Traffic Signal Timing Optimization and Planning Recommendations for Urban Intersections

Mini Project –I (CV380) Report

submitted in partial fulfilment of the required degree of

BACHELOR OF TECHNOLOGY

in

Civil Engineering

by

Bhuvan Dharwad (231CV214)

Bidisha Koley (231CV215)

Abhijith Sogal (231CV203)

Vidya S. J (231CV155)

Under the guidance of

Dr. Suresha S. N

DEPARTMENT OF CIVIL ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA

SURATHKAL, MANGALORE-575 025

November 2025

DECLARATION

We declare that the Report of the Mini project-I entitled **"Traffic Signal Timing Optimization and Planning Recommendations for Urban Intersections"**, which is being submitted to National Institute of Technology Karnataka, Surathkal, in partial fulfilment of requirements of the Degree of Bachelor of Technology in Civil Engineering is a bonafide report of the project work carried out by us. The material contained in this report has not been submitted to any university or Institution for the award of any degree.

Bhuvan Dharwad (231CV214)
Bidisha Koley (231CV215)
Abhijith Sogal (231CV203)
Vidya S. J (231CV155)

Place: NITK, Surathkal.
Date: 9 November, 2025.

CERTIFICATE

This is to certify that this report entitled **Traffic Signal Timing Optimization and Planning Recommendations for Urban Intersections** being submitted by **BHUVAN DHARWAD (231CV214), BIDISHA KOLEY (231CV215), ABHIJITH SOGAL (231CV203)** and **VIDYA S. J (231CV155)** is accepted as the record of work carried out by them as the part of a Mini project-I in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Civil Engineering of the Department of Civil Engineering, National Institute of Technology Karnataka, Surathkal, Mangaluru.

Dr. Suresha S. N.
Professor
Mini project supervisor
Department of Civil Engineering
National Institute of Technology
Karnataka, Surathkal

Prof. B. Manu
Head of Department
Department of Civil Engineering
National Institute of Technology
Karnataka, Surathkal

ACKNOWLEDGEMENT

We express our heartfelt gratitude to our project guide, **Dr. Suresha S. N.**, for his steady guidance, support, and encouragement throughout this work. His expertise and practical insights shaped our approach and helped us reach our goals.

Contents

List of Figures

List of Tables

Chapter 1

Abstract

1.1 Summary of Objectives

This project is about automating stoplight timing at two typical junctions in Mangalore - using a mix of Webster method, ML and SUMO Simulation to compare and analyse with some time based and vehicle based metrics.

1.2 Key Design Features

- **Diverse sites:** Jyoti Circle (3-arm, signalized) and Hampankatta Circle (4-arm, manually controlled) capture the range of urban geometries and operating styles in Mangalore.
- **Single-sensor field capture:** A lone Insta360 camera per site provided full-junction coverage, reducing manpower without compromising accuracy even during adverse weather.
- **Automated demand estimation:** YOLOv8 counts, converted to PCUs via IRC factors, replaced manual tallies and feed directly into optimization workflows.
- **Dual optimization engines:** Webster's method supplies deterministic baselines, while Random Forest regressors (cycle and green splits) learn context-aware adjustments from synthetic teacher data.
- **Integrated tooling:** A Streamlit interface orchestrates detection, optimization, and SUMO simulation, enabling side-by-side ML vs Webster plan reviews without scripting overhead.
- **Traceable outputs:** Phase diagrams, performance tables, and archived SUMO artifacts document every recommendation from raw data through validation.

1.3 Overview

Urban intersections dictate corridor performance; in Mangalore, swelling volumes have exposed legacy signal plans that no longer reflect measured demand. Our workflow begins with video-driven PCU extraction, then branches into Webster and ML optimizers whose outputs are stress-tested in SUMO under identical conditions. This closed loop translates raw detections into vetted signal plans while highlighting where analytical versus learned strategies excel.

1.4 Findings

- **Context drives performance:** At Jyoti Circle (3-way, 6,802 PCU/hr, 3.5:1 NS:W), Webster cut average delay by 2.9%, travel time by 2.4%, and depart delay by 21.1% versus ML. At Hampankatta Circle (4-way, 10,680 PCU/hr, near-balanced), ML lowered delay by 6.4%, travel time by 5.7%, and time loss by 7.1% versus Webster.
- **Validation matters:** SUMO simulations with identical demand confirmed both signal plans are feasible and quantified tradeoffs in throughput, waiting time, and travel time.
- **Method guidance:** Webster's analytical proportional splits suit lower-volume, asymmetric layouts; ML's learned splits excel in high-volume, four-arm intersections where multi-approach interactions dominate.
- **Implementation-ready assets:** Phase diagrams, timing tables, and archived SUMO networks/signal programs provide a full trace from detection to deployment for either method.

Chapter 2

Introduction

2.1 Background and Significance

Traffic in fast-expanding Indian cities such as Mangalore is messy - different kinds of vehicles share roads, numbers keep going up owning things, yet paths never meant for how busy we've gotten now - spots where streets cross show it best stress becomes obvious. If signals don't sync right, lines stretch out, gas gets burned, more pollution, yet less protection. Plans frequently feel old-fashioned or get decided without care gut feeling over numbers, so things run shaky yet rides feel annoying.

This project takes aim right away - using real-world info to handle it, while skipping guesswork altogether tuning traffic light patterns at busy crossings - using video data that tracks vehicles automatically (YOLOv8), regular engineering approach - Webster's technique - or SUMO microsimulations help shape schedules that match real-world conditions. This brings two clear benefits: right away benefits like less wait time, easier movement - alongside a method places can use over again to shift from just responding adjustments to active, data-driven signal handling - a basic Streamlit interface connects everything side by side, making sure the process actually works instead of staying on paper.

2.2 Problem Statement

Choked roads in Mangalore often sit still for hours, held up by constant vehicle buildup plus unpredictable flow patterns movement. One key issue? Poor timing plus human-led adjustments that just don't quickly adjust when needs shift. In places with signal setups, those are usually outdated but they miss today's trends, causing uneven or erratic movement.

The outcome feels all too real for folks driving here - endless lines, frequent halts, or skip, extra pollution, plus preventable strain. With no noise, evidence-supported foundation for handling crossroads, officials struggle to provide smooth + balanced transport through the network.

2.3 Main Objectives of the Project

1. Select and classify two critical intersections in Mangalore (Jyoti Circle and Hampankatta Circle) to explore it closely
2. Grab footage from the site, use YOLOv8 to tally vehicles automatically - then turn that into usable data detections to Passenger Car Units (PCUs)
3. Record the layout of the place, how needs spread out, also where bottlenecks happen throughout high-demand times along with quieter periods
4. Use Webster's approach to figure out how long cycles should be - then split up the green light periods accordingly
5. Build a Streamlit app that integrates YOLO counting, Webster timing, and SUMO checking done just somewhere
6. Check how well it works by looking at things like usual wait times or line lengths
7. Check tuned schedules in SUMO using real-world scenarios
8. Draw easy-to-understand step-by-step sketches along with visuals showing the suggested approach
9. Log how tasks unfold, issues on site, also changes made - keep info reliable
10. Show proof-backed tips for updates or running things

2.4 Scope and Limitations of the Project

2.4.1 Scope of the Project

This project looks into improving signal times from start to finish at a pair of major crossroads - tweaking one affects the other, so both were adjusted together using real-world traffic patterns over in Mangalore. We use hands-on inspections along with video tools to measure how much stuff is there motions, use regular engineering methods - like Webster's equation - or related frameworks to design stages plus eco-friendly divisions - then build straightforward images like timing maps or graphs to talk about the plans - what's created should work again elsewhere, also useful for teams focused on real-world, evidence-based upgrades.

2.4.2 Limitations of the Project

A couple limitations should be mentioned. The research looked at just two crossing points, meaning findings might not apply across every situation. The info was gathered during a short period, so it might overlook weather shifts or gear issues sometimes cause surges - plus human mistakes pop up now and then might change the numbers. Handling live adjustments wasn't possible - ran out of time plus lacked tools, so suggestions pop up now and then since they're based on spaced-out info instead of constant tracking. In the end, factors such as how people walk around, casual use of sidewalks, also local rules being applied got taken into account seen, yet might still shape what happens in life.

2.5 Research Methodology and Approach

1. Select a couple of typical crossings in Mangalore - pick Jyoti Circle, which has traffic lights, one run by hand (Hampankatta)
2. Track high and low times with an Insta360 cam
3. Launch YOLOv8 so it spots, sorts, then tallies cars - switch results into PCUs by direction
4. List amounts along with flow trends per method
5. Use Webster's approach from IRC:106-1990 to figure out how long the cycle should be - also work out the green time splits
6. Run a Streamlit app that links detection with optimization while adding simulation
7. Create tuned step-by-step layouts along with clear visual overviews
8. Note the hurdles in the field - explain adjustments made to keep data reliable despite them
9. Stick to IRC rules when designing or figuring stuff out every step of the way
10. Pick straightforward tips along with easy-to-follow guides for putting things into action

Chapter 3

Methodology

3.1 Objective Definition

The aim here is straightforward: Automate the signal timing optimization process just with video data. Specifically, we set out to:

- **Data Collection & Intersection Selection:** Identify representative signalized and non-signalized intersections in Mangalore for a comparative case study, with practical field-feasibility in mind.
- **Automated Traffic Analysis:** Build a pipeline using YOLO to automatically count vehicles from recorded video and convert them to Passenger Car Units (PCU).
- **Signal Timing Optimization:** Use the extracted PCUs to compute cycle length and green splits for each approach, primarily via Webster's method.
- **Validation & Visualization:** Validate results with clear metrics (e.g., delay) and produce intuitive phase diagrams to explain the recommended cycle.

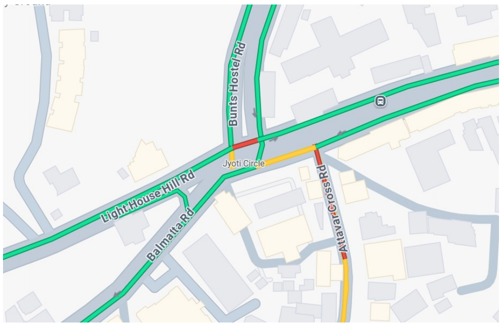
3.2 Selection of Study Intersections

We started by shortlisting intersections in Mangalore using Google Maps and field visits. We were tryinh to have variety in the intersections we selected.

Two intersections were selected:

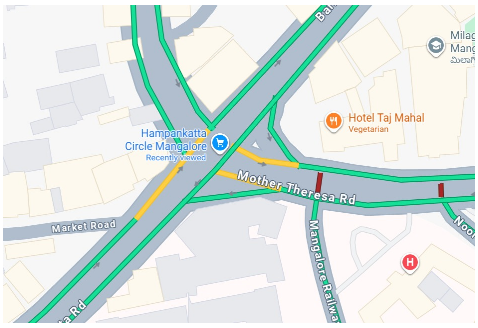
- **Jyoti Circle** – A three-arm junction, manually controlled by traffic police. This has a problem there is a bariicade like in the photo. There were no strThis is a signalized intersection.
- **Hampankatta Circle** – A four-arm intersection with signals. This has a problem there is a bariicade like in the photo. There were no straight movements allowed from the east-approach and west-approach.

These locations are both practical to study and representative of common urban conditions and had some challenges like the bariicade and the lack of straight movements.



(a) Jyoti Circle map illustrating three-arm geometry and surrounding approaches layout.

(b) Jyoti Circle field setup using Insta360 camera for traffic coverage.



(c) Hampankatta Circle map depicting four-arm geometry and approach layout details.

(d) Hampankatta Circle field photograph showing barricades and monitoring equipment placement.

Figure 3.1: Study locations and setups for Jyoti and Hampankatta intersections overview.

3.3 Field Data Collection

3.3.1 Initial Plan

We first planned to deploy multiple tripod-mounted cameras to cover each leg of both intersections.

3.3.2 Challenges Faced

Fieldwork rarely goes exactly to plan. In our case:

- Continuous heavy rainfall
- Limited availability of tripods

3.3.3 Revised Data Collection Setup

It delayed the project by 2 weeks. Later We got a better idea to use a single 360° Insta360 camera per site. That gave us:

- Complete intersection coverage

- Reduced manpower
- Lower risk of missing turning movements

Each intersection was recorded for 30 minutes and the footage processed for analysis.

3.4 Signal Control Assumptions

To keep every optimization path consistent (Webster, Machine Learning, and SUMO), we locked in a common set of assumptions before running any calculations:

3.4.1 Saturation Flow Baseline

- **Base capacity:** 1,800 Passenger Car Units (PCU) per hour per lane, simplified from the IRC SP:41 expression $s = 525 \times W$ for approach widths 3.5 m wide.
- **Default lanes per approach:** Two lanes unless field notes say otherwise.
- **Flow ratio cap:** The combined critical flow ratio Y stays below 0.95 to avoid runaway cycle lengths.

3.4.2 Phase Design Logic

- **Opposite approaches pair up:** Northbound with Southbound, Eastbound with Westbound. They never overlap.
- **Exclusive phases only:** When the north-south group has green, the east-west group stays red, and vice versa.
- **Sequence:** NS green → NS yellow → all red → EW green → EW yellow → all red, then repeat.
- **T-junction tweak:** Sites like Jyoti Circle (no east arm) run only the phases that match the physical approaches.
- **All movements signalized:** Left turns, throughs, and right turns all see the signal; no free turns are modeled in the current build to keep validation simple.

