count,
self.traffic_light.green_time)
        queue_length = self._calculate_queue_length()
        avg_speed = self._calculate_average_speed()
        emergency_count = sum(1 for v in self.get_all_vehicles() if
v.is_emergency)

        # Base reward inversely proportional to wait time
        max_expected_wait = 120.0  # seconds
        normalized_wait = min(wait_time / max_expected_wait, 1.0)
        base_reward = 100 * (1 - normalized_wait)

        # Reward for throughput
        throughput_reward = 0
        if self.last_state is not None:
            last_queue = self.last_state.get("queue_length", 0)
            queue_change = last_queue - queue_length
            throughput_reward = max(0, queue_change) * 10

        # Speed reward
        max_expected_speed = 15.0  # m/s
```

```
            normalized_speed = min(avg_speed / max_expected_speed, 1.0) if
avg_speed > 0 else 0
            speed_reward = normalized_speed * 50

            # Queue length penalty
            road_capacity = self.road_length * self.num_lanes * 4
            queue_ratio = min(queue_length / road_capacity, 1.0)
            queue_penalty = -80 * queue_ratio

            # Emergency vehicle penalty
            emergency_penalty = 0
            if emergency_count > 0:
                emergency_vehicles = [v for v in self.get_all_vehicles() if
v.is_emergency]
                stopped_emergency = sum(1 for v in emergency_vehicles if
v.speed < v.stopped_speed)
                if stopped_emergency > 0:
                    emergency_penalty = -100 * stopped_emergency

            # Total reward
            total_reward = (base_reward + throughput_reward + speed_reward +
                           queue_penalty + emergency_penalty)
            return max(-200, min(total_reward, 200))  # Capped between -200
and 200
```

The base reward was inversely proportional to wait time, normalized against a maximum expected wait of 120 seconds. This provided a foundation that encouraged the reduction of vehicle waiting times. Additional reward components included throughput reward (10 points per reduction in queue length), speed reward (up to 50 points based on normalized average speed), and penalties for queue length (-80 points maximum) and stopped emergency vehicles (-100 points per vehicle). The final reward was capped between -200 and 200 to prevent extreme values from destabilizing the learning process. This mathematical balancing allowed the AI to learn and develop complex traffic management strategies that no human programmer could explicitly code as fixed rules, allowing it to adapt to the actual traffic conditions it encounters.

7. Simulation Procedure and Experimental Design

The experimental design employed a comparative approach that directly evaluated traditional fixed-time traffic control against AI-based adaptive control under identical conditions. This controlled comparison allowed for a rigorous assessment of potential improvements offered by AI-based systems.

The simulation environment modeled multiple urban intersections with varying characteristics. Ten intersections were simulated, each with different configurations of traffic volume, road types (suburban and highway), and congestion levels (low, medium, and high). By testing across this range of conditions, the study ensured results would be applicable to diverse real-world scenarios.

Each intersection was simulated twice - once with traditional control and once with AI control - while maintaining identical traffic conditions, ensuring a fair comparison. The simulation spanned a seven-day period to capture complete weekly traffic patterns, including both weekdays commuting patterns and weekend variations.

The temporal dimension was carefully structured, with each day divided into hourly simulations. Within each hour, the simulation used small time increments to accurately model vehicle movements, traffic signal changes, and interactions between different road users.

Traffic volumes varied realistically throughout the day, with higher volumes during morning and evening rush hours and lower volumes overnight.

8 . Performance Metrics and Analysis Methods

The simulation employed multiple performance metrics to comprehensively evaluate both traditional and AI traffic control systems. Key metrics included average vehicle wait time (seconds), emergency vehicle response times, total vehicle throughput, fuel consumption (liters), and emissions (CO_2 grams). These metrics were chosen to assess both operational efficiency and environmental impact. Additionally, secondary metrics tracked pedestrian and bicycle waiting times to ensure multimodal optimization.

```python
# Calculate performance metrics
metrics = {}
# Average wait times
wait_times = grouped['wait_time'].mean()
metrics['wait_time'] = {
    'traditional': wait_times['traditional'],
    'ai': wait_times['ai']
}
# Calculate improvement percentages
improvements = {}
for metric in metrics:
    trad_value = metrics[metric]['traditional']
    ai_value = metrics[metric]['ai']
    improvement = ((trad_value - ai_value) / trad_value) * 100
    improvements[metric] = improvement
```

Statistical analysis utilized Welch's t-test to determine if differences between control systems were statistically significant ($p < 0.05$). This non-parametric approach was selected as it does not assume equal variances between groups, making it more suitable for traffic data, which often exhibits heteroscedasticity. The analysis also calculated percentage improvements across different metrics, with particular attention to peak-hour performance when traffic systems face their greatest challenges.

```python
    # Perform statistical significance tests
    stats_results = {}
    for metric in metrics:
        trad_data = df[df['control_type'] == 'traditional'][metric]
        ai_data = df[df['control_type'] == 'ai'][metric]

        t_stat, p_value = stats.ttest_ind(trad_data, ai_data,
equal_var=False)
        stats_results[f'{metric}_ttest'] = {
            't_statistic': t_stat,
            'p_value': p_value,
            'significant': p_value < 0.05
        }
```

9. Environmental Impact Modeling

The environmental impact assessment integrated fuel consumption and emissions modeling to quantify the ecological footprint of different traffic control strategies. This dual approach allowed for a comprehensive evaluation of environmental consequences beyond traditional traffic efficiency metrics. The model accounted for both idling consumption during waiting periods and the additional fuel used during acceleration phases after stopping, which together constitute the primary sources of urban traffic emissions.

Python
```
        def calculate_fuel_consumption(self, wait_time, vehicle_count,
hour=None):
            if vehicle_count == 0:
                return 0.0

            # Base consumption rates per vehicle type (liters per second while
idle)
            car_idle_rate = self.vehicle_specs['car']['idle_rate']
            suv_idle_rate = self.vehicle_specs['suv']['idle_rate']
            truck_idle_rate = self.vehicle_specs['truck']['idle_rate']

            # Calculate consumption based on vehicle distribution
            car_percentage = self.vehicle_specs['car']['percentage']
            suv_percentage = self.vehicle_specs['suv']['percentage']
            truck_percentage = self.vehicle_specs['truck']['percentage']

