AI – Driven Traffic Congestion Prediction & FPGA – Based Signal Controller

Submitted by

Nagalli Chethana
(126180029@sastra.ac.in,8790884050)

Baranika R
(126180060@sastra.ac.in, 9345590560)

K. Parvathavardhini Priya Sadhvi
(126180019@sastra.ac.in, 6302539504)

Kondini Vanitha
(126180021@sastra.ac.in, 9000631426)

Abstract

Urban traffic congestion has become a major challenge due to rapid urbanization, high vehicle density, and non-adaptive fixed-time traffic signal systems. Traditional controllers fail to react to sudden congestion variations, resulting in increased waiting time, fuel consumption, and pollution. This project presents a hybrid **AI–ML + FPGA-based intelligent traffic control system** capable of predicting congestion levels and dynamically adjusting traffic signal timings in real-time.

A synthetic dataset was generated using the SUMO traffic simulator for low, medium, high, and peak traffic densities. A **CNN–LSTM hybrid deep learning model** was trained to forecast traffic congestion several timesteps ahead. The model output is transmitted to an **FPGA-based traffic light controller**, designed using a deterministic FSM that adapts the green time based on the predicted congestion. Emergency vehicle override and a fail-safe operational mode are also incorporated to ensure system safety and reliability.

Simulation results demonstrate effective congestion prediction, reduced average waiting time, and improved lane throughput compared to fixed-time signal control. This hybrid system proves to be a promising approach for next-generation smart city traffic management.

Table of Contents

List of Figures

Abbreviations

1. AI – Artificial Intelligence
2. ML – Machine Learning
3. FPGA – Field Programmable Gate Array
4. CNN – Convolutional Neural Network
5. LSTM – Long Short-Term Memory
6. FSM – Finite State Machine
7. SUMO – Simulation of Urban Mobility
8. UART – Universal Asynchronous Receiver-Transmitter
9. ASIC – Application Specific Integrated Circuit

INTRODUCTION

1.1 Traffic Congestion in Urban Areas

Urban traffic congestion has become a major challenge across fast-growing cities due to increased vehicle ownership, expansion of road networks, and unbalanced distribution of traffic loads. Intersections are key bottlenecks where traffic accumulates, resulting in long delays, reduced travel efficiency, and driver frustration. Peak-hour congestion further increases fuel consumption and emissions, contributing to environmental deterioration and economic losses.

With the growing demand for smart transportation systems, researchers and urban planners are focusing on developing intelligent traffic control solutions. One of the most promising directions involves utilizing data-driven mechanisms to understand traffic patterns and optimize signal timings dynamically. Such systems aim to ensure smoother vehicle flow, shorter waiting times, and improved overall mobility.

1.2 Limitations of Traditional Traffic Signal Systems

Conventional traffic light systems operate using fixed or preset timing cycles, regardless of real-time conditions. This rigid timing strategy fails to account for fluctuating vehicle density, sudden congestion build-up, or special situations such as accidents or lane blockages. As a result, some lanes may remain unnecessarily red even when there is no incoming traffic, while heavily congested lanes do not receive extended green time.

Moreover, these systems lack predictive intelligence and emergency-handling capability. They cannot foresee traffic buildup in advance, making them reactive rather than proactive. Such limitations highlight the need for an advanced system capable of adapting signal timings in real time based on congestion levels predicted using machine learning models.

1.3 Need for AI-Driven Control

Machine learning-driven traffic control brings the advantage of predictive capability. By analyzing historical patterns, temporal trends, and real-time input, AI models can forecast congestion before it occurs. This allows traffic signals to be adjusted ahead of time, preventing queue buildup and reducing overall delay. Deep learning architectures like CNN–LSTM are particularly effective for capturing spatio-temporal traffic behavior.

AI-driven control enables intelligent decision-making such as optimizing green durations, balancing flow across intersections, and adjusting cycles for different traffic scenarios. This level of adaptiveness is not possible with traditional systems, making AI essential for modern smart cities striving for seamless transportation efficiency.

1.4 FPGA for Real-Time Traffic Control

While AI models provide intelligence, they are often too computationally heavy to directly control hardware signals with deterministic timing. FPGAs, on the other hand, excel at real-time, parallel, and reliable operations. They can implement state machines for traffic lights with microsecond-level precision, ensuring safety and consistent behavior even under varying conditions.