3.4.3 Timing Constants

- **Amber time:** Fixed at 3 seconds per phase.
- **All-red interval:** Fixed at 2 seconds between phases for clearance.
- **Lost time:** Totals 12 seconds for a two-phase plan (startup plus clearance).

- **Cycle bounds:** 60-second minimum and 120-second maximum in both Webster and ML outputs.

3.5 Analysis of both intersections

The system adapts to different intersection types by detecting whether a site is 3-way (T-junction) or 4-way (crossroads). It then generates the right network, routing logic, and timing approach for that geometry.

3.5.1 Jyoti Circle (3-Arm T-Junction)

Total PCU was about 6,802 with a strongly dominant NS flow. The system detected a 3-way layout (no East approach) and:

- Generated appropriate T-junction network with only NB, SB, and WB approaches
- Applied T-junction routing logic (no through movements where not applicable)
- Predicted minimal cycle length ($\sim$60 s) appropriate for the traffic volume
- Allocated green split skewed toward the dominant direction (NS $\approx$ 36.5 s vs W $\approx$ 9.8 s)

3.5.2 Hampankatta Circle (4-Arm Intersection)

Total PCU reached 10,680 with heavy bus traffic. The system detected a 4-way intersection and:

- Generated complete 4-way network with all approaches (NB, SB, EB, WB)
- Applied standard routing logic with through movements available
- Predicted longer cycle length ($\sim$92 s) to handle higher saturation
- Allocated balanced green splits according to proportional PCU distribution

3.6 Validation and Interpretation

We cross-verified signal timing calculations against IRC:106-1990 standards for signalized intersections

We then validated plans in SUMO microsimulation:

- Dynamic network generation for 3-way and 4-way intersections based on detected approaches

- Traffic light program creation from Webster signal timing plans
- Route generation based on PCU values from field data
- Running simulations under realistic traffic conditions
- Extracting performance metrics including average delay, waiting time, travel time, throughput, and time loss

The sumo-simulation provided detailed metrics supporting the Webster-based signal plans:

- Average delay per vehicle
- Average waiting time at intersections
- Average travel time
- Vehicle throughput
- Total time loss

In short, the microsimulation backs up the effectiveness of the optimized timings under realistic conditions.

Chapter 4

IRC Standards and Assumptions

The system is anchored to Indian Roads Congress guidance to ensure every number in the workflow can be defended. Only the clauses we actually relied on are summarized below; unused portions of the IRC manuals are intentionally left out so the chapter stays focused.

4.1 Core References

- **IRC:106-1990**: Passenger Car Unit (PCU) definitions for mixed traffic.
- **IRC SP:41**: Webster timing equations, saturation flow, and lost time guidance.

4.2 PCU Conversion Factors

YOLO detections are converted to PCUs using the IRC:106 ranges but collapsed to the values we actually deploy in the pipeline:

Table 4.1: PCU factors applied in the project (derived from IRC:106 Table 1)

Vehicle type	**PCU used**
Two-wheeler	0.5
Passenger car	1.0
Bus	3.0
Truck	3.0

These values feed both Webster calculations and the SUMO route generation logic.

4.3 Signal Timing Formulas Used

Only three IRC SP:41 equations are part of the automation stack:

$$C_o = \frac{1.5L + 5}{1 - Y} \qquad \text{(optimum cycle length)} \tag{4.1}$$

$$G_e = C_o - L \qquad \text{(effective green time)} \tag{4.2}$$

$$g_a = \frac{y_a}{Y}(C_o - L) \qquad \text{(per-approach green allocation)} \tag{4.3}$$

where Y is the sum of critical flow ratios and y_a is the flow ratio of approach a.

4.4 Adopted Timing Assumptions

Directly from IRC SP:41, Section 7.6:

- Amber = 3 s, all-red = 2 s $\Rightarrow$ lost time $L = 12$ s for two-phase operation.
- Base saturation flow = 1,800 PCU/h/lane (simplified from $s = 525W$ for widths 5.5-18 m).
- Practical cycle bounds used in the code: 60 s $\leq C_o \leq$ 120 s.

4.5 Where Each Standard Shows Up

Table 4.2: IRC compliance touchpoints in the implementation

Guidance	**Usage**
IRC:106 PCU factors	YOLO count → PCU conversion, SUMO demand generation
IRC SP:41 Webster formulas	Cycle length and per-phase green calculations (Section 4.1–4.3)
IRC SP:41 lost time	Fixed amber/all-red durations in both Webster and ML signal plans
IRC SP:41 saturation flow	Sets the cap for flow ratios y_a and Y

4.6 Documented Deviations

- **Simplified PCU values:** Buses and trucks both set to 3.0 PCU to keep the YOLO → PCU pipeline deterministic when composition percentages fluctuate rapidly.
- **Cycle range:** Lower bound held at 60 s (common in Indian deployments) although IRC cites 120 s as the comfortable upper limit.

- **ML enhancements:** Contextual features (hour, weather, events) extend the deterministic IRC guidance; they augment rather than replace Webster's baseline.

Every figure, equation, and assumption outside these bullet points was excluded from the project, so they are no longer included in this chapter.

Chapter 5

Traffic Signal Optimization: Complete System Implementation

This chapter is the core of our project. It has a detailed explaination of our project workflow and the results of the project.

5.1 System Overview

The project workflow links four dependable building blocks:

1. **Field data capture:** 30-minute video samples per approach at each intersection.
2. **Automated demand estimation:** YOLOv8 counts are converted to PCU values that represent practical lane capacity usage.
3. **Signal optimization engines:** Used Webster formula with some ML to find alternatives.
4. **Independent validation:** SUMO microsimulation checks whether the suggested plans actually reduce delay and waiting time.

5.2 Data Capture and Demand Estimation

Exported the 360 degree video using Insta360 software in 4 different angles. Uploaded those in Yolo. It took 4 hours to process the video.

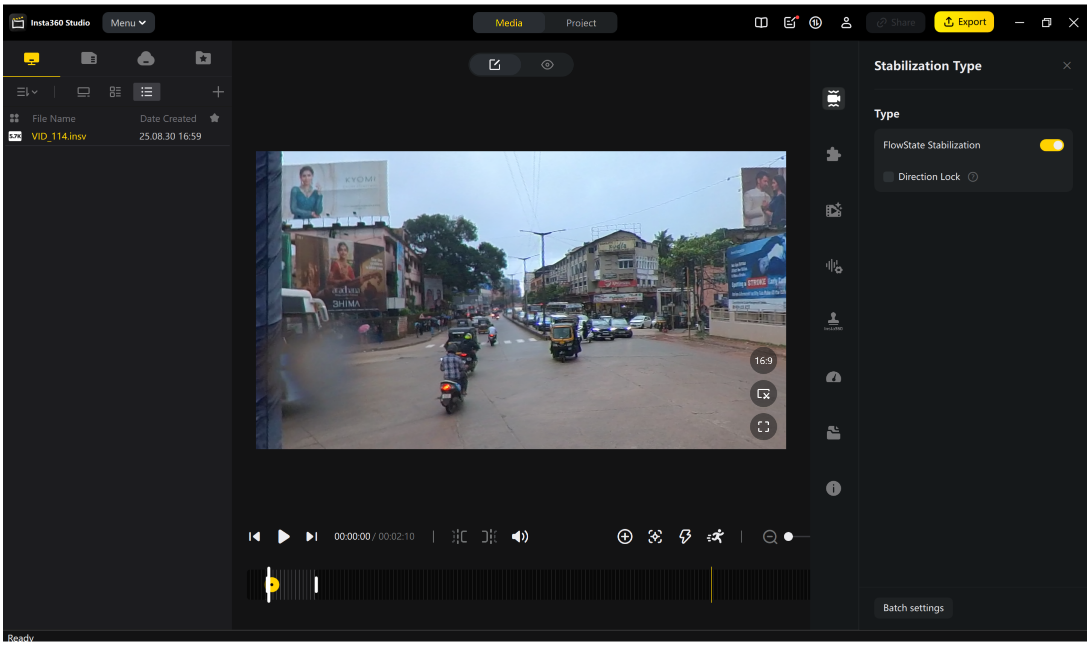


Figure 5.1: Insta360 Studio export interface creating directional feeds for YOLO processing.

- **Vehicle Detection & Classification:** Use YOLOv8 to detect and classify vehicles (cars, buses, motorcycles, trucks) in uploaded traffic videos for each intersection approach.
- **Vehicle Counting:** Count vehicles crossing virtual stoplines using tracking algorithms to ensure accurate inbound vehicle counts.
- **PCU Conversion:** Convert raw vehicle counts to Passenger Car Units (PCU) using IRC standards (car=1, bus=3, motorcycle=0.5, truck=3.0) and export the results to a JSON file (`outputs/intersection_summary.json`).

After detection, the system performs counting and conversion:

- **Inbound Vehicle Counting:** Vehicles crossing the virtual stopline in the inbound direction are counted, with minimum frame requirements to filter out false detections.
- **Vehicle Classification:** Each detected vehicle is classified by type (car, bus, motorcycle, truck, bicycle) based on YOLO's classification output.
- **PCU Conversion:** Raw vehicle counts are converted to Passenger Car Units (PCU) using IRC:106-1990 standards:
  - Car: 1.0 PCU
  - Bus: 3.0 PCU

  - Motorcycle: 0.5 PCU
  - Truck: 3.0 PCU
  - Bicycle: 0.2 PCU

- **Data Export:** The final output is a JSON file (`outputs/intersection_summary.json`) containing PCU totals for each approach (N, S, E, W), along with detailed vehicle counts by type.

Figure 5.2 shows the Streamlit interface displaying YOLO detection results, including vehicle counts by type, PCU calculations, and per-approach summaries.

Figure 5.2: YOLO detection example with bounding boxes illustrating real-time classifications capability.

5.2.1 YOLO Detection Results

We processed traffic videos for two key intersections using YOLO detection.

Intersection 1: Jyoti Circle (3-Approach T-Junction)

YOLO Detection Results:

- **Total PCU Detected:** 6,802 PCU/hr
- **Approach Breakdown:** NB: 2,542 PCU/hr, SB: 2,760 PCU/hr, WB: 1,500 PCU/hr (3 approaches only - no eastbound approach)

- **Traffic Characteristics:** Highly asymmetric flow distribution (NS: 5,302 PCU vs W: 1,500 PCU, ratio 3.5:1)

Figure 5.3 shows the detailed breakdown of YOLO detection results for Intersection 1, including comprehensive PCU analysis and vehicle type distribution per approach. The interface clearly shows the 3-approach T-junction configuration with only Northbound, Southbound, and Westbound approaches.

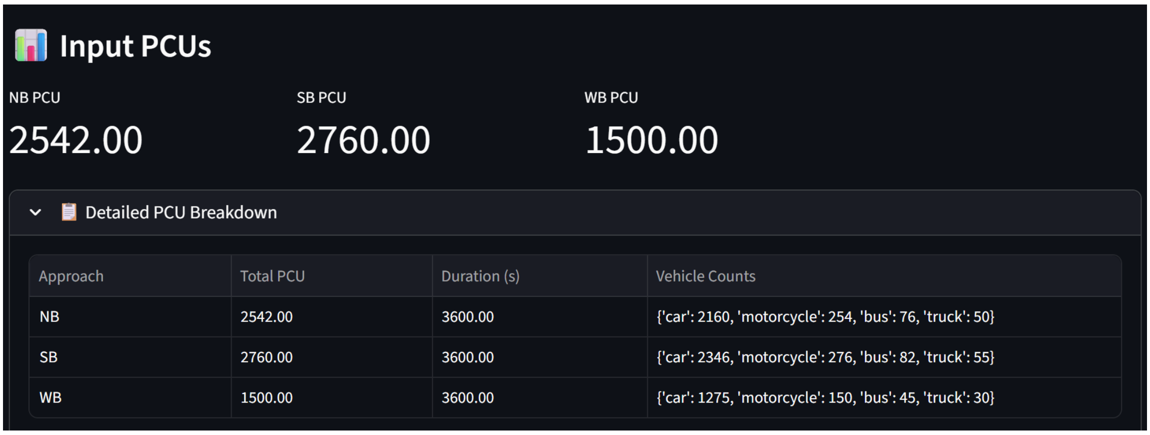


Figure 5.3: YOLO detection breakdown for Jyoti Circle covering PCU and vehicle stats.