            # Average idle rate weighted by vehicle type distribution
            avg_idle_rate = (car_idle_rate * car_percentage +
                             suv_idle_rate * suv_percentage +
                             truck_idle_rate * truck_percentage)

            # Calculate base fuel consumption from idling
            base_consumption = avg_idle_rate * wait_time * vehicle_count

            # Add fuel consumed during acceleration after stopping
            car_accel = self.vehicle_specs['car']['acceleration']
            suv_accel = self.vehicle_specs['suv']['acceleration']
            truck_accel = self.vehicle_specs['truck']['acceleration']
```

```
        avg_accel_consumption = (car_accel * car_percentage +
                                 suv_accel * suv_percentage +
                                 truck_accel * truck_percentage)

        acceleration_consumption = avg_accel_consumption * vehicle_count

        # Total fuel consumption
        return base_consumption + acceleration_consumption
```

The emissions modeling component translated fuel consumption into environmental pollutants using established conversion factors. The model incorporated both direct CO_2 emissions from fuel combustion and adjusted for additional factors that influence real-world emissions, such as the engine inefficiency during prolonged idling periods.

```python
    def calculate_emissions(self, fuel_consumption, vehicle_count,
wait_time):
        """Calculate emissions based on fuel consumption"""
        if vehicle_count == 0 or fuel_consumption == 0:
            return 0.0