By integrating AI predictions into FPGA hardware, we combine decision-making intelligence with reliable real-time execution. This hybrid approach ensures accurate congestion forecasting alongside deterministic control, making FPGA a suitable platform for traffic management in safety-critical environments.

1.5 Motivation

The motivation behind this project arises from the growing need for adaptive, intelligent traffic control systems that can reduce congestion and improve road efficiency. With advancements in AI and embedded hardware, it is now feasible to design a system where predictive insights drive real-world traffic signal behavior.

This project aims to bridge the gap between data-driven intelligence and real-time hardware control. The integration of AI and FPGA introduces a scalable, robust, and responsive traffic management solution suitable for smart cities and future intelligent transportation infrastructures.

PROBLEM STATEMENT AND OBJECTIVES

2.1 Problem Statement

Urban intersections experience highly unpredictable traffic patterns due to rapid urbanization, varying vehicle inflow rates, road bottlenecks, and unplanned traffic behavior. Current fixed-time traffic signal controllers operate using predetermined timing cycles that do not respond to real-time vehicle density or congestion levels. As a result:

- Vehicles experience long waiting times at red signals even when the intersecting lane has low traffic.
- There is no mechanism to forecast impending congestion based on temporal traffic patterns.
- Traditional controllers cannot prioritize emergency vehicles such as ambulances, fire engines, or police vehicles.
- Lack of adaptability leads to fuel wastage, increased emissions, and reduced traffic throughput.
- Hardware-based controllers without predictive intelligence cannot dynamically modify timing without manual intervention.

Additionally, although AI models offer strong predictive capabilities, they cannot directly control real-time signals due to computational latency and nondeterministic software response times. Conversely, FPGA-based controllers are fast and reliable but lack intelligence for predictive decision-making.

Thus, there is a critical need for a **hybrid AI + FPGA based traffic management system** that combines:

- ML-driven congestion prediction
- Deterministic real-time control
- Emergency handling capability
- Fail-safe and secure operation

to improve traffic efficiency and reduce congestion in smart cities.

2.2 Objectives

The primary objectives of this project are as follows:

1. Develop a Traffic Dataset Using SUMO

- Create realistic road networks and traffic flows.
- Generate congestion scenarios such as low, medium, high, and peak density.

2. Design a CNN–LSTM Hybrid ML Model

- Extract spatial features using CNN layers.
- Learn temporal patterns with LSTM layers.
- Predict congestion levels 1–5 timesteps ahead.

3. Implement an FPGA-Based Traffic Signal Controller

- Design a multi-state FSM to control traffic lights.
- Integrate dynamic green-time adjustment capability.
- Ensure deterministic and low-latency hardware operation.

4. Integrate AI Prediction with FPGA Control

- Transmit ML output to FPGA via UART/I2C interface.
- Map predicted congestion levels to corresponding green times.

5. Implement Emergency Vehicle Override

- Detect ambulance request input.
- Enforce immediate green signal for priority lanes.

6. Design a Fail-Safe Mode

- Provide default fixed timing when ML communication fails.
- Ensure continuous traffic flow even during faults.

7. Validate System Through Simulations

- Verify ML accuracy with performance metrics.
- Simulate FPGA timing behavior using testbenches and waveforms.
- Compare fixed-time vs adaptive control performance.

EXPERIMENTAL WORK / METHODOLOGY

3.1 Dataset Generation using SUMO

SUMO was used to simulate real-world intersection conditions by creating traffic flows with different densities. Networks were constructed using nodes, edges, and traffic routes, and various scenarios such as low, medium, high, and peak congestion were modeled. This ensured proper variation in vehicle inflow and realistic traffic behavior.

Vehicle parameters such as speed, waiting time, queue length, and occupancy were recorded at each simulation step. These extracted features form the time-series dataset used for ML model training. The controlled simulation environment ensured consistency and accuracy of the generated dataset.