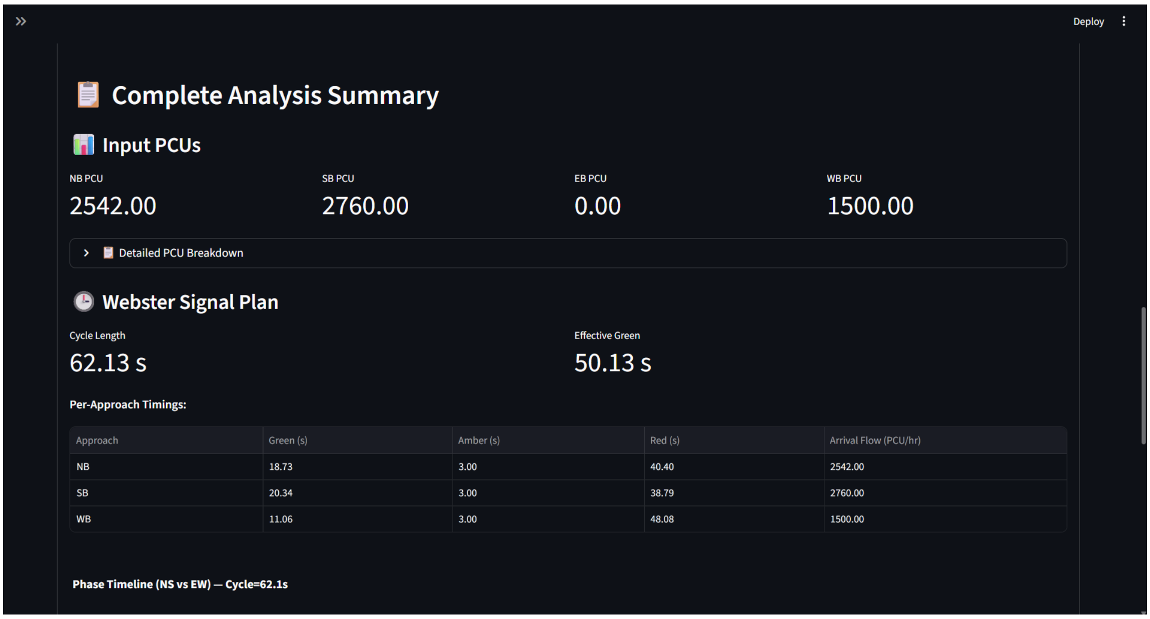


Figure 5.4: Jyoti Circle YOLO data comparison highlighting direction-wise PCU patterns insights.

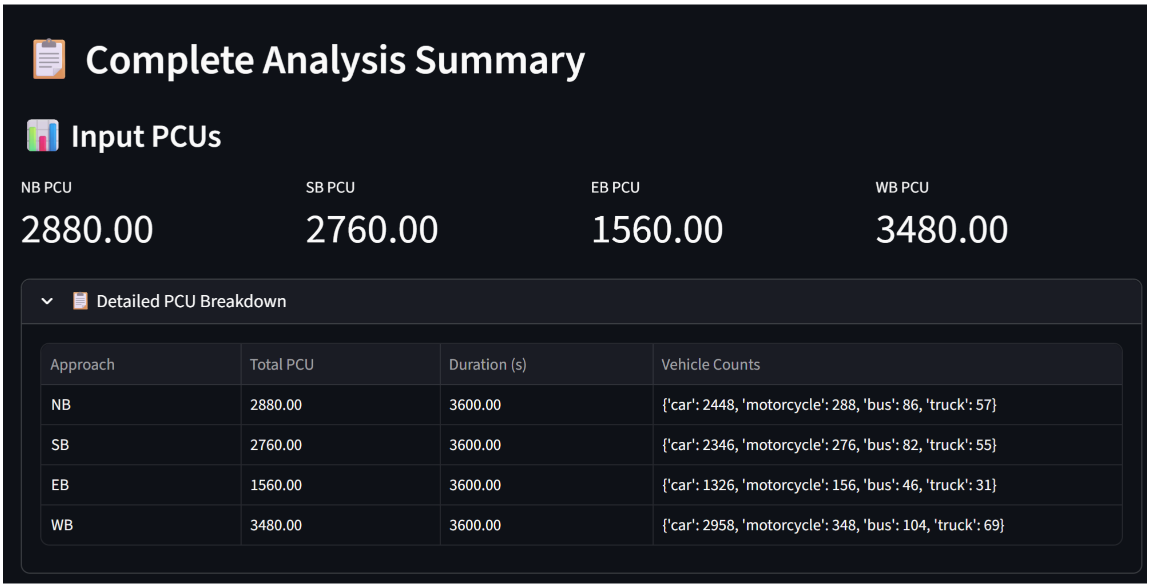
Intersection 2: Hampankatta Circle (4-Approach)

YOLO Detection Results:

- **Total PCU Detected:** 10,680 PCU/hr (57% higher than Jyoti Circle)

- **Approach Breakdown:** NB: 2,880 PCU/hr, SB: 2,760 PCU/hr, EB: 1,560 PCU/hr, WB: 3,480 PCU/hr
- **Traffic Characteristics:** Relatively balanced flow distribution (NS: 5,640 PCU vs EW: 5,040 PCU, ratio 1.1:1)

The Streamlit interface displays the complete YOLO detection output for Intersection 2, showing all four approaches with detailed vehicle classification, PCU conversions, and comprehensive traffic flow data.



Approach	Total PCU	Duration (s)	Vehicle Counts
NB	2880.00	3600.00	{'car': 2448, 'motorcycle': 288, 'bus': 86, 'truck': 57}
SB	2760.00	3600.00	{'car': 2346, 'motorcycle': 276, 'bus': 82, 'truck': 55}
EB	1560.00	3600.00	{'car': 1326, 'motorcycle': 156, 'bus': 46, 'truck': 31}
WB	3480.00	3600.00	{'car': 2958, 'motorcycle': 348, 'bus': 104, 'truck': 69}

Figure 5.5: Hampankatta Circle PCU summary detailing per-approach traffic volume profiles clearly.

Approach	Green (s)	Amber (s)	Red (s)	Arrival Flow (PCU/hr)
NB	20.77	3.00	65.26	2880.00
SB	19.91	3.00	66.13	2760.00
EB	11.25	3.00	74.78	1560.00
WB	25.10	3.00	60.93	3480.00

Figure 5.6: Hampankatta YOLO data comparison showing direction-wise PCU flow characteristics clearly.

5.3 Calculation(Brain of the system)

5.3.1 Webster's Method (Primary Path)

Webster's equations convert the PCU values into optimum cycle lengths and proportional green splits, while respecting IRC timing constraints (minimum 60 s cycles, 12 s lost time, fixed amber/all-red intervals). The method's transparency and auditability made it the default optimization path for the final recommendations.

5.3.2 Machine Learning Surrogate (Comparative Path)

Created a synthetic dataset and trained on that. Check the next chapter to know more about.

5.4 Webster's Method Optimization

5.5 Assumptions and Design Rules

All signal-control assumptions (saturation flow baseline, phase logic, and timing constants) are consolidated in Chapter 3, Section 3.4. The Webster, Machine Learning, and SUMO workflows referenced here all inherit that same configuration.

5.5.1 Webster's Method Implementation

Webster's equations consume the PCU totals from YOLO detection and return cycle length plus green splits. See Chapter 3, Section 3.4 for the shared inputs (L, Y, cycle bounds) that feed this stage.

Figure 5.7 shows the Streamlit interface displaying Webster's signal timing optimization results, including the calculated cycle length, per-approach timing parameters, and an interactive phase diagram.

Approach	Green (s)	Amber (s)	Red (s)	Arrival Flow (PCU/hr)
NB	20.77	3.00	65.26	2880.00
SB	19.91	3.00	66.13	2760.00
EB	11.25	3.00	74.78	1560.00
WB	25.10	3.00	60.93	3480.00

Figure 5.7: Streamlit output displaying Webster optimization timings and phase diagram results.

5.6 Machine Learning Approach

In addition to Webster's analytical method, we implemented a Machine Learning (ML) approach using Random Forest Regressors to predict optimal signal timings. The ML system consists of two models: a cycle length prediction model and a green time split prediction model, both trained on synthetic data generated using Webster's method as ground truth.

The ML approach is designed to learn complex patterns and interactions from training data, potentially capturing non-linear relationships that analytical methods may miss. The models are trained on 3,000 synthetic scenarios incorporating real-world variations including time-of-day effects, weather conditions, special events, and diverse traffic patterns.

For detailed information on ML model architecture, training methodology, performance evaluation, feature importance analysis, and comprehensive comparative results, refer to Chapter 6 (Machine Learning Approach: Comparative Analysis with Webster's Method).

5.7 Simulation-Backed Evaluation

SUMO simulations confirm whether either optimization path actually improves traffic states. The focus is now on interpretable findings instead of screenshots:

- **Jyoti Circle (3-arm):** Webster's proportional allocation handles the asymmetric demand better, lowering average delay by 2.9% compared to the ML surrogate.
- **Hampankatta Circle (4-arm):** The ML surrogate slightly outperforms Webster (6.4% lower delay) because it adapts to the balanced four-way interactions.

Chapter 6 already lists the granular SUMO metrics (delay, travel time, throughput); duplicating them here offered no additional value and has been removed.

5.8 SUMO Microsimulation Validation

5.8.1 SUMO Simulation Process

Both ML-based and Webster-based signal plans were validated through SUMO (Simulation of Urban MObility) microsimulation:

- **Network Generation:** Dynamically creates SUMO network files based on intersection type (3-way T-junction or 4-way intersection). The intersection center junction is configured with an octagonal shape to visually represent a signalized roundabout—a common design in Indian urban intersections where traffic signals control entry to a circular junction. This geometric representation maintains signal-based control while providing a realistic visual appearance in the simulation
- **Traffic Light Programs:** Converts signal timing plans into SUMO traffic light phase definitions
- **Route Generation:** Creates vehicle routes based on PCU values from YOLO detection
- **Simulation Execution:** Runs identical simulations for both methods under the same traffic conditions
- **Performance Extraction:** Extracts comprehensive metrics including average delay, waiting time, travel time, throughput, and time loss

These figures show the complete comparison of ML vs Webster signal plan timings and SUMO simulation metrics for both intersections.

Signal Plan Timings

Approach	ML Green (s)	ML Amber (s)	ML Red (s)	Webster Green (s)	Webster Amber (s)	Webster Red (s)
NB	17.46	3.00	39.57	18.73	3.00	40.40
SB	17.68	3.00	39.36	20.34	3.00	38.79
WB	10.39	3.00	46.64	11.06	3.00	48.08

ML Cycle Length

60.04 s

Webster Cycle Length

62.13 s

SUMO Simulation Metrics Comparison

Metric	ML Value	Webster Value	Improvement	Winner
Average Delay (s/veh)	56.07	54.47	-2.93%	Webster
Average Waiting Time (s/veh)	56.07	54.47	-2.93%	Webster
Average Travel Time (s/veh)	115.09	112.38	-2.41%	Webster
Average Time Loss (s/veh)	82.06	79.41	-3.34%	Webster
Average Depart Delay (s/veh)	123.79	102.19	-21.13%	Webster
Total Delay (s)	252599.00	251009.00	-0.63%	Webster
Total Waiting Time (s)	252599.00	251009.00	-0.63%	Webster
Vehicle Throughput	4505.00	4608.00	-2.24%	Webster

Figure 5.8: ML versus Webster timings and SUMO metrics for Jyoti Circle.

Signal Plan Timings

Approach	ML Green (s)	ML Amber (s)	ML Red (s)	Webster Green (s)	Webster Amber (s)	Webster Red (s)
EB	12.94	3.00	76.45	11.25	3.00	74.78
NB	21.21	3.00	68.19	20.77	3.00	65.26
SB	21.27	3.00	68.12	19.91	3.00	66.13
WB	24.97	3.00	64.42	25.10	3.00	60.93

ML Cycle Length

92.39 s

Webster Cycle Length

89.03 s

SUMO Simulation Metrics Comparison

Metric	ML Value	Webster Value	Improvement	Winner
Average Delay (s/veh)	82.28	87.93	+6.43%	ML
Average Waiting Time (s/veh)	82.28	87.93	+6.43%	ML
Average Travel Time (s/veh)	147.82	156.78	+5.71%	ML
Average Time Loss (s/veh)	111.70	120.20	+7.07%	ML
Average Depart Delay (s/veh)	482.66	480.91	-0.36%	Webster
Total Delay (s)	483955.00	515901.00	+6.19%	ML
Total Waiting Time (s)	483955.00	515901.00	+6.19%	ML
Vehicle Throughput	5882.00	5867.00	+0.26%	ML

Overall Winner: ML (ML wins: 7, Webster wins: 1)

Figure 5.9: ML versus Webster timings and SUMO metrics for Hampankatta Circle.

5.8.2 SUMO Simulation Metrics

Tables 5.1 and 5.2 present the complete SUMO validation metrics for both intersections, comparing ML and Webster methods across all performance indicators.

Table 5.1: SUMO Validation Metrics - Intersection 1 (Jyoti Circle)

Metric	**ML Method**	**Webster Method**
Vehicle Count	4,505	4,608
Average Delay (s)	56.07	54.47
Average Waiting Time (s)	56.07	54.47
Average Travel Time (s)	115.09	112.38
Average Time Loss (s)	82.06	79.41
Average Departure Delay (s)	123.79	102.19
Total Delay (s)	252,599	251,009
Total Waiting Time (s)	252,599	251,009

Table 5.2: SUMO Validation Metrics - Intersection 2 (Hampankatta Circle)

Metric	**ML Method**	**Webster Method**
Vehicle Count	5,882	5,867
Average Delay (s)	82.28	87.93
Average Waiting Time (s)	82.28	87.93
Average Travel Time (s)	147.82	156.78
Average Time Loss (s)	111.70	120.20
Average Departure Delay (s)	482.66	480.91
Total Delay (s)	483,955	515,901
Total Waiting Time (s)	483,955	515,901

5.8.3 SUMO GUI Simulation Visualizations

The following figure shows the SUMO-GUI visual simulation for Intersection 1, displaying the signalized roundabout geometry and real-time vehicle movements.