        # CO2 emissions in grams per liter of fuel
        co2_per_liter = 2300  # grams

        # Calculate CO2 emissions
        co2_emissions = fuel_consumption * co2_per_liter
```

```
            # Add inefficiency factor for prolonged idling
            # Engine catalytic converters are less effective during extended
idling
             if wait_time <= 10:
                    inefficiency_factor = 1.0  # No increase for wait times
             up to 10 seconds
             elif wait_time <= 60:
                    inefficiency_factor = 1.2  # 20% increase for wait times
                    between 10-60 seconds
             elif wait_time <= 300:
                    inefficiency_factor = 1.3  # 30% increase for wait times
                    between 1-5 minutes
             else:
                    inefficiency_factor = 1.4  # 40% increase for wait times
                    exceeding 5 minutes
         return co2_emissions * inefficiency_factor * emissions_factor
```

To account for idling inefficiencies, the code incorporates a tiered system for prolonged waiting times. Research indicates that catalytic converters lose effectiveness during extended idling, leading to increased emissions (Transport Research Laboratory et al., 2020). For example, idling for over five minutes increases emissions by 40%, while shorter durations between 10 seconds and one minute increase emissions by 20%. These thresholds are implemented using conditional logic to dynamically adjust the inefficiency factor.

10. Result Visualizations and Reporting

The simulation results were visualized through a comprehensive set of charts and graphs to facilitate interpretation. The system stored detailed logs of the simulation as well as performance metrics, storing them in the results/directory with timestamp-based organization, which can be found in the appendix.

Findings & Analysis

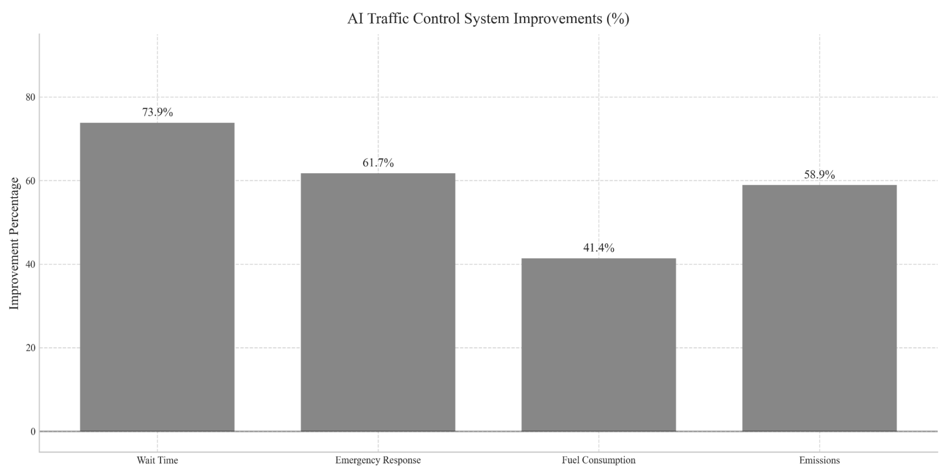
This section presents the findings from the traffic simulation, comparing the performance of the AI-optimized traffic system against traditional traffic light control. The analysis focuses on key performance indicators (KPIs), including average wait times, fuel consumption, emergency vehicle response times, and overall system efficiency. The results demonstrate the extent to which AI can optimize traffic flow and mitigate common urban traffic challenges.

As illustrated in *Figure 2*, the AI-based traffic control system demonstrated significant improvements over traditional systems in four key performance areas:

- Wait time reduction: 73.9% decrease
- Emergency response improvement: 61.7% decrease in emergency vehicle wait times
- Fuel consumption reduction: 41.4% decrease
- Emissions reductions: 58.9% decrease

Figure 2

AI Traffic Control System Improvements (%)

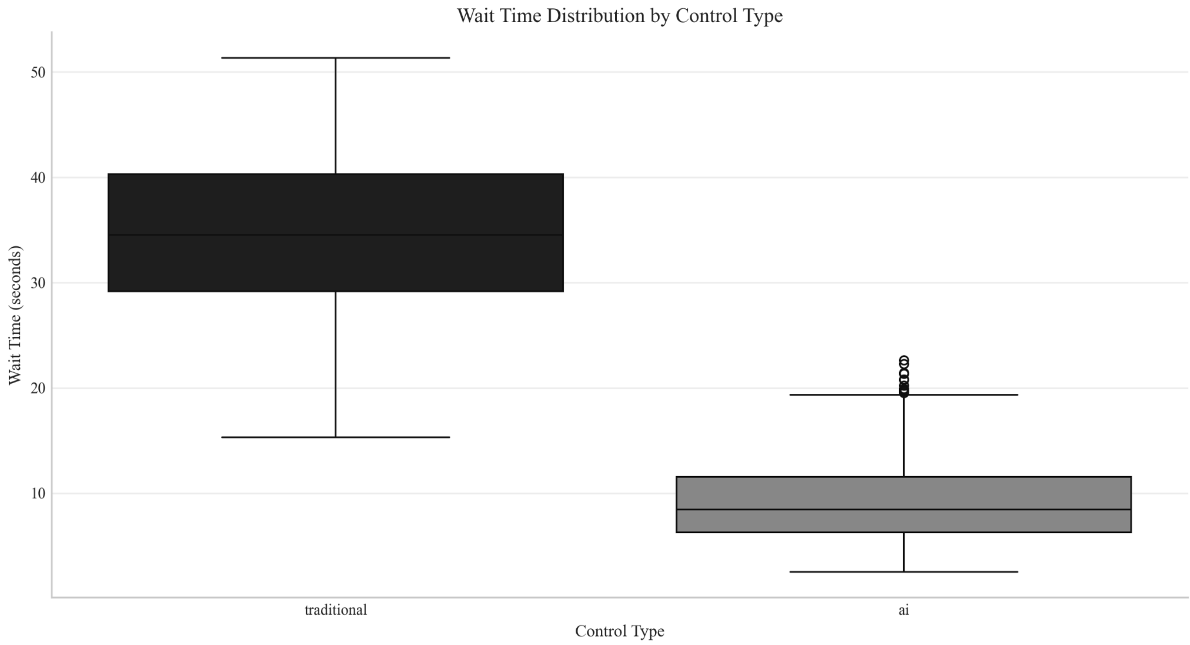


Wait Time Reduction

Figure 3 shows a stark contrast between the two control systems regarding average vehicle wait times. The traditional traffic control system resulted in significantly longer wait times per intersection (mean of 34.81 seconds ± 7.45) compared to the AI-controlled system (mean of 9.10 seconds ± 3.61). This represents a 73.9% reduction in wait time, which is statistically significant (Mann-Whitney U = 4063478.0, $p < 0.001$).

Figure 3

Wait Time Distribution by Control Type

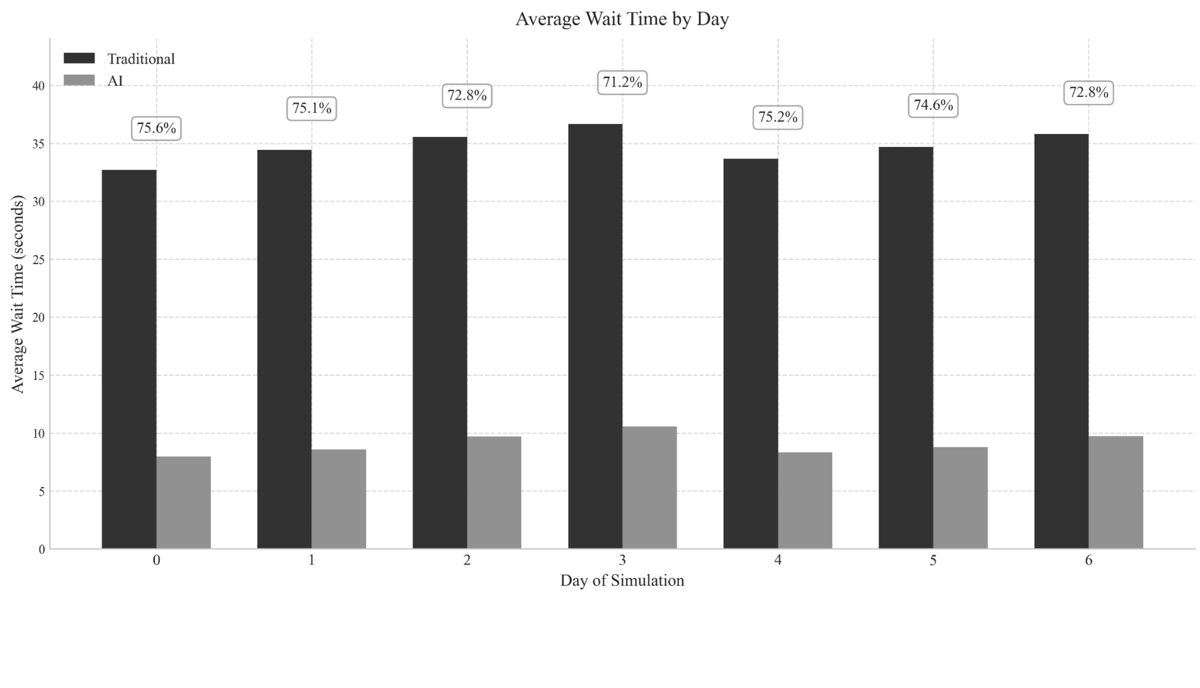


Note: The circles represent outliers, likely caused by delays due to the prioritization of emergency vehicles, which increased individual wait times. This is further discussed in the Emergency Vehicle Response Times section.

The statistical analysis further validated these findings through bootstrap confidence intervals with 10,000 resamples at a 95% confidence level. The difference in mean wait times was 25.71 seconds with a 95% confidence interval of [25.12, 25.63] seconds, demonstrating the robustness of the observed improvement. The lower variability in wait times for the AI system (standard deviation: 3.61 seconds) compared to the traditional system (standard deviation: 7.45 seconds) indicates that the AI system not only reduced average wait times but provided more consistent performance across different traffic conditions. The AI system demonstrated consistency of improvement and stable performance across all days, with daily averages ranging from 7.89 to 11.11 seconds, while the traditional system showed higher variability with daily averages between 32.81 and 36.06 seconds (*Figure 4*). The overall day-to-day improvement averaged 73.5% ± 2.3%, showing remarkable consistency in the AI system's performance advantage.