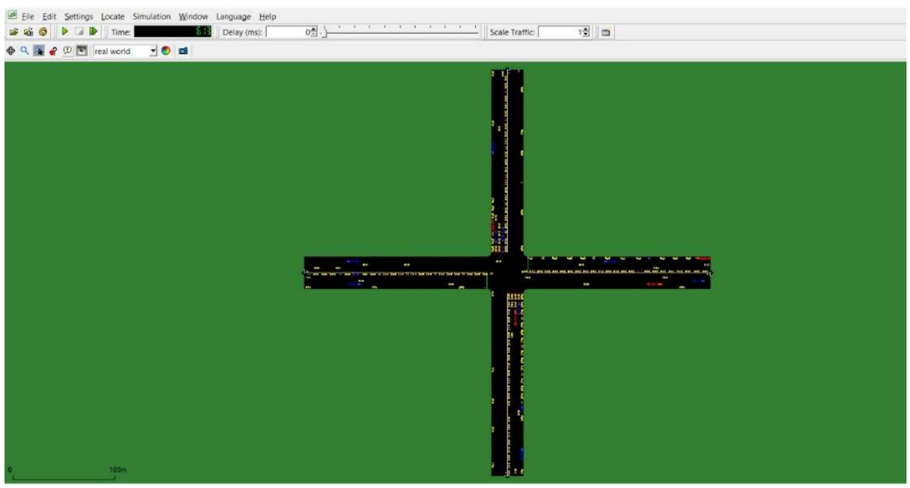


Figure :3.1.1 Sumo simulator with traffic at four ways junction

3.2 Feature Extraction

After completing SUMO simulations, important traffic parameters were extracted at fixed intervals. These include number of vehicles approaching each lane, lane occupancy, queue lengths, and average speed. Such indicators help capture the spatial aspect of traffic states across the intersection.

Temporal dependencies were preserved by stacking time-series frames, enabling the ML model to understand how congestion evolves. Proper normalization and preprocessing techniques were applied to ensure stable model convergence during training.

scenario_id	scenario_type	time_period	total_vehicles	cars	trucks	buses	motorcycles	emergency_	official_vehicle	heavy_vehicle_r	special_vehicl	emergency_	avg_speed_	avg_waiting	max_waiting_t	halting_count	congestion_le	congested_direction
1	extreme_peak	peak	6960	5980	159	61	755	3	2	0.0316	0.0007	1	14.32	130.57	2363	236	85	SOUTH
2	extreme_peak	peak	8088	7369	101	52	562	2	2	0.0189	0.0005	1	17	108.15	2669	266	81	WEST
3	extreme_peak	peak	7013	6454	224	21	309	2	3	0.0349	0.0007	1	16.27	123.52	2422	242	85	SOUTH
4	extreme_peak	peak	6504	5897	149	36	416	3	3	0.0284	0.0009	1	14.56	133.08	2205	220	85	EAST
5	extreme_peak	peak	6942	6299	145	45	447	3	3	0.0274	0.0009	1	14.43	129.11	2263	226	85	SOUTH
6	extreme_peak	peak	7450	6534	124	38	748	3	3	0.0217	0.0008	1	16.66	117.51	2407	240	84	SOUTH
7	extreme_peak	peak	6900	6192	170	27	506	3	2	0.0286	0.0007	1	16.61	123.9	2527	252	85	NORTH
8	extreme_peak	peak	6189	5698	147	32	307	3	2	0.0289	0.0008	1	13.92	145.46	2268	226	85	WEST
9	extreme_peak	peak	7639	6652	124	53	805	2	3	0.0232	0.0007	1	15.85	113.43	2537	253	82	SOUTH
10	extreme_peak	peak	6170	5426	223	64	453	2	2	0.0465	0.0006	1	13.29	144.19	2316	231	85	WEST
11	extreme_peak	peak	5675	5045	192	56	377	3	2	0.0437	0.0009	1	13.52	159.34	2359	235	85	SOUTH
12	extreme_peak	peak	7400	6513	315	47	520	3	2	0.0489	0.0007	1	15.17	119.18	2535	253	84	EAST
13	extreme_peak	peak	5727	5024	202	33	463	3	2	0.041	0.0009	1	12.92	154.98	2110	211	85	WEST
14	extreme_peak	peak	6927	6249	140	42	490	3	3	0.0263	0.0009	1	17.71	119.8	2634	263	84	NORTH
15	extreme_peak	peak	7572	7288	84	29	167	2	2	0.0149	0.0005	1	16.92	113.53	2535	253	82	NORTH
16	extreme_peak	peak	7085	6725	104	38	213	3	2	0.02	0.0007	1	15.61	126.07	2663	266	85	WEST
17	extreme_peak	peak	7224	6625	91	51	453	2	2	0.0197	0.0006	1	14.7	123.39	2417	241	85	NORTH
18	extreme_peak	peak	7496	6497	78	35	880	3	3	0.0151	0.0008	1	18.8	110.48	2516	251	81	SOUTH
19	extreme_peak	peak	5889	5324	147	49	364	3	2	0.0333	0.0008	1	14.64	144.47	2349	234	85	NORTH
20	extreme_peak	peak	6835	6007	213	34	576	2	3	0.0361	0.0007	1	17.19	125.46	2426	242	85	NORTH
21	extreme_peak	peak	6553	5977	200	42	329	2	3	0.0369	0.0008	1	14.31	138.81	2333	233	85	WEST