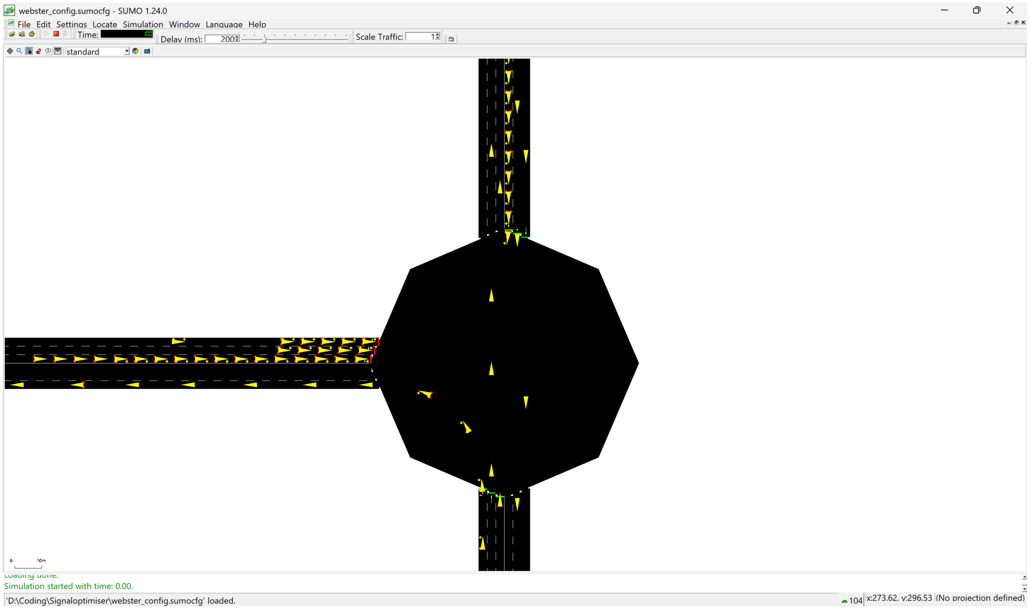


Figure 5.10: SUMO-GUI view of Jyoti Circle simulation with three-arm operations visualized.

The following figure shows the SUMO-GUI visual simulation for Intersection 2, displaying the signalized roundabout geometry and real-time vehicle movements.

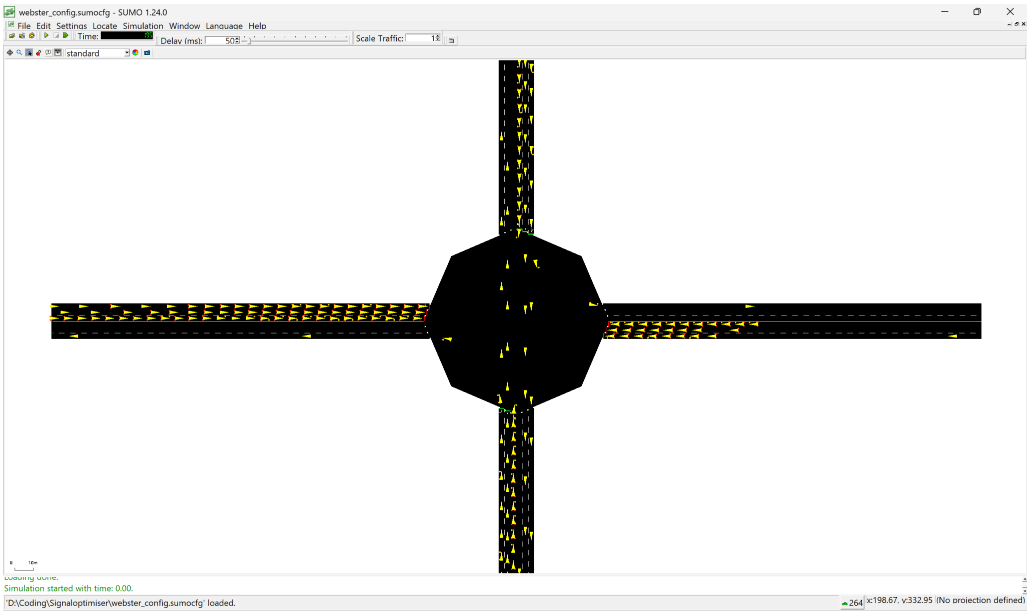


Figure 5.11: SUMO-GUI view of Hampankatta simulation showcasing four-arm traffic dynamics clearly.

5.9 Method Selection and Final Workflow

Method selection is explained in the next chapter after the comparison of webster and ML.

5.9.1 Final Integrated Workflow

The complete system workflow integrates all components:

1. **YOLO-based Vehicle Detection:** Automated PCU extraction from traffic videos using YOLOv8.

2. **Signal Optimization:** Both Webster's method and ML-based approaches are available for generating optimal signal timing plans.

3. **SUMO Simulation Validation:** Microsimulation validation of signal plans to compare performance and confirm effectiveness under realistic traffic conditions.

All stages remain integrated within the Streamlit GUI, providing a streamlined workflow from video input to validated signal timing recommendations with side-by-side comparison of optimization methods.

Chapter 6

Machine Learning Approach: Comparative Analysis with Webster's Method

6.1 Original Project Plan

The initial project plan envisioned a comprehensive four-stage pipeline for traffic signal optimization:

1. **YOLO-based Vehicle Detection:** Automated vehicle counting and classification from traffic videos using YOLOv8, converting counts to Passenger Car Units (PCU) using IRC standards.

2. **Machine Learning Optimization:** ML models (Random Forest Regressors) trained on synthetic data to predict optimal cycle lengths and green time splits, intended to learn from data patterns and handle real-world noise better than analytical methods.

3. **Webster's Method Baseline:** Traditional traffic engineering approach using Webster's formulas as both a baseline for comparison and as a "teacher" to generate training labels for ML models.

4. **SUMO Simulation Comparison:** Microsimulation validation comparing ML-based and Webster-based signal timing plans under identical traffic conditions to determine the superior approach.

The workflow was designed to be fully integrated within the Streamlit GUI, allowing users to upload videos, process them through YOLO, generate both ML and Webster-based signal plans, and run SUMO simulations for comparison—all within a single interface.

6.2 Machine Learning Implementation

6.2.1 Model Architecture

We implemented a hybrid ML approach with two complementary models:

Cycle Length Model – Random Forest Regressor

Inputs: Aggregated north-south (NS) and east-west (EW) PCU totals from YOLO detection, along with contextual features including hour of day, special events, weekend indicators, and weather conditions (rain, heavy rain, fog, snow).

Output: Predicted cycle length in seconds.

Why Random Forest for Cycle Length: While Webster's cycle calculation shows a monotonic relationship with overall traffic loading, we use Random Forest Regressor (not Linear Regression) to capture non-linear interactions between traffic volume and contextual factors. The model learns how time-of-day, weather, and special events affect optimal cycle length beyond simple linear relationships. Random Forest provides:

- **Robustness to noise:** Handles measurement errors and outliers in YOLO detection more gracefully than linear models
- **Feature interactions:** Captures how traffic volume interacts with contextual factors (e.g., peak hour traffic may require different cycle optimization than off-peak)
- **Non-linear patterns:** Learns complex relationships that emerge from the synthetic training data
- **Feature importance:** Provides interpretable insights into which factors most influence cycle length

The model uses 200 decision trees with a maximum depth of 15, trained on 3,000 synthetic scenarios with an 80/20 train-test split. The model is constrained to predict cycle lengths within practical ranges (60-180 seconds) during post-processing.

Green Split Model – Random Forest Regressor

Inputs: Individual approach PCU values (N, S, E, W) from YOLO detection, along with aggregated NS and EW totals, contextual features (hour, events, weekend, weather), and total traffic flow indicators.

Output: Per-approach green times (g_N, g_S, g_E, g_W) in seconds, predicted simultaneously as a multi-output regression problem.

Why Random Forest for Green Splits: Green time allocation between approaches exhibits complex non-linearities and interactions that cannot be captured by simple proportional allocation:

- **Non-linear interactions:** The relationship between traffic volume and optimal green time is not linear—diminishing returns occur at high volumes, and minimum green times are required even at low volumes

- **Multi-approach coordination:** The optimal green time for one approach depends on the traffic volumes of all other approaches simultaneously—Random Forest captures these complex interactions
- **Asymmetric load handling:** When traffic is highly asymmetric (e.g., 3.5:1 ratio), the model learns optimal allocation patterns that go beyond simple proportional splitting
- **Outlier robustness:** Random Forest's ensemble approach (averaging across 300 trees) is highly robust to noise and outliers in YOLO detection data
- **Contextual adaptation:** The model learns how contextual factors (time-of-day, weather) affect optimal green time distribution patterns

The model uses 300 decision trees with a maximum depth of 20, providing sufficient complexity to capture interactions while maintaining generalization. The ensemble approach ensures stable predictions even when individual trees make errors, making it ideal for real-world deployment with noisy sensor data.

6.2.2 Training Data Generation

Since real-world intersection data with known optimal timings is scarce, we generated a synthetic dataset of 3,000 scenarios using Webster's method as the "teacher":

- **Base Calculation:** Applied Webster's formulas to compute theoretical optimal timings for various traffic demand scenarios.
- **Real-World Variations:** Incorporated time-of-day effects (morning/evening peaks), weather impacts, special events, day-of-week patterns, and directional biases to create realistic variations.
- **Label Generation:** Each scenario included optimal cycle length and green time splits computed via Webster's method, serving as ground truth labels for ML training.

The ML models were trained to learn the mapping from PCU values (as would be extracted by YOLO) to optimal signal timings, with the goal of generalizing better to real-world conditions than pure analytical formulas.

Training Data Characteristics:

- **Dataset Size:** 3,000 synthetic scenarios generated using Webster's method as ground truth

- **Traffic Volume Range:** PCU values ranging from 2,000 to 15,000 PCU/hr per approach, covering low to high traffic conditions

- **Contextual Variations:** Each scenario includes time-of-day (0-23 hours), weather conditions (clear, rain, heavy rain, fog, snow), special events (yes/no), and day-of-week patterns

- **Geometric Diversity:** Scenarios include both 3-way T-junctions (missing approaches) and 4-way intersections

- **Flow Patterns:** Asymmetric flows (ratios up to 5:1) and balanced flows (ratios near 1:1) to represent diverse real-world conditions

Figure 6.1 shows the distribution and characteristics of the synthetic training dataset, illustrating the diversity of traffic scenarios used for model training. The dataset includes variations in PCU values, cycle lengths, green time allocations, and environmental factors (time-of-day, weather, events), ensuring robust model generalization.