Figure 4

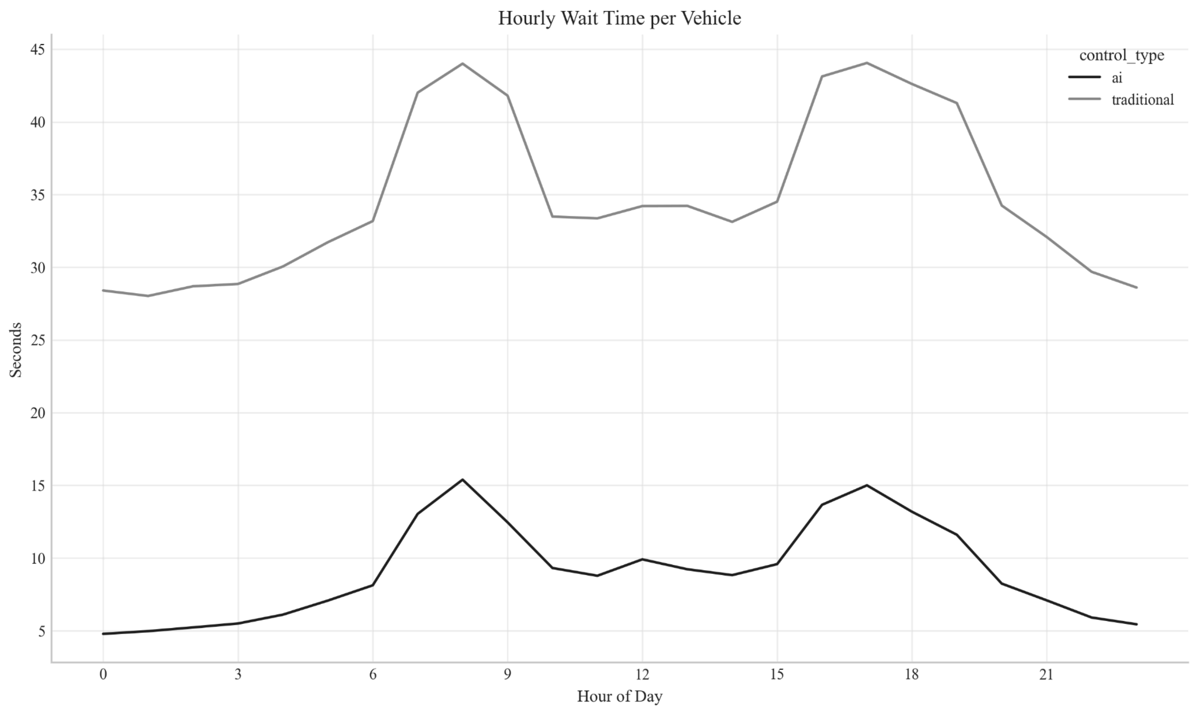
Average Wait Time by Day



As shown in *Figure 5*, the consistent gap between the two systems throughout the 24-hour period demonstrates the AI system's superiority across all traffic conditions, not just during specific periods. The proportional improvement varies throughout the day, with the greatest percentage improvement occurring during early morning hours (0-5), where the AI system reduces wait times by approximately 82%.

Figure 5

Hourly Wait Time per Vehicle



During peak hours, while the absolute difference is larger (approximately 29 seconds), the percentage improvement decreases to approximately 65.0%. This pattern suggests that while the AI system maintains significant advantages under all conditions, its relative efficiency advantage is somewhat constrained during extreme congestion periods, pointing to physical limitations that affect both systems.

Intersection Type Performance

The analysis of different intersection types revealed that the AI system maintained consistent performance across all intersection configurations, while the traditional system showed significant variations between highway and suburban intersections.

As shown in *Table 1*, highway intersections under traditional control experienced the longest wait times, averaging approximately 39.41 seconds across all traffic volumes (39.30 s, 39.72 s, and 39.21 s for high, medium, and low congestion respectively), while suburban intersections averaged around 31.45 seconds (31.56 s, 31.30 s, and 31.48 s). In contrast, the AI system maintained remarkably consistent wait times between 8.97 and 9.48 seconds across all intersection types, representing a reduction of 76.3% for highway intersections and 71.2% for suburban intersections.

Table 1

Wait Time and Cycle Time Performance by Intersection Type

Type	Congestion	System	Wait time (s)
Highway	High	Traditional	39.30 ± 5.63
Highway	High	AI	9.29 ± 3.58
Highway	Medium	Traditional	39.72 ± 5.55
Highway	Medium	AI	9.48 ± 3.80
Highway	Low	Traditional	39.21 ± 5.88
Highway	Low	AI	9.17 ± 3.55
Suburban	High	Traditional	31.56 ± 6.7
Suburban	High	AI	9.08 ± 3.50
Suburban	Medium	Traditional	31.30 ± 6.95
Suburban	Medium	AI	9.31 ± 3.72
Suburban	Low	Traditional	31.48 ± 6.77
Suburban	Low	AI	8.97 ± 3.85

Note: Values represent mean wait times with standard deviations.

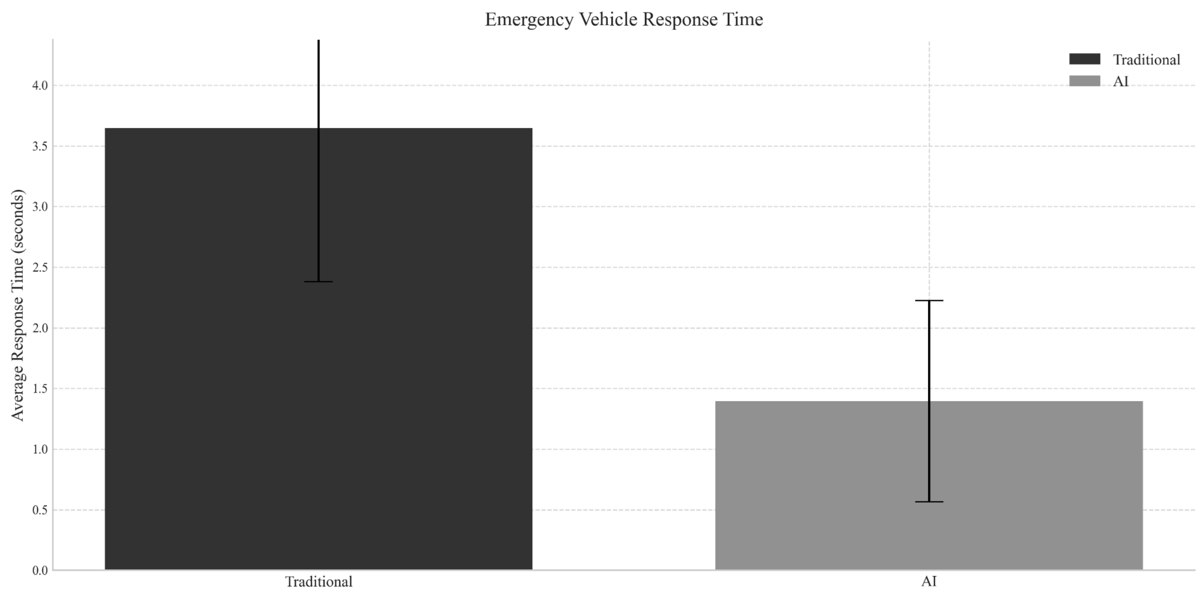
The AI system's consistent performance across different intersection types suggests that it is adaptable and robust. Unlike traditional systems, which exhibit significant variability in performance based on intersection type, the AI system can maintain optimal traffic flow regardless of the specific conditions. This adaptability is crucial for managing the complexities of urban traffic, where conditions can change rapidly.

Emergency Vehicle Response Times

Emergency vehicle response times showed similar patterns of improvement. As shown in *Figure 6*, the AI traffic control system reduced emergency vehicle wait times by 61.7% compared to traditional systems. This translates to emergency vehicles spending approximately just 1.41 seconds (±0.82) at intersections under AI control versus 3.69 seconds (±1.22) with traditional control mechanisms.

Figure 6

Emergency Vehicle Response Time



From a public safety standpoint, this improvement is especially important since shorter emergency response times directly correspond to better outcomes in an emergency.

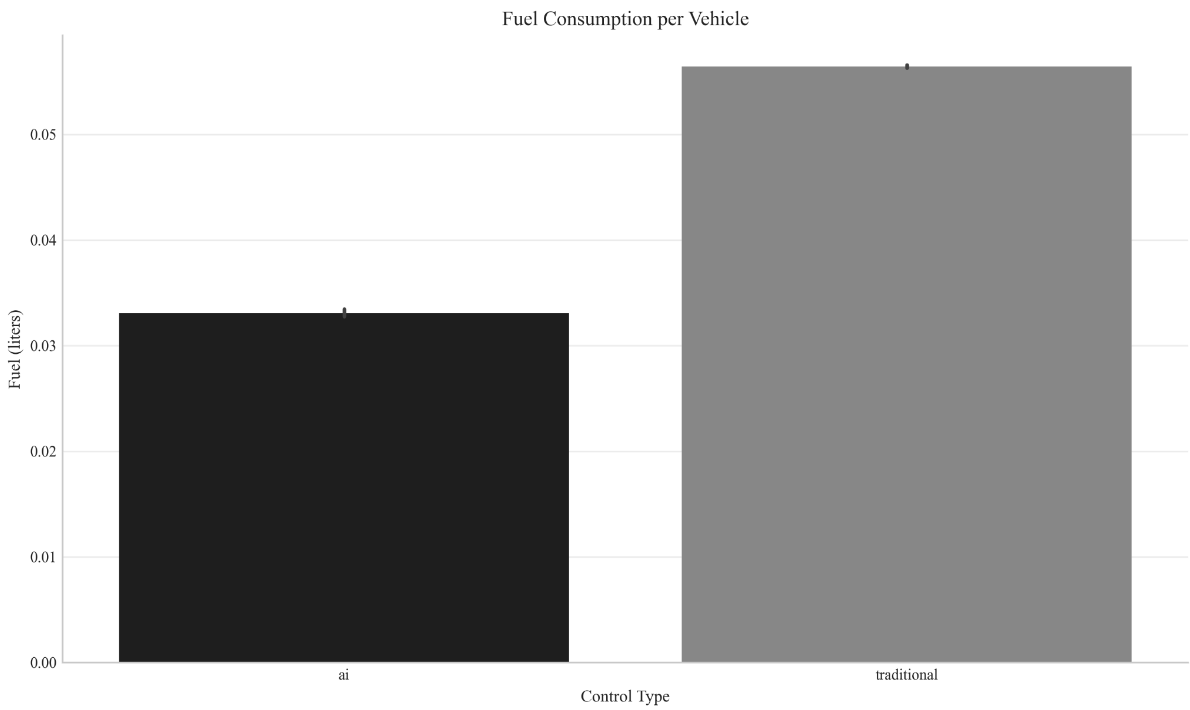