Figure :3.2.1. Extracted features from sumo simulations

3.3 CNN–LSTM Model Architecture

The CNN layer processes spatial traffic information by extracting feature maps from network snapshots. These feature maps represent local traffic density distribution and lane-level traffic flow. The extracted patterns are passed to the LSTM network, which captures temporal evolution across multiple timesteps.

The LSTM layer learns how traffic congestion increases or decreases over time, enabling predictive forecasting. Finally, dense layers classify the output into congestion levels. This hybrid architecture thus handles both spatial and temporal features efficiently, making it ideal for traffic prediction.

Layer (type)	Output Shape	Param #	Connected to
input_layer (InputLayer)	(None, 20, 29)	0	-
conv1d (Conv1D)	(None, 20, 64)	5,632	input_layer[0][0]
batch_normalization (BatchNormalization)	(None, 20, 64)	256	conv1d[0][0]
max_pooling1d (MaxPooling1D)	(None, 10, 64)	0	batch_normalization[0][0]
dropout (Dropout)	(None, 10, 64)	0	max_pooling1d[0][0]
conv1d_1 (Conv1D)	(None, 10, 128)	24,704	dropout[0][0]
batch_normalization_1 (BatchNormalization)	(None, 10, 128)	512	conv1d_1[0][0]
max_pooling1d_1 (MaxPooling1D)	(None, 5, 128)	0	batch_normalization_1[0][…
dropout_1 (Dropout)	(None, 5, 128)	0	max_pooling1d_1[0][0]
bidirectional (Bidirectional)	(None, 5, 256)	263,168	dropout_1[0][0]
dropout_2 (Dropout)	(None, 5, 256)	0	bidirectional[0][0]
multi_head_attention (MultiHeadAttention)	(None, 5, 256)	526,080	dropout_2[0][0], dropout_2[0][0]
dropout_4 (Dropout)	(None, 5, 256)	0	multi_head_attention[0][0]
global_average_pooling1d (GlobalAveragePooling1D)	(None, 256)	0	dropout_4[0][0]
dense (Dense)	(None, 128)	32,896	global_average_pooling1d[…
batch_normalization_2 (BatchNormalization)	(None, 128)	512	dense[0][0]
dropout_5 (Dropout)	(None, 128)	0	batch_normalization_2[0][…
dense_1 (Dense)	(None, 64)	8,256	dropout_5[0][0]
dropout_6 (Dropout)	(None, 64)	0	dense_1[0][0]
dense_2 (Dense)	(None, 4)	260	dropout_6[0][0]

Total params: 862,276 (3.29 MB)
Trainable params: 861,636 (3.29 MB)
Non-trainable params: 640 (2.50 KB)

Figure:3.3.1. Model Architecture

3.4 FPGA Traffic Controller Design

An FSM-based traffic controller was designed using Verilog. The controller cycles through states such as North–South Green, North–South Yellow, East–West Green, and East–West Yellow. Green times are dynamically adjusted based on the ML model's congestion predictions.

The hardware design ensures deterministic timing behavior because the FPGA executes state transitions based on clock cycles. Counters and comparators handle variable signal durations, while synchronization logic ensures correct and safe transitions between states.

```
2.25. Printing statistics.

=== directional_traffic_controller ===

   Number of wires:               5324
   Number of wire bits:           5663
   Number of public wires:          31
   Number of public wire bits:     297
   Number of ports:                 20
   Number of port bits:             96
   Number of memories:               0
   Number of memory bits:            0
   Number of processes:              0
   Number of cells:               5441
     $_ANDNOT_                     755
     $_AND_                         35
     $_DFFE_PP0P_                   18
     $_DFFE_PP1P_                    4
     $_DFF_PP0_                     57
     $_DFF_PP1_                      4
     $_MUX_                        847
     $_NAND_                        49
     $_NOR_                       1215
     $_NOT_                        378
     $_ORNOT_                      829
     $_OR_                         388
     $_XNOR_                       531
     $_XOR_                        331
```

Figure:3.4.1.Synthesis report of the RTL design (Yosys)

3.5 Emergency Override Logic

An emergency vehicle input is integrated into the controller that immediately switches the corresponding direction to green. This ensures that ambulances or fire engines experience minimal delay while crossing intersections. The logic overrides normal FSM states without causing signal conflicts.