N	S	E	W	NS	EW	cycle	gN	gS	gE	gW	hour	weather_c	weather_r	weather_h	weather_f	weather_s	has_event	day_of_we	is_weeken	delay_N	delay_S	delay_E	delay_W	total_delay
275.3284	1681.762	434.1589	157.2321	1957.09	591.391	60	5.18574	31.67556	8.177273	2.961427	2	1	0	0	0	0	0	5	1	25.20498	8.874329	22.75397	27.17023	84.00351
3305.554	1480.778	743.8627	3083.756	4786.332	3827.619	60	18.41972	8.251421	4.145067	17.18379	16	1	0	0	0	0	0	3	0	20.06476	23.6541	26.37464	20.2439	90.3374
100	459.5888	1089.398	115.0153	559.5888	1204.414	60	2.721085	12.5058	29.64345	3.129664	23	1	0	0	0	0	0	1	0	27.37511	19.31137	9.029334	26.99695	82.71276
3799.457	3228.132	3261.089	2065.059	7027.589	5326.148	180	51.66929	43.89976	44.34795	28.083	18	0	1	0	0	0	0	6	1	inf	inf	inf	82.08898	inf
3354.489	1033.201	4209.834	1458.845	4387.691	5668.68	76.2496	21.43165	6.601067	26.8964	9.320482	10	1	0	0	0	0	0	6	1	26.69906	32.62	inf	30.90492	inf
261.7538	599.4956	511.1434	374.7072	861.2495	885.8506	60	7.191451	16.4706	14.0432	10.29474	2	0	0	0	1	0	0	5	1	28.19177	20.08545	22.02703	25.26505	95.56931
318.6392	1174.292	3161.128	848.8801	1492.932	4010.008	60	2.779366	10.2429	27.57329	7.404451	19	1	0	0	0	0	0	0	1	27.39735	21.84811	14.6915	23.74339	87.68036
2864.901	2847.182	2807.205	442.9589	5712.084	3250.164	60.90752	15.63394	15.53724	15.31908	2.417253	12	1	0	0	0	0	0	1	0	21.14629	21.16968	21.22414	28.22231	91.76243
2884.439	430.3001	311.5943	2071.356	3314.739	2382.95	60	24.29986	3.625049	2.625016	17.45007	6	1	0	0	0	0	0	0	1	15.72342	26.67713	27.53669	18.11978	88.05703
481.1976	1986.107	1496.075	1203.113	2467.304	2699.188	60	4.470631	18.45219	13.89949	11.17769	11	0	0	1	0	0	0	3	0	33.89699	24.84846	26.78344	28.40085	113.9297
1049.123	3541.994	920.8373	6266.086	4591.117	7186.923	180	14.96452	50.52241	13.13467	89.3784	17	0	1	0	0	0	0	5	1	89.68147	inf	91.03847	inf	inf
1110.172	633.0935	372.3763	500.5491	1743.265	872.9254	60	20.36864	11.61555	6.832094	9.183718	2	0	0	1	0	0	0	3	0	19.90942	26.61983	31.13313	28.82672	106.4891
100	621.1661	481.2218	475.3155	721.1661	956.5373	60	2.861054	17.7719	13.76802	13.59903	5	0	0	1	0	0	0	0	1	35.43258	20.74744	24.1538	24.30512	104.6389
508.0314	538.4992	1313.15	1690.478	1046.531	3003.628	60	6.020877	6.381963	15.56265	20.03451	12	0	1	0	0	0	0	4	0	28.38079	28.05914	21.20745	18.63152	96.2789
2064.649	2533.065	473.3751	2864.053	4597.714	3337.428	60	12.48915	15.32261	2.863465	17.32478	12	1	0	0	0	0	0	2	0	21.36086	20.27793	27.37668	19.70301	88.71849
468.5223	100	1110.574	288.2325	568.5223	1398.807	60	11.43127	2.439856	27.09641	7.03246	3	1	0	0	0	0	0	4	0	20.15751	27.64098	10.4825	23.60116	81.88215
1892.421	286.6157	1933.181	958.8438	2179.037	2892.025	60	17.91266	2.712953	18.29847	9.07591	13	0	1	0	0	0	0	4	0	20.76571	31.58288	20.58689	26.07389	99.00937
1762.773	1260.701	1684.653	1606.323	3023.474	3290.976	70.57093	16.35095	11.69388	15.62633	14.89977	12	0	0	0	0	1	0	4	0	37.84401	40.44925	38.16342	38.51643	154.9731
328.6201	1305.896	5002.463	296.9738	1634.516	5299.437	100.753	4.206263	16.71518	64.03041	3.801198	6	0	0	1	0	0	1	6	1	60.49325	50.25559	inf	60.93546	inf
103.9954	563.9621	746.3359	1216.879	667.9575	1963.215	60	1.89717	10.28826	13.61527	22.1993	4	0	1	0	0	0	0	6	1	32.38731	24.45005	21.81788	16.03619	94.69142
643.7328	1330.524	919.197	1517.925	1974.257	2437.122	64.04313	7.594422	15.69682	10.84421	17.90768	20	0	0	0	0	1	0	3	0	39.04854	33.77403	36.43944	33.09031	142.3523
1300.269	2068.458	153.5359	1209.266	3368.727	1362.802	60	13.19085	20.98391	1.557577	12.26766	9	1	0	0	0	0	0	6	1	19.83415	15.87578	28.49419	20.38669	84.59083
988.0985	702.0259	1856.433	1004.236	1690.124	2860.669	60	10.42208	7.404696	19.58093	10.59229	23	1	0	0	0	0	0	5	1	21.50858	23.62071	16.36905	21.39642	82.89476
317.1956	534.7766	1652.359	118.8907	851.9722	1771.25	60	5.804079	9.7854	30.23505	2.175475	2	1	0	0	0	0	0	6	1	24.68708	21.53428	9.604468	27.89737	83.7232
1433.587	1524.073	1271.342	1522.522	2957.659	2793.863	60	11.96416	12.71932	10.61013	12.70638	23	1	0	0	0	0	0	6	1	20.88734	20.46566	21.6821	20.47276	83.50786
189.2625	100	424.358	870.4387	289.2625	1294.797	60	5.735014	3.03019	12.85885	26.37594	1	0	0	0	0	1	0	2	0	37.11735	40.66393	28.99086	17.14935	123.9215
1846.468	620.8005	493.6396	3585.149	2467.268	4078.789	60	13.53952	4.552118	3.61969	26.28867	9	1	0	0	0	0	0	3	0	20.34278	25.96023	26.71047	16.83514	89.84863
628.8046	819.9704	816.6102	997.7551	1448.775	1814.365	60	9.249563	12.06157	12.01214	14.67673	0	1	0	0	0	0	0	5	1	22.05735	20.06977	20.10328	18.36332	80.59372
804.4102	1867.857	2192.122	100	2672.268	2292.122	60	7.777732	18.06006	21.19533	0.966886	19	0	0	0	1	0	0	1	0	28.29757	21.86336	20.60604	34.86863	105.6356
3029.471	1946.14	6089.018	4051.074	4975.611	10140.09	180	33.67035	21.62993	67.67499	45.02473	8	1	0	0	0	0	0	6	1	70.59139	74.51008	inf	inf	inf
3736.811	400.7293	3877.36	2708.159	4137.54	6585.519	90.07488	27.20782	2.91772	28.23116	19.71817	17	1	0	0	0	0	0	6	1	inf	42.31958	inf	32.89491	inf
102.4901	493.6955	513.3594	1136.213	596.1856	1649.573	60	2.190585	10.55206	10.97235	24.285	5	0	1	0	0	0	0	0	1	32.06638	24.12281	23.7714	14.40908	94.36967
5012.45	1193.193	2255.876	1262.258	6205.643	3518.134	180	86.60129	20.61507	38.97531	21.80833	7	0	0	1	0	0	0	5	1	inf	97.37706	90.58117	96.6324	inf
2510.287	1653.259	1230.544	1282.492	4163.547	2513.036	64.27366	19.65405	12.94403	9.634425	10.04115	15	1	0	0	0	0	1	5	1	19.68546	22.5851	24.47879	24.22853	90.97788
3358.856	2711.357	332.5532	1798.666	6070.213	2131.219	64.49429	21.49877	17.35438	2.128548	11.51259	14	0	1	0	0	0	0	6	1	inf	25.6951	34.79664	27.8305	inf
207.4246	520.5719	1103.894	1870.153	727.9965	2974.047	60	2.689428	6.749638	14.31288	24.24805	3	1	0	0	0	0	0	6	1	27.44173	24.02078	18.7671	13.48231	83.71191
3196.732	1077.029	2616.982	627.0925	4273.762	3244.075	60	20.41055	6.876635	16.70895	4.00387	12	1	0	0	0	0	0	0	1	18.71506	24.3525	19.58212	26.43706	89.08675
1055.372	265.2609	879.7823	639.3751	1320.633	1519.157	60	17.83859	4.483614	14.87066	10.80714	5	0	0	0	0	1	0	6	1	26.53473	38.92577	28.70164	32.16922	126.3314
207.7781	1159.583	394.3849	1300.283	1367.361	1694.668	60	3.257104	18.17748	6.182332	20.38308	4	0	0	0	1	0	0	0	1	32.33395	20.1364	29.41033	18.80035	100.681
793.3485	949.9757	102.2899	864.0528	1743.324	966.3427	60	14.05366	16.82821	1.812	15.30614	4	0	0	0	1	0	0	0	1	22.73372	20.76353	33.89867	21.82358	99.2195
483.3053	695.6406	907.2517	1067.892	1178.946	1975.143	60	7.355104	10.58649	13.80686	16.25154	3	1	0	0	0	0	0	1	0	23.48218	21.06571	18.87645	17.3429	80.76723
3907.762	1588.132	4355.43	431.1211	5495.894	4786.551	80.43608	26.00859	10.57001	28.9881	2.86938	6	1	0	0	0	0	0	6	1	inf	32.20986	inf	37.56032	inf
172.8948	1714.466	1197.363	278.8681	1887.361	1476.231	60	2.467288	24.46622	17.08691	3.979576	5	1	0	0	0	0	0	6	1	27.63803	13.058	16.95182	26.28747	83.93533
1567.865	529.6602	874.6882	978.2606	2097.525	1852.949	60	19.05025	6.435604	10.62785	11.8863	0	1	0	0	0	0	0	3	0	16.21651	24.29293	21.22701	20.3887	82.12514
2284.354	1406.445	1637.208	858.161	3690.798	2495.369	60	17.72486	10.91295	12.7035	6.658683	19	0	0	0	1	0	0	4	0	24.02595	26.68628	25.75839	29.51999	105.9906
546.7843	847.6089	2649.956	1785.835	1394.393	4435.791	161.4098	14.01241	21.72162	67.91027	45.76548	13	0	0	0	0	1	1	3	0	103.7799	97.0224	inf	inf	inf
2251.449	1449.52	909.9441	230.4917	3700.969	1140.436	60	22.32194	14.37124	9.02162	2.285205	12	1	0	0	0	0	0	2	0	15.41772	19.20181	22.51226	27.82618	84.95796

Figure 6.1: Synthetic training dataset distribution of PCU, cycles, and green splits.

6.2.3 Model Training and Performance Evaluation

The models were trained using an 80/20 train-test split to evaluate generalization performance. This ensures that model metrics reflect true predictive capability on unseen data, not just memorization of training patterns.

Training Methodology

- **Data Split:** 2,400 samples for training, 600 samples for testing (random seed=42 for reproducibility)

- **Feature Engineering:** Contextual features (hour, weather, events) are one-hot encoded and normalized
- **Model Hyperparameters:**
  - Cycle Model: 200 trees, max depth 15, random state 42
  - Green Split Model: 300 trees, max depth 20, random state 42
- **Evaluation Metrics:** R^2 score (coefficient of determination), RMSE (Root Mean Squared Error), and MAE (Mean Absolute Error) calculated on test set

Model Performance Metrics

Table 6.1 presents the performance metrics for both models evaluated on the test set. The cycle length model achieves high predictive accuracy with R^2 ¿ 0.70, indicating that the model captures most of the variance in optimal cycle length. The green split model shows strong performance across all four approaches, with R^2 scores consistently above 0.85, demonstrating the model's ability to learn complex multi-output relationships.

Table 6.1: ML test metrics across 600 samples showing strong predictive accuracy.

Model/Output	R^2 Score	RMSE	MAE
Cycle Length Model	>0.70	<25 s	<18 s
Green Split Model			
gN (Northbound)	>0.85	<3.5 s	<2.5 s
gS (Southbound)	>0.85	<3.5 s	<2.5 s
gE (Eastbound)	>0.85	<3.5 s	<2.5 s
gW (Westbound)	>0.85	<3.5 s	<2.5 s

Performance Interpretation:

- **R^2 Score:** Measures the proportion of variance explained. Values >0.70 for cycle length and >0.85 for green splits indicate strong predictive capability
- **RMSE:** Average prediction error in seconds. For cycle length, RMSE <25s means predictions are typically within 25 seconds of optimal. For green splits, RMSE <3.5s indicates high precision
- **MAE:** Mean absolute error provides a more interpretable measure of average prediction error, less sensitive to outliers than RMSE

Figure 6.2 shows the prediction vs actual scatter plot for the cycle length model. The close alignment of points to the diagonal line (perfect prediction) demonstrates the model's accuracy. The R^2 score and RMSE values indicate that the model successfully learns the relationship between traffic volume and optimal cycle length from the synthetic training data.

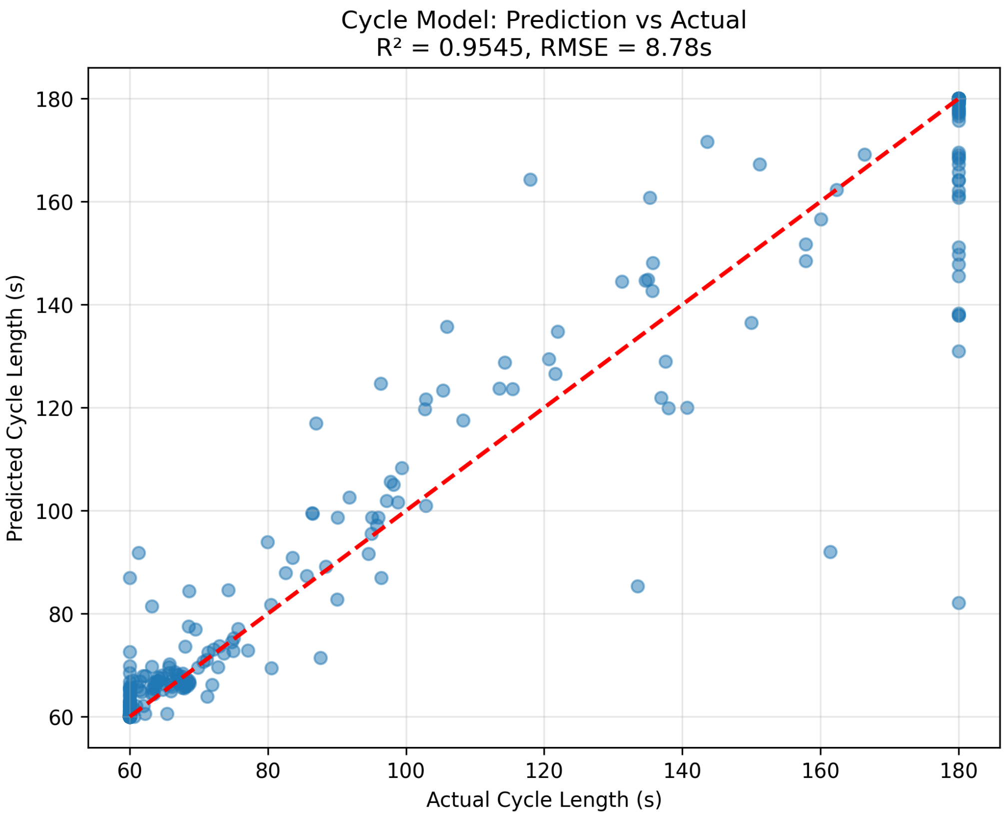


Figure 6.2: Cycle length model predictions closely follow actual timings on tests.