With a 95% confidence interval for the difference of [2.19, 2.32] seconds, the statistical significance of this variance was verified ($p < 0.001$, Mann-Whitney $U = 3786133.0$). Although the differences seem small in absolute terms, they add up over several intersections during an emergency response path and can greatly affect results in life-threatening events where every second counts. The data shows clear differences in emergency vehicle wait times across different hours of the day; both systems show higher delays during periods of maximum traffic. During the morning (7-9 AM) and evening (4-7 PM) rush hours, traditional systems led to waiting times reaching up to 4.81 seconds (peak at hour 16), while AI systems maintained a relatively consistent performance even under congested conditions, typically keeping emergency wait times below 1.55 seconds. For emergency response planning in particular, this consistency helps since it offers more consistent travel times independent of traffic conditions. The AI system achieves this through its preemption capability: Unlike traditional systems where emergency vehicles must slow down at intersections with red lights to avoid colliding with vehicles approaching from crossroads, the AI system preemptively detects approaching emergency vehicles (60.0% preemption rate) and adjusts signals to create clear paths and open ways through intersections. This design approach reflects real-world scenarios while enhancing safety; emergency vehicles in AI-controlled intersections experience minimal slowdowns and safer passage conditions. Even during the most congested periods, the maximum emergency wait time for AI-controlled intersections (1.55 seconds during peak) remains substantially lower than the traditional system's average (3.69 ± 1.22 seconds), demonstrating how intelligent traffic management can maintain critical emergency service performance even under challenging traffic conditions.

Fuel Consumption

The AI condition/test condition yielded notable improvements in fuel usage, as seen in *Figure 7*, yielding an improvement from 0.055 liters to 0.035 liters in fuel consumed per intersection, a 36% reduction.

Figure 7

Fuel Consumption per Vehicle



Note: Y-axis represent liters consumed per intersection visit

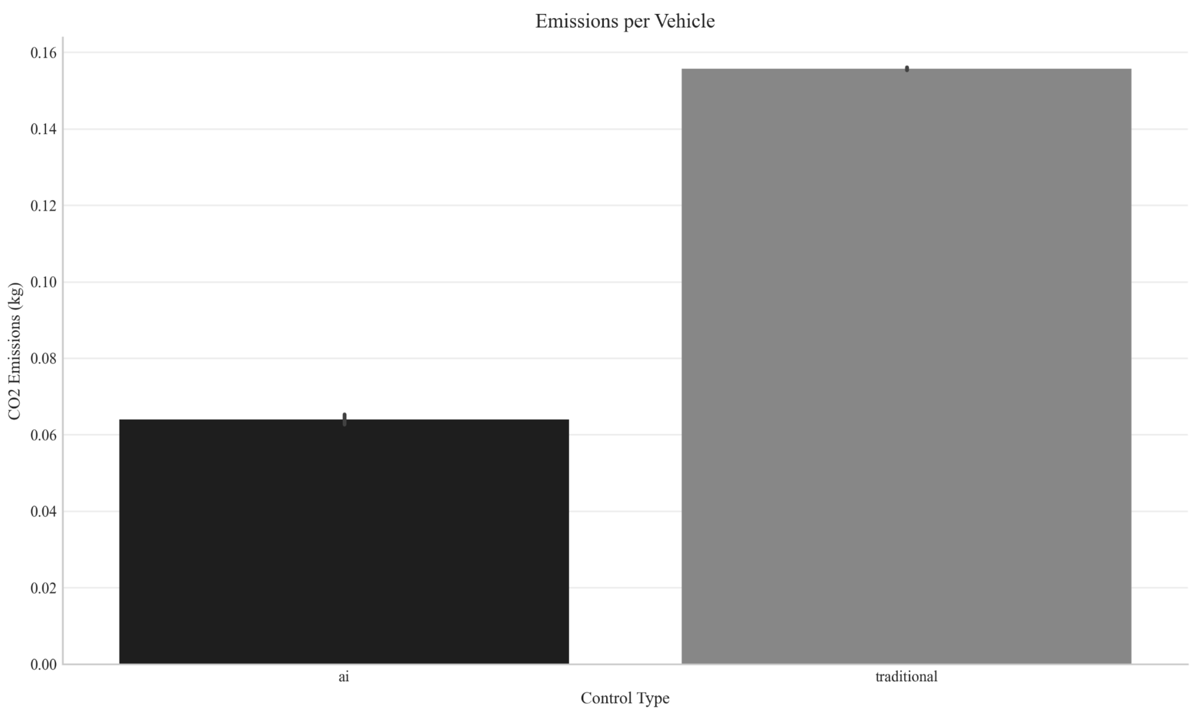
The improvement in fuel consumption was statistically significant ($p < 0.001$). There was a clear correlation between fuel consumption and time spent idling at intersections, with each additional second of idling associated with a proportional increase in fuel usage. This relationship helps explain the AI system's enhanced fuel efficiency: by minimizing wait times and improving traffic flow, it concurrently reduced fuel consumption. This difference was particularly evident during peak hours, specifically 7-9AM and 4-7PM, where the larger traffic volume resulted in notably higher wait times and therefore higher fuel consumption in the traditional system. While these effects were still present in the AI system, it managed traffic more effectively, resulting in a substantial reduction in overall fuel consumption, as illustrated in *Figure 3*.

Emissions Reduction Performance

The AI traffic control condition drove significant reductions in CO_2 emissions. This is demonstrated in *Figure 8*, where approximately 0.063 kg of CO_2 were emitted per vehicle as opposed to 0.156 kg in the traditionally controlled system, suggesting a dramatic reduction of almost 60%.

Figure 8

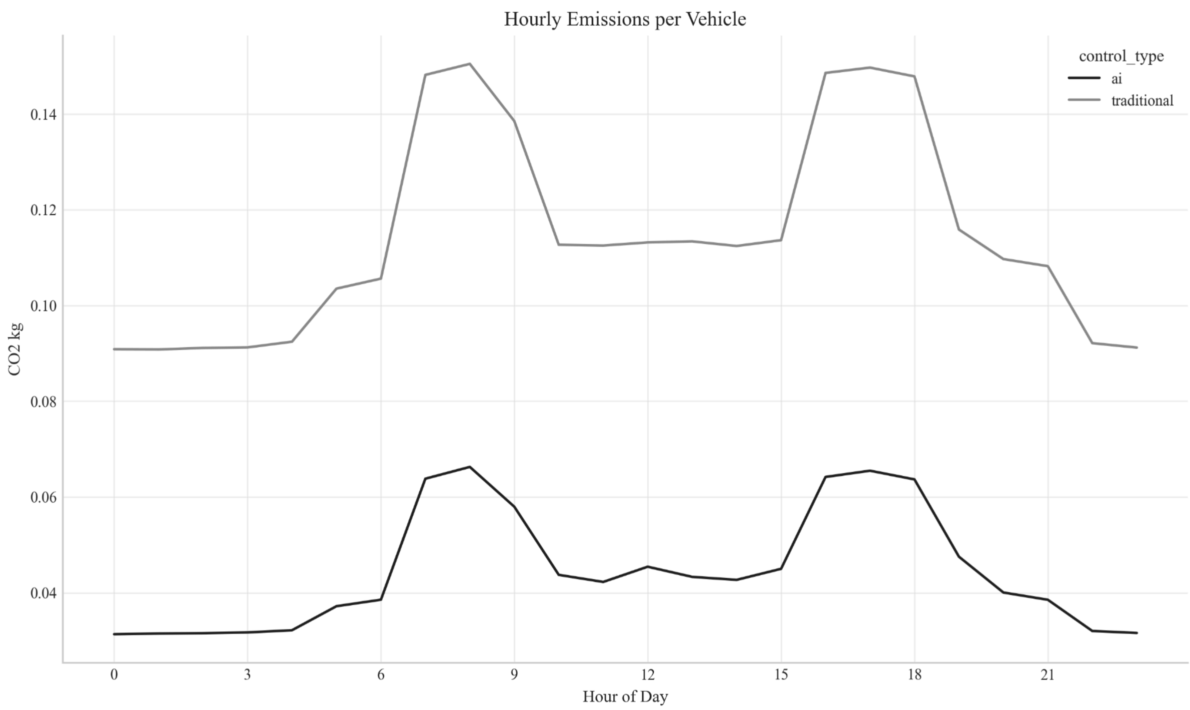
Emissions per Vehicle



As illustrated in *Figure 9*, morning peak hours (7-8 AM) generated the highest emissions in both systems, with the traditional system producing up to an overall of 42,810 kg of CO_2 at 7 AM compared to 17,645 kg from the AI system, representing a 58.8% improvement. This demonstrates the ability of the AI system to outperform the traditional alternative, even under extreme conditions.

Figure 9

Hourly Emissions per Vehicle



The experiment utilized a Mann-Whitney U test, as the data were unlikely to follow a normal distribution. The results indicated that the observed differences were highly significant (Mann-Whitney U = 354,166, $p < 0.001$). The consistency of these reductions across different traffic conditions and times of day suggests that the AI system's improvements in emissions are robust. This demonstrates its ability to rapidly adjust to dynamic environments, signaling its potential effectiveness for real world applications.

Comparison with Existing Research

The study found significant improvements across all measured metrics, including travel times, fuel consumption, and emissions. These results are consistent with other studies but show higher performance improvements in some metrics. For instance, the “Smart Junctions” project reported a 34% reduction in wait times (VIVACITY, 2020), while the AI4UTC trial in Huddersfield achieved a 60% reduction in congestion during peak hours (Court-Dobison & Lawson, 2024). However, the Autonomous Smart Traffic Management System (ASTMS) showed a 70% reduction in vehicle pass delays and a 50% increase in traffic flow rates (Goenawan, n.d.). These differences show the differences in performance across different contexts and simulation setups.