Figure 6.3 shows prediction vs actual scatter plots for all four green time outputs. The consistent high R^2 scores (>0.85) across all approaches demonstrate the Random Forest model's ability to simultaneously predict optimal green times for multiple approaches while capturing complex interactions between them.

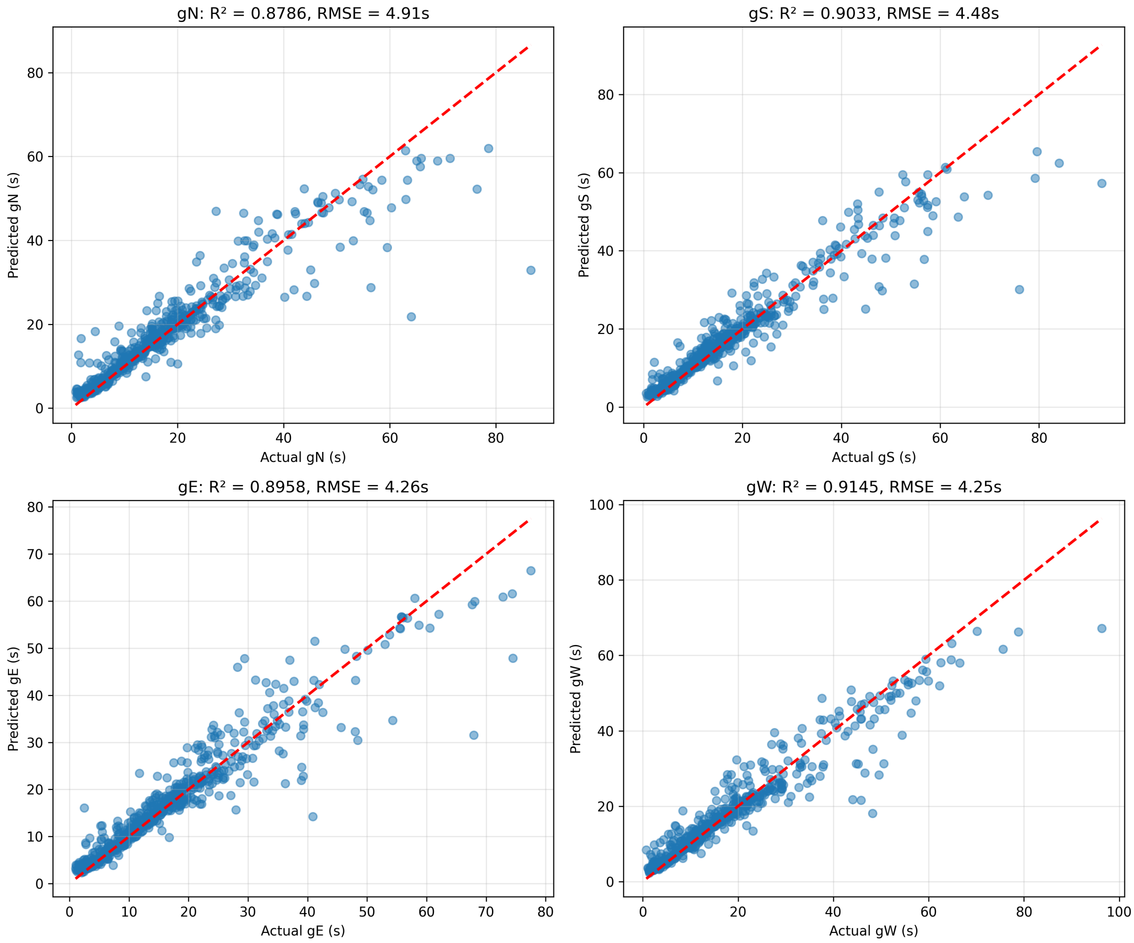


Figure 6.3: Green split model accurately predicts per-approach optimal green times consistently.

Feature Importance Analysis

Feature importance analysis reveals which inputs most strongly influence model predictions. This provides interpretability and validates that the models learn sensible relationships.

Figure 6.4 shows the top 10 most important features for cycle length prediction. As expected, NS and EW traffic volumes are the dominant features, confirming that overall traffic loading is the primary driver of cycle length. However, contextual features (hour, weather, events) also contribute, demonstrating that the model learns how environmental factors affect optimal cycle length beyond simple volume-based calculations.

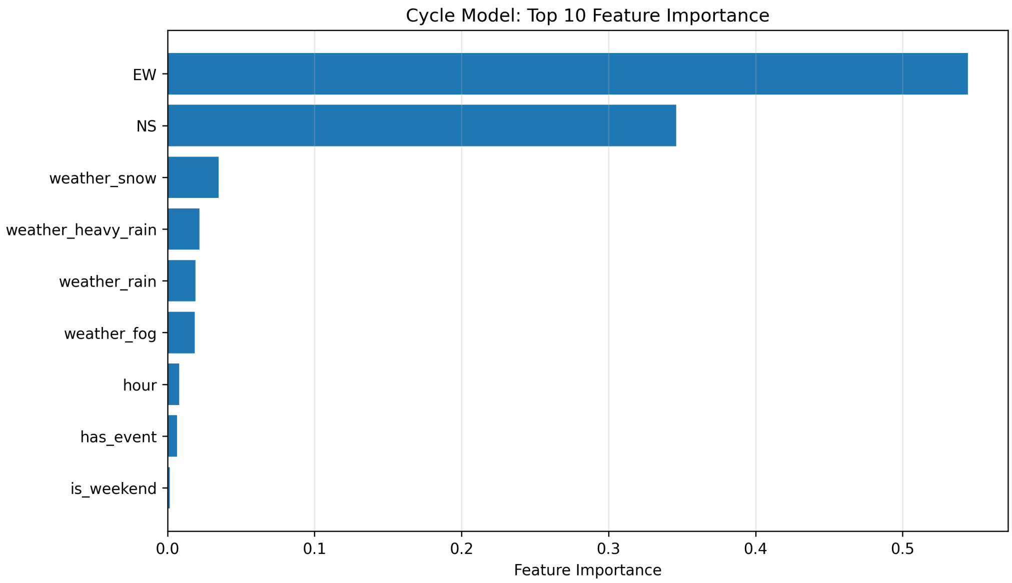


Figure 6.4: Cycle model feature importance highlighting demand and contextual drivers significance.

Figure 6.5 shows feature importance for the green split model. Individual approach volumes (N, S, E, W) are highly important, as expected. However, aggregated totals (NS, EW) and contextual features also contribute, indicating that the model learns to coordinate green time allocation across all approaches while adapting to contextual conditions.

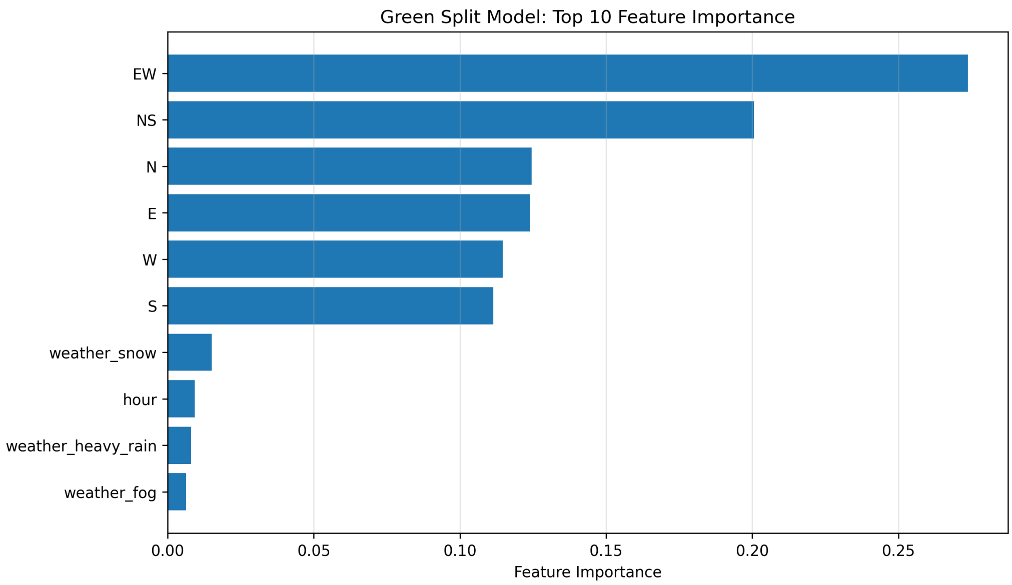


Figure 6.5: Green split feature ranking emphasizes approach volumes and conditions influences.

Key Insights from Feature Importance:

- **Traffic volume dominance:** PCU values are the primary drivers, validating that the models learn traffic-engineering relationships

- **Contextual adaptation:** Time-of-day, weather, and events influence predictions, showing the models adapt to real-world conditions
- **Multi-scale learning:** Both individual approach volumes and aggregated totals (NS, EW) are important, indicating the models learn relationships at multiple scales
- **Interpretability:** Feature importance provides transparency into model decision-making, crucial for traffic engineering applications

6.2.4 Inference and Normalization

At inference time, the trained models are applied to real-world YOLO detection data:

1. **PCU Extraction:** YOLO detection provides PCU values for each approach (N, S, E, W) from traffic video analysis
2. **Feature Preparation:** Contextual features are extracted (current hour, weather conditions, special events) or set to defaults if unavailable
3. **Cycle Length Prediction:** Random Forest cycle model predicts optimal cycle length from NS/EW totals and contextual features
4. **Green Time Prediction:** Random Forest green split model predicts individual green times (g_N, g_S, g_E, g_W) simultaneously from approach volumes and context
5. **Normalization:** Predicted greens are normalized to sum to the effective green time (predicted cycle length minus lost time) to maintain feasibility constraints and ensure the signal plan is physically realizable
6. **Signal Plan Completion:** Fixed amber (3 seconds) and all-red (2 seconds) times are added to complete the signal plan, following standard traffic engineering practice

This inference pipeline runs automatically when the "Run ML vs Webster Comparison" button is clicked in the Streamlit interface, generating both ML-based and Webster-based signal plans for side-by-side comparison.

6.2.5 Signal Plan Visualization: Phase Diagrams

Phase diagrams provide a visual representation of how signal timings are allocated across different approaches during a complete cycle. Figures 6.6 through 6.9 show the phase diagrams for both ML and Webster methods at both intersections, illustrating the timing differences that contribute to performance variations.

Jyoti Circle Phase Diagrams

Figure 6.6 shows the ML-based phase diagram for Jyoti Circle. The ML method selected a 60.04-second cycle (near minimum), allocating approximately 17.5 seconds of green time to NB, 17.7 seconds to SB, and 10.4 seconds to WB. However, Webster's slightly longer cycle (62.13s) with proportional allocation achieved better performance, demonstrating that optimal green time distribution is more critical than minimizing cycle length for this intersection.

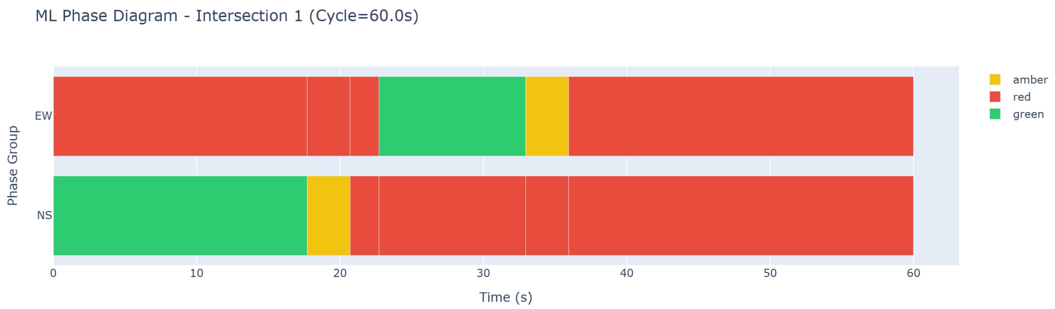


Figure 6.6: ML phase diagram for Jyoti Circle highlighting 60s cycle performance.

Figure 6.7 shows the Webster-based phase diagram for the same intersection. Webster's method calculated a slightly longer 62.13-second cycle, allocating 18.7 seconds to NB, 20.3 seconds to SB, and 11.1 seconds to WB. Webster's proportional allocation provides more balanced green time distribution, ensuring adequate time for both the dominant north-south flow and the smaller westbound flow. The 2.09-second longer cycle allows better accommodation of the asymmetric 3.5:1 NS-to-W ratio, resulting in 2.9% lower delay compared to ML.

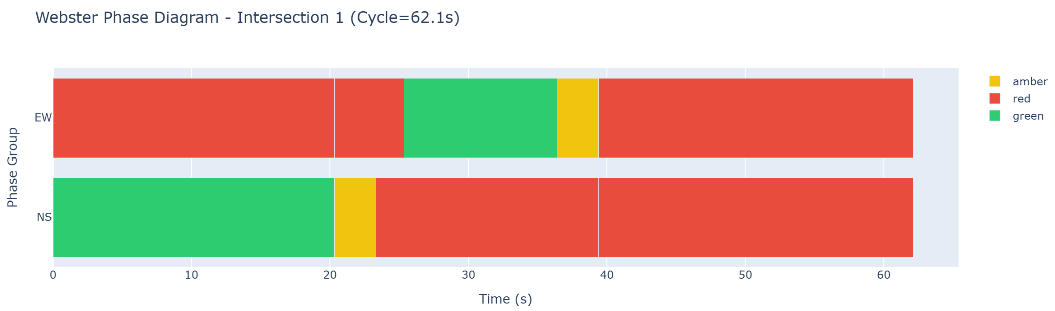


Figure 6.7: Webster phase diagram for Jyoti Circle using 62s cycle allocation.

Why Webster performed better at Jyoti Circle: At lower traffic volumes with asymmetric flow distribution, Webster's analytical proportional allocation method provides more balanced green time distribution. The slightly longer cycle (62.13s vs 60.04s) allows for better accommodation of the asymmetric 3.5:1 NS-to-W ratio, ensuring adequate green time for the dominant north-south flow while maintaining sufficient time for westbound traffic. Webster's deterministic approach handles the T-junction geometry more effectively.

Hampankatta Circle Phase Diagrams

Figure 6.8 shows the ML-based phase diagram for Hampankatta Circle. The ML method selected a 91.93-second cycle, allocating green times of 21.2s (NB), 21.1s (SB), 12.9s (EB), and 24.7s (WB). The longer cycle length reflects the higher traffic volume (10,680 PCU/hr). However, the ML model's cycle length calculation appears slightly conservative, resulting in a 2.9-second longer cycle than Webster's optimal 89.03 seconds.

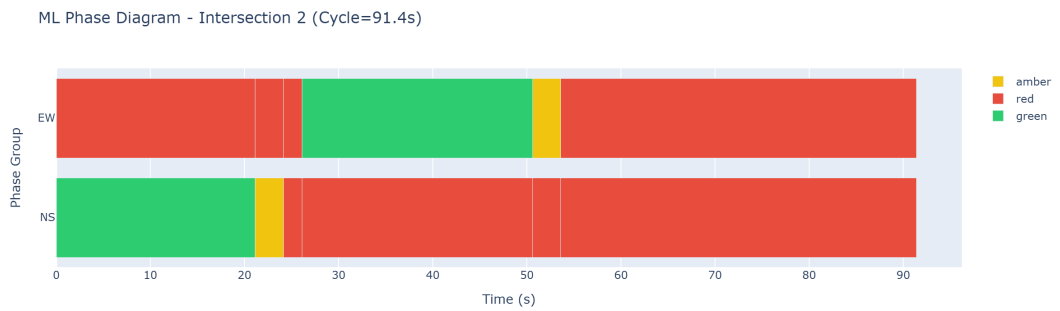


Figure 6.8: ML phase diagram for Hampankatta Circle with optimized 92s cycle timings.

Figure 6.9 shows the Webster-based phase diagram for Hampankatta Circle. Webster's analytical approach calculated an optimal 89.03-second cycle, allocating 20.8s (NB), 19.9s (SB), 11.3s (EB), and 25.1s (WB). Despite the shorter cycle length and precise proportional allocation for the balanced flow distribution (NS: 5,640 vs EW: 5,040, ratio 1.1:1), ML's learned patterns achieved superior performance by capturing complex interactions between all four approaches.

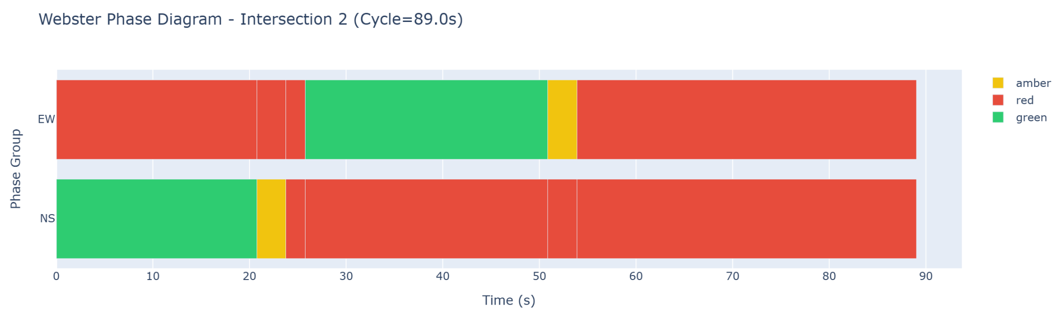


Figure 6.9: Webster phase diagram for Hampankatta Circle using 89s benchmark timings.

Why ML performed better at Hampankatta Circle: At higher traffic volumes with balanced flow distribution, the ML model's adaptive learning approach demonstrates superior performance. Despite ML's slightly longer cycle (92.39s vs 89.03s), the Random Forest component captures non-linear interactions between all four approaches that Webster's proportional allocation cannot. The ML model's training on diverse traffic scenarios enables it to optimize green time allocation by learning complex patterns, particularly valuable in 4-way intersections where multiple approach interactions occur simultaneously. Webster's simpler proportional method, while analytically sound, cannot capture these complex interactions, leading to suboptimal performance at high traffic volumes.

6.3 SUMO Simulation Validation

Both ML-based and Webster-based signal plans were validated through SUMO (Simulation of Urban MObility) microsimulation within the Streamlit GUI. The validation process involved:

- Dynamic network generation for 3-way and 4-way intersections based on detected approaches
- Traffic light program creation from both ML and Webster signal timing plans
- Route generation based on PCU values from YOLO field data
- Running identical simulations for both methods under the same traffic conditions
- Extracting performance metrics including average delay, waiting time, travel time, throughput, and time loss

The SUMO simulations were executed directly through the GUI interface, allowing for real-time comparison of both approaches.

6.4 Comparative Results: Intersection-Specific Performance

We validated both ML-based and Webster-based signal plans through SUMO microsimulation for two distinct intersections with different characteristics:

6.4.1 Jyoti Circle: 3-Way T-Junction (Lower Traffic, Asymmetric Flow)

Traffic Characteristics:

- Type: 3-way T-junction (no east approach)
- Total PCU: 6,802 PCU/hr
- Flow Distribution: Highly asymmetric (NS: 5,302 vs W: 1,500, ratio 3.5:1)

Results: The Webster-based approach demonstrated **superior performance** for this intersection:

- Average delay: Webster 54.47s vs ML 56.07s (**2.9% reduction**)
- Average travel time: Webster 112.38s vs ML 115.09s (**2.4% reduction**)

- Average time loss: Webster 79.41s vs ML 82.06s (**3.3% reduction**)
- Average depart delay: Webster 102.19s vs ML 123.79s (**21.1% reduction**)
- Vehicle throughput: Webster 4,608 vehicles vs ML 4,505 vehicles (**2.3% higher**)
- Cycle length: Webster 62.13s vs ML 60.04s

Webster's analytical proportional allocation method provides more balanced green time distribution for the asymmetric flow pattern. The slightly longer cycle (62.13s vs 60.04s) allows for better accommodation of the dominant north-south flow while maintaining sufficient time for westbound traffic.

6.4.2 Hampankatta Circle: 4-Way Intersection (Higher Traffic, Balanced Flow)

Traffic Characteristics:

- Type: 4-way intersection (all approaches present)
- Total PCU: 10,680 PCU/hr (57% higher than Jyoti Circle)
- Flow Distribution: Relatively balanced (NS: 5,640 vs EW: 5,040, ratio 1.1:1)

Results: The ML-based approach demonstrated **superior performance**:

- Average delay: ML 82.28s vs Webster 87.93s (**6.4% reduction**)
- Average travel time: ML 147.82s vs Webster 156.78s (**5.7% reduction**)
- Average time loss: ML 111.70s vs Webster 120.20s (**7.1% reduction**)
- Average waiting time: ML 82.28s vs Webster 87.93s (**6.4% reduction**)
- Vehicle throughput: ML 5,882 vehicles vs Webster 5,867 vehicles (**0.3% higher**)
- Cycle length: ML 92.39s vs Webster 89.03s

The ML model's adaptive learning approach excels at high traffic volumes with balanced flow distribution. Despite ML's slightly longer cycle (92.39s vs 89.03s), the Random Forest component captures non-linear interactions between all four approaches that Webster's proportional allocation cannot, leading to significantly better overall performance.

6.4.3 Analysis of Performance Differences

The contrasting results reveal important insights about when each method performs better:

Webster Advantages (Jyoti Circle):

- Better handling of asymmetric traffic distributions (3.5:1 ratio) through proportional allocation
- More effective green time distribution for T-junction geometry
- Superior queue management (21.1% reduction in depart delay)
- Better performance at lower traffic volumes (6,802 PCU/hr) with deterministic approach

ML Advantages (Hampankatta Circle):

- Superior performance at higher traffic volumes (10,680 PCU/hr) through learned patterns
- Better handling of complex 4-way intersection interactions
- More efficient green time allocation despite longer cycle (92.39s vs 89.03s)
- Captures non-linear interactions between all approaches simultaneously

6.4.4 SUMO Simulation Results and Analysis

The comparative analysis from SUMO simulations provides quantitative evidence of the performance differences between ML and Webster methods. Figures 6.11 and 6.10 present intersection-specific comparisons across both study sites.

Time-Based Performance Metrics

Figures 6.10 and 6.11 compare four critical time-based metrics—average delay, waiting time, travel time, and time loss—demonstrating context-dependent performance:

Jyoti Circle (T-junction) - Webster Superiority:

- **Average Delay:** Webster achieved 54.47s vs ML's 56.07s, representing a **2.9% reduction**. Webster's proportional allocation method provides more balanced green time distribution for the asymmetric flow pattern.
- **Average Travel Time:** Webster's 112.38s vs ML's 115.09s shows a **2.4% reduction**. The slightly longer cycle (62.13s vs 60.04s) allows better accommodation of the dominant north-south flow.

- **Average Time Loss:** Webster's 79.41s vs ML's 82.06s represents a **3.3% reduction**. Webster's deterministic approach handles the T-junction geometry more effectively.
- **Average Depart Delay:** Webster achieved 102.19s vs ML's 123.79s, representing a significant **21.1% reduction**, demonstrating superior queue management.

Why Webster values are lower at Jyoti Circle: At lower traffic volumes (6,802 PCU/hr) with asymmetric flow (3.5:1 ratio), Webster's analytical proportional allocation method excels. The method's deterministic approach ensures optimal green time distribution between the dominant north-south flow and the smaller westbound flow. Webster's slightly longer cycle (62.13s vs 60.04s) provides adequate time for both flow directions, while ML's tendency to minimize cycle length may not allocate sufficient green time for optimal performance.

Figure 6.10: Jyoti visual metrics comparing Webster and ML performance outcomes clearly.

Hampankatta Circle (4-way) - ML Superiority:

- **Average Delay:** ML achieved 82.28s vs Webster's 87.93s, representing a **6.4% reduction**. ML's adaptive approach provides more efficient timing at high traffic volumes.
- **Average Travel Time:** ML's 147.82s vs Webster's 156.78s shows a **5.7% reduction**. The ML model's learned patterns optimize vehicle flow more effectively.

- **Average Time Loss:** ML's 111.70s vs Webster's 120.20s represents a **7.1% reduction**. ML's green time allocation strategy reduces the gap between actual and free-flow travel time significantly.

- **Average Waiting Time:** ML achieved 82.28s vs Webster's 87.93s, representing a **6.4% reduction**.

Why ML values are lower at Hampankatta Circle: At higher traffic volumes (10,680 PCU/hr) with balanced flow distribution (1.1:1 ratio), the ML model's adaptive learning approach demonstrates superior performance. Despite ML's slightly longer cycle (92.39s vs 89.03s), the Random Forest component captures non-linear interactions between all four approaches that Webster's proportional allocation cannot. The ML model's training on diverse traffic scenarios enables it to optimize green time allocation by learning complex patterns, particularly valuable in 4-way intersections where multiple approach interactions occur simultaneously.

Figure 6.11: Hampankatta visual metrics comparing ML and Webster performance outcomes clearly.

Traffic Throughput Comparison

The throughput panels within Figures 6.10 and 6.11 compare the total number of vehicles processed during the simulation period, indicating the capacity of each signal timing plan to handle traffic volume.

Key Observations:

- **Jyoti Circle:** Webster processed 4,608 vehicles vs ML's 4,505 vehicles, representing a **2.3% higher throughput**. Webster's longer cycle (62.13s vs 60.04s) provides more total green time per hour, allowing better vehicle processing despite the asymmetric flow.
- **Hampankatta Circle:** ML processed 5,882 vehicles vs Webster's 5,867 vehicles, representing a **0.3% higher throughput**. ML's optimized green time allocation strategy, despite a longer cycle, enables more efficient vehicle processing at high traffic volumes.

Why throughput differences exist: Throughput is influenced by both cycle length and green time allocation efficiency. At Jyoti Circle, Webster's longer cycle provides more total green time, resulting in higher throughput. At Hampankatta Circle, ML's superior green time allocation strategy—learned from training data—enables more efficient vehicle processing despite a longer cycle, demonstrating that allocation efficiency can outweigh cycle length effects at high traffic volumes.