The superior performance observed in this study may be attributed to several factors. First, the simulation design was highly controlled with consistent data inputs across 10 locations and 3,360 hours of simulation time, which likely minimized external variability. In contrast, real-world trials such as those conducted by AI4UTC had limitations like delayed sensor data and pre-uploaded strategies, which constrained real-time adaptability (Court-Dobison & Lawson, 2024). Furthermore, this study’s reinforcement learning model was extensively trained and optimized for specific traffic scenarios, which may have given it an advantage over less specialized implementations, such as the “Smart Junctions” project, which focused on general-purpose adaptability (VIVACITY, 2020). Lastly, the differences in metrics (e.g., congestion vs. wait times) and urban layouts (e.g., single intersections vs. multi-location networks) also explain the variation in results.

Assumptions and Limitations

Despite efforts to create a realistic traffic simulation environment, several assumptions were necessary to model traffic behavior and infrastructure characteristics. Vehicle dimensions were simplified and based on common vehicle sizes using standardized sizes gathered through secondary research (Extra Space, 2025): cars (4.5 m × 1.8 m), SUVs (5.0 m × 2.0 m), and trucks (7.0 m × 2.5 m). This assumption does not account for variations in vehicle sizes due to modifications, different manufacturers, or unique vehicle types, such as motorcycles and buses. Another key assumption was made for the traffic composition, which assumed it to be 60% cars, 30% SUVs, and 10% trucks, with emergency vehicles comprising 5% of total traffic, as specific regional distribution data and actual emergency vehicle frequencies in urban environments were unavailable. Vehicle speeds were ceiled at 50 km/h for cars and SUVs and 40 km/h for trucks, which, although based on common urban speed limits, do not reflect a real-world scenario. Traffic patterns were modeled using time-based multipliers with peak hours (7-9 AM, 4-7 PM) experiencing up to 1.8 times the base volume of vehicles per hour, while off-peak periods saw reduced volumes down to 0.3 times during night hours (22-5). Pedestrian (20% arrival rate) and bicycle (10% arrival rate) interactions were simplified, potentially underestimating their impact on traffic flow in heavily populated areas. The road infrastructure assumed standardized dimensions with 3.7m lane widths (following standard highway design guidelines), 100 m road segments (chosen for computational efficiency while maintaining realistic intersection spacing), and 30 m turn lanes. The model assumes uniform driver behavior and perfect compliance with traffic rules, neglecting the unpredictability of human behavior and potential traffic violations. Fuel consumption rates were estimated at 0.00025 L/s (cars), 0.0004 L/s (SUVs), and 0.0006 L/s (trucks) for idling, with acceleration consumption of 0.015 L, 0.025 L, and 0.04 L per stop-start cycle respectively; these values were approximated from various vehicle efficiency studies but may vary significantly based on vehicle age and maintenance.

Furthermore, several limitations affect the simulation's scope and accuracy that must be acknowledged. From a technical perspective, computational constraints restrict the simulation to processing only 10 intersections simultaneously and require fixed road segment lengths of 100 m to manage memory usage efficiently. The model employs necessary simplifications such as linear acceleration/deceleration models and discrete time steps rather than continuous simulation. Notable traffic flow limitations include the inability to simulate lane-changing behavior, merging traffic, complex turning movements, vehicle overtaking, or traffic accidents. Moreover, weather conditions, road maintenance, and other environmental factors that could impact traffic flow were not considered. The behavioral aspect of traffic simulation is particularly constrained, as the model cannot replicate human factors such as driver aggression, distraction, variable reaction times, or route choice decisions. Also, the simulation uses a simplified road network design with uniform intersection spacing and does not account for complex road geometries or varying grades that might affect vehicle performance. The AI system's learning process, while effective, operates under ideal conditions with perfect sensor data and instantaneous response times, which may not be achievable in real-world implementations. All these assumptions and limitations suggest that while the simulation provides valuable insights into the potential improvements that could be achieved through the AI-controlled traffic systems, it does not necessarily provide the accurate magnitude of these improvements.

Conclusion

The experiment provides strong evidence that AI has the potential to transform the way that we handle commuting and traffic as a whole. This study found statistically significant reductions in wait time, CO_2 emissions, fuel consumptions and emergency response times, which could have wide-reaching impacts on not only individual livelihood but also potentially save lives both through combating climate change and reducing emergency response times. The gathered data strongly supports the hypothesis that "AI-integrated traffic lighting systems will outperform the traditional interval-based system, showing significant improvements in several key areas". There are noted improvements in wait time, emergency response, carbon emissions, and fuel consumption, and statistical analysis suggests that these results are both accurate and reproducible. The study design facilitated the minimization of extraneous variables, which suggests with a high degree of certainty that these improvements were due to the implementation of AI traffic management systems in the simulation.

Real-World Implications

In light of the fairly conclusive results of the study, it is important to consider the real-world implications of implementing such a system. AI systems could in theory help enhance quality of life significantly, as reduced commuting times (which the evidence suggests this system would effectively do), are strongly correlated with increased happiness and wellbeing (Hen, 2022). Overall, the implementation of this system could yield significant positive impacts on the world at large. However, as with all potential benefits of integrating AI, there are numerous safety considerations that must be taken into account: Firstly, the simulation used a simplified traffic model; it remains challenging to fully replicate the complexities of human behavior in a computational model. This could lead to potential fringe cases: For example, would the AI automatically shut off in case of a traffic accident? Would it be able to recognize all types of vehicles equally well? Could pedestrians who are used to relying on patterns be confused by inconsistent traffic light behavior which could then lead to accidents? These are concerns that would best be assessed with a small scale test in real life. While this study has accounted for some of these factors, like introducing a minimum cycle time in the simulation to ensure human drivers have enough time to react, and a hard lock preventing multiple green lights on the same crossing, it is still impossible to fully account for real world factors without an ecological study. More research is needed.

Additionally, there will likely be a lot of variance across different economic and developmental situations. In countries with more intricate road networks, implementing AI systems could be more challenging. Conversely, countries with relatively bare networks might not benefit significantly enough from this system to justify the increased costs of this approach. The system could also be more liable to hacking, as it would require communication with an external system. The fact that large quantities of data would need to be communicated back and forth with the AI system at low latency, introduces significant technical challenges and is likely the reason why such a system has not been implemented as of yet. Another point of concern would be the detection mechanism for traffic volume: it might be possible to use existing traffic cameras and integration with CV