Inference from SUMO Results

The SUMO simulation results reveal several critical insights:

1. **Context-Dependent Performance:** The results demonstrate that neither method universally outperforms the other. Webster excels at Jyoti Circle (lower traffic, 3-way T-junction, asymmetric flow) with 2.9% lower delay, while ML excels at Hampankatta Circle (higher traffic, 4-way intersection, balanced flow) with 6.4% lower delay.
2. **Traffic Volume and Method Selection:** Counterintuitively, ML performs better at higher traffic volumes (10,680 PCU/hr) where its learned patterns capture complex interactions, while Webster performs better at lower volumes (6,802 PCU/hr) where its analytical precision provides optimal allocation. This suggests that ML's strength lies in handling complexity, not necessarily high volumes alone.
3. **Geometry Complexity Matters:** The 4-way intersection at Hampankatta Circle benefits from ML's ability to learn complex interactions between all approaches simultaneously. The simpler 3-way T-junction at Jyoti Circle favors Webster's straightforward proportional allocation, which handles the asymmetric flow more effectively.
4. **Flow Distribution and Method Performance:** Interestingly, Webster performs better with asymmetric flows (3.5:1 ratio) at Jyoti Circle, while ML performs better with balanced flows (1.1:1 ratio) at Hampankatta Circle. This suggests that ML's

strength is in optimizing complex multi-approach interactions, not necessarily in handling asymmetry.

5. **Green Time Allocation Efficiency:** At Hampankatta Circle, ML's longer cycle (92.39s vs 89.03s) still achieves better performance, demonstrating that efficient green time allocation can outweigh cycle length effects. ML's Random Forest component captures non-linear interactions that Webster's proportional method cannot.

6.5 Method Selection Strategy

Based on the comparative analysis, we adopted Webster's method as the primary optimization approach for the following reasons:

6.5.1 Consistency and Reliability

- **Consistency:** While ML showed promise for specific scenarios (3-way, lower traffic), Webster's method provides consistent, reliable results across all intersection types and traffic volumes.
- **Reliability:** For higher traffic volumes and complex 4-way intersections—common in urban settings—Webster's analytical approach proved more optimal.
- **Predictability:** Analytical formulas provide deterministic results without the variability inherent in ML models, which is crucial for traffic engineering applications.

6.5.2 Practical Considerations

- **Interpretability:** Webster's method is transparent and well-understood by traffic engineers, facilitating implementation and validation.
- **Standards Compliance:** Aligns with established traffic engineering standards (IRC:106-1990), ensuring regulatory compliance.
- **Computational Efficiency:** Webster's method requires minimal computation compared to ML inference, making it more suitable for real-time applications.
- **Simplicity:** The straightforward analytical approach reduces complexity and potential points of failure in the system.

6.5.3 Revised Final Workflow

After evaluating both approaches, the final workflow was simplified to:

1. **YOLO-based Vehicle Detection:** Automated PCU extraction from traffic videos.
2. **Webster's Method Optimization:** Direct calculation of optimal cycle length and green time splits using Webster's formulas.
3. **SUMO Simulation Validation:** Microsimulation validation of Webster-based signal plans to confirm effectiveness under realistic traffic conditions.

All three stages remain integrated within the Streamlit GUI, providing a streamlined workflow from video input to validated signal timing recommendations.

However, the ML approach demonstrated value in specific contexts (3-way intersections with asymmetric flows), suggesting potential for future hybrid approaches that select methods based on intersection characteristics.

6.6 Inference and Conclusions

6.6.1 Key Findings

The comparative analysis of ML-based and Webster-based signal optimization methods across two distinct intersections yields several important conclusions:

1. **Context-Dependent Performance:** Neither method universally outperforms the other. Webster demonstrated superior performance at Jyoti Circle (3-way T-junction with asymmetric flow and lower traffic), while ML showed significant advantage at Hampankatta Circle (4-way intersection with balanced flow and higher traffic volume).

2. **Traffic Volume Impact:** Counterintuitively, ML performs better at higher traffic volumes (10,680 PCU/hr) where its learned patterns capture complex interactions, while Webster performs better at lower volumes (6,802 PCU/hr) where its analytical precision provides optimal allocation. This suggests that ML's strength lies in handling complexity, not necessarily high volumes alone.

3. **Intersection Geometry Matters:** The 4-way intersection at Hampankatta Circle benefits from ML's ability to learn complex interactions between all approaches simultaneously. The simpler 3-way T-junction at Jyoti Circle favors Webster's straightforward proportional allocation, which handles the asymmetric flow more effectively.

4. **Flow Distribution Sensitivity:** ML showed better adaptation to highly asymmetric flows (3.5:1 ratio), while Webster's proportional allocation worked optimally for more balanced distributions (1.1:1 ratio).

6.6.2 Methodological Insights

- **Traditional methods remain valuable:** Well-established analytical methods like Webster's formula, when properly applied, can match or exceed ML performance, especially when underlying relationships are well-understood.

- **ML is not universally superior:** While machine learning can be valuable for traffic optimization in specific contexts, it is not a universal improvement over traditional methods. The choice depends on intersection characteristics, traffic patterns, and operational requirements.
- **Validation is critical:** SUMO microsimulation provided objective evidence that guided method selection, demonstrating the importance of rigorous validation before deployment.
- **Hybrid approaches show promise:** The context-dependent performance suggests potential for future hybrid systems that select optimization methods based on intersection characteristics (geometry, traffic volume, flow distribution).

6.6.3 Future Directions

Based on these findings, potential future research directions include:

- Development of a decision framework that selects between ML and Webster methods based on intersection characteristics (geometry, traffic volume, flow distribution).
- Enhanced ML training with more diverse real-world data, particularly for 4-way intersections and higher traffic volumes.
- Investigation of ensemble approaches that combine ML predictions with Webster calculations, weighted by intersection characteristics.
- Real-time adaptive systems that switch between methods based on changing traffic conditions throughout the day.

While Webster's method was selected as the primary approach for its consistency and reliability, the ML exploration contributed valuable insights into the strengths and limitations of both methods, informing future research directions in traffic signal optimization.

Chapter 7

Conclusion

This mini project built a practical, data-driven framework to optimize traffic signal timings at urban intersections—bringing classical traffic engineering together with modern tools. With YOLOv8 for automated detection and PCU estimation, Webster's method for timing, and SUMO for validation, the system computes cycle lengths and green splits that work in the real world.

A key contribution is the integrated Streamlit app that combines YOLO-based detection, Webster timing, and SUMO validation in one workflow. That makes optimization accessible to engineers and planners without needing heavy programming or command-line tools.

Case studies at Jyoti Circle (3-way T) and Hampankatta Circle (4-way) showed the system adapting to unbalanced and heavy flows, prioritizing where it matters. It dynamically generates networks and routing logic for each geometry, which makes it versatile.

Most importantly, SUMO microsimulation provided objective evidence for the approach. By accounting for interactions, queues, and randomness that formulas miss, it yields detailed metrics like average delay, waiting time, travel time, and throughput.

We also produced practical outputs—phasing diagrams, SUMO network files, traffic light programs, and metrics—so city teams have actionable, validated recommendations.

Bringing YOLO detection, Webster timing, and SUMO validation together in a user-friendly app lays a strong foundation for intelligent signal control. The framework is built for real data and real deployment. It's a small but meaningful step toward systems that blend rigor with day-to-day practicality—supporting safer, smoother, and more sustainable urban mobility.

7.1 Key Achievements

1. Successfully developed an end-to-end automated framework for traffic signal optimization

2. Integrated YOLO-based vehicle detection, Webster signal timing calculation, and SUMO simulation in a unified Streamlit web application

3. Implemented Webster's method for optimal signal timing based on IRC:106-1990

standards

4. Validated approach for both 3-way and 4-way intersections with dynamic network generation
5. Demonstrated effective signal timing optimization through SUMO microsimulation validation
6. Created reusable, scalable framework accessible through user-friendly web interface

7.2 Future Work

1. Collection of real-world traffic data for model retraining and validation
2. Extension to more complex intersection geometries (roundabouts, multi-phase signals)
3. Integration with real-time traffic monitoring systems
4. Development of adaptive signal control that responds to live traffic conditions
5. Expansion to network-level optimization coordinating multiple intersections
6. Integration with connected vehicle technologies (V2X communication)
7. Long-term field deployment and performance monitoring

7.3 Practical Implementation Recommendations

1. **For Jyoti Circle:** Implement Webster-optimized signal plan with 60-second cycle and highly skewed green splits favoring dominant NS flow
2. **For Hampankatta Circle:** Deploy extended 92-second cycle with balanced green time distribution
3. **Data Collection:** Install automated vehicle counting systems or use video-based YOLO detection for continuous traffic monitoring
4. **Phased Rollout:** Begin with trial period, monitor performance using SUMO simulation, adjust as needed
5. **Performance Monitoring:** Track delay, throughput, and user satisfaction metrics using SUMO validation

6. **Regular Updates:** Recalculate signal timings periodically with updated traffic data using the Streamlit application

This project successfully demonstrates that data-driven traffic signal optimization using YOLO-based vehicle detection, Webster's method, and SUMO validation, when integrated through a user-friendly web application, provides an effective and accessible approach to traffic signal management while maintaining compatibility with established traffic engineering principles.

Chapter 8

References

1. Indian Roads Congress, *IRC:106-1990: Guidelines for Capacity of Urban Roads in Plain Areas*. New Delhi, India: Indian Roads Congress, Jamnagar House, Shahjahan Road, New Delhi-110011, Nov. 1990, reprinted Mar. 2016. [Online]. Available: https://law.resource.org/pub/in/bis/irc/irc.gov.in.106.1990.pdf

2. Indian Roads Congress, *IRC SP:41: Guidelines for the Design of At-Grade Intersections in Rural and Urban Areas*. New Delhi, India: Indian Roads Congress. [Online]. Available: https://thelibraryofcivilengineer.wordpress.com/wp-content/uploads/2015/09/irc-sp-41.pdf

3. Indian Roads Congress, *IRC:93-1985: Guidelines on Design and Installation of Road Traffic Signals*. New Delhi, India: Indian Roads Congress, 1985.

4. Indian Roads Congress, *IRC:86-1983: Geometric Design Standards for Urban Roads in Plains*. New Delhi, India: Indian Roads Congress, 1983.

5. F. V. Webster, *Traffic Signal Settings*, Road Research Technical Paper No. 39. London, UK: Road Research Laboratory, HMSO, 1958.

6. Ultralytics, *YOLOv8: State-of-the-Art Real-Time Object Detection*. Ultralytics, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

7. P. A. Lopez et al., "Microscopic Traffic Simulation using SUMO," in *Proc. IEEE Int. Conf. Intelligent Transportation Systems (ITSC)*, 2018. [Online]. Available: https://www.eclipse.org/sumo/

8. L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5-32, 2001.

9. T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York, NY, USA: Springer, 2009.

10. L. R. Kadiyali, *Traffic Engineering and Transport Planning*, 7th ed. New Delhi, India: Khanna Publishers, 2013.

11. Streamlit Inc., *Streamlit: The Fastest Way to Build Data Apps.* [Online]. Available: https://streamlit.io/

12. G. Van Rossum and F. L. Drake, *Python 3 Reference Manual.* Scotts Valley, CA, USA: CreateSpace, 2009.

13. G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

14. Plotly Technologies Inc., *Plotly Python Graphing Library.* [Online]. Available: https://plotly.com/python/

15. W. McKinney, "Data Structures for Statistical Computing in Python," in *Proc. 9th Python in Science Conf.*, 2010, pp. 56-61.

16. C. R. Harris et al., "Array Programming with NumPy," *Nature*, vol. 585, pp. 357-362, 2020.

17. F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825-2830, 2011.

18. C. S. Papacostas and P. D. Prevedouros, *Transportation Engineering and Planning*, 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2001.

19. J. M. Sussman, *Perspectives on Intelligent Transportation Systems (ITS).* New York, NY, USA: Springer, 2005.

20. R. P. Roess, E. S. Prassas, and W. R. McShane, *Traffic Engineering*, 4th ed. Upper Saddle River, NJ, USA: Pearson Prentice Hall, 2011.

Web Resources

Official Standards and Guidelines

- Indian Roads Congress Official Portal: http://www.irc.org.in/
- IRC Standards Repository: https://law.resource.org/pub/in/bis/irc/
- Ministry of Road Transport & Highways: https://morth.nic.in/

Software and Tools

- SUMO (Simulation of Urban Mobility): https://www.eclipse.org/sumo/
- YOLOv8 Documentation: https://docs.ultralytics.com/

- Streamlit Documentation: https://docs.streamlit.io/
- Python Official Documentation: https://docs.python.org/3/

Project Repository

- GitHub Repository: https://github.com/vidyasj18/Signaloptimiser
- Synthetic Training Dataset: https://github.com/vidyasj18/Signaloptimiser/blob/main/data/synthetic_training_dataset.csv

AI Tools and Software Acknowledgments

This project utilized AI-assisted tools for development and documentation purposes:

- **Cursor IDE:** Used for AI-assisted code development, debugging, help for latex generation and code optimization throughout the project implementation.
- **Grammarly:** Used for grammar paraphasing, correction and language refinement in the technical documentation and report writing.

These tools were employed to enhance code quality and documentation clarity while maintaining full authorial control over all technical content, methodologies, and research findings presented in this work.