programs, however, this raises both technical and ethical concerns, particularly regarding privacy and data security. With the government having access to extensive sensitive demographic data that could potentially be misused as vulnerabilities could be exploited, leading to disruptions in traffic flow or even malicious attacks on critical infrastructure. That said, it is clear that the benefits of this system are high, and deserve closer scrutiny to see how they can be best used. As technology advances at its current rapid pace, it is envisaged that these ideas will be further refined, which will enable further, more focused research into the validity of this model.

Works Cited

American Association of State Highway and Transportation Officials. (2018). *A Policy on Geometric Design of Highways and Streets, 2018*. American Association of State Highway and Transportation Officials.

Anysphere Inc. (2025). *Cursor* (0.47) [AI code editor].

Beedham, M. (2022). *The True Environmental Cost of Inner-city Congestion*. TomTom. https://www.tomtom.com/newsroom/explainers-and-insights/the-true-environmental-cost-of-inner-city-congestion/

Chen, D., Zhu, M., Yang, H., Wang, X., & Wang, Y. (2023). *Data-driven Traffic Simulation: A Comprehensive Review*. Arxiv. https://arxiv.org/pdf/2310.15975

Court-Dobison, C., & Lawson, J. (2024, August 22). *Can AI control traffic lights, signals and intersections?* Traffic Technology Today. https://www.traffictechnologytoday.com/features/feature-should-ai-control-traffic-lights.html

EPHA. (2020). *Health costs of air pollution in European cities and the linkage with transport*. European Public Health Alliance (EPHA). https://epha.org/wp-content/uploads/2020/10/final-health-costs-of-air-pollution-in-european-cities-and-the-linkage-with-transport.pdf

Extra Space. (2025). *Average Car Dimensions: A Guide to Length, Width, & Height*. Extra Space Storage. https://www.extraspace.com/blog/self-storage/average-car-dimensions/

FHWA. (2021). *Traffic Signal Timing Manual*. Federal Highway Administration. https://ops.fhwa.dot.gov/publications/fhwahop08024/chapter5.htm

Goenawan, C. R. (n.d.). *ASTM : Autonomous Smart Traffic Management System Using Artificial Intelligence CNN and LSTM*. Arxiv. https://arxiv.org/html/2410.10929v6

Hen, L. (2022, December 29). The Effect of Commuting Time on Quality of Life: Evidence from China. *International Journal of Environmental Research and Public Health*, *20*(1). https://www.mdpi.com/1660-4601/20/1/573

INRIX. (2024). *INRIX 2024 Global Traffic Scorecard: Employees & Consumers Returned to Downtowns, Traffic Delays & Costs Grew*. INRIX. https://inrix.com/press-releases/2024-global-traffic-scorecard-us/

Kim, C., Park, Y., & Sang, S. (2008). Spatial and temporal analysis of urban traffic volume. *Ersi*. Author: Kim, C., Park, Y., & Sang, S. Title: Spatial and temporal analysis of urban traffic volume Year: 2008 Publisher: Esri URL: https://proceedings.esri.com/library/userconf/proc08/papers/papers/pap_1613.pdf

Levy, J. I. (2010). Evaluation of the public health impacts of traffic congestion: a health risk assessment. *Environmental Health*. https://ehjournal.biomedcentral.com/articles/10.1186/1476-069X-9-65#citeas

NACTO. (2009). *Signal Cycle Lengths*. National Association of City Transportation Officials. https://nacto.org/publication/urban-street-design-guide/intersection-design-elements/traffic-signals/signal-cycle-lengths/

Tapani, A. (2008). *Traffic Simulation Modelling of Rural Roads and Driver Assistance Systems*. Digitala Vetenskapliga Arkivet. https://www.diva-portal.org/smash/get/diva2:63/FULLTEXT01.pdf

Transport Research Laboratory, Barlow, T., & Cairns, O. (2020). *Idling Action Research - Review of Emissions Data*. https://www.trl.co.uk/uploads/trl/documents/PPR987-TRL-Idling-Analysis-Summary.pdf

VIVACITY. (2020). *SMART JUNCTIONS: TRANSFORMING TRAFFIC IN GREATER MANCHESTER WITH ARTIFICIAL INTELLIGENCE*. JCT Consultancy. https://www.jctconsultancy.co.uk/Symposium/Symposium2020/PapersForDownload/7%20AI%20to%20control%20traffic%20signals%20Lessons%20from%20live%20deployments%20in%20Manchester%20-%20Raquel%20Velasco%20and%20David%20Watts.pdf

Appendix

The simulation code used in this research is available on GitHub at the following link:

GitHub Repository:

https://github.com/sahandkhademi/traffic_system_ai_traditional_stimulation