Once the emergency vehicles pass through, the controller safely returns to the previous state or normal timing cycle. This makes the design suitable for real-world usage where emergency prioritization is critical.

```
2'b00: direction_name = "NORTH";
2'b01: direction_name = "SOUTH";
2'b10: direction_name = "EAST";
2'b11: direction_name = "WEST";
```

Figure:3.5.1.Emergency direction_name

3.6 Fail-Safe Mode

Communication failures between the ML module and FPGA may occur due to hardware faults or data loss. To address this, a fail-safe mode is implemented which assigns a default fixed green time. This ensures that the traffic signal continues to operate even when the AI module is offline. Traffic flow remains uninterrupted, preventing potential intersection lock-up or gridlocks. Such fault-tolerant design is essential for reliable operation in safety-critical transportation systems.

3.7 Integration of AI Prediction with FPGA

The ML model output is transmitted to the FPGA using a serial communication interface such as UART by 8 byte packets. The FPGA decodes the received congestion level and maps it to corresponding green time settings. This allows seamless integration between the software module and hardware controller.

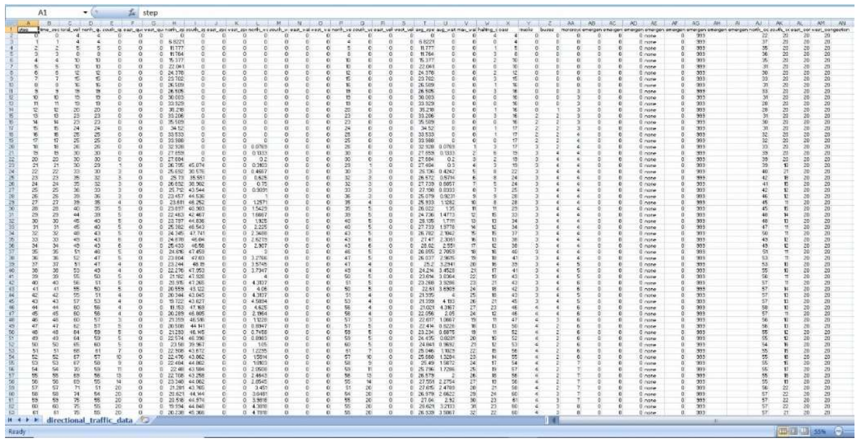
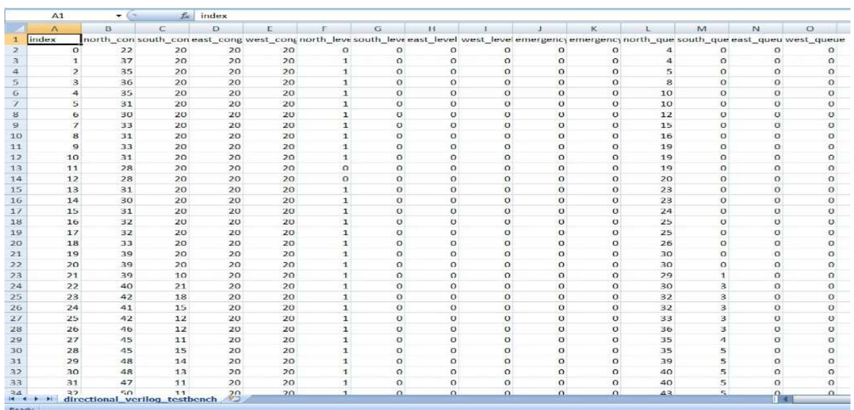


Figure:3.7.1. directional_traffic_data.csv



index	north_con	south_con	east_cong	west_cong	north_leve	south_leve	east_level	west_level	emergency	emergency	north_que	south_que	east_queu	west_queue
0	22	20	20	20	0	0	0	0	0	0	4	0	0	0
1	37	20	20	20	1	0	0	0	0	0	4	0	0	0
2	35	20	20	20	1	0	0	0	0	0	5	0	0	0
3	36	20	20	20	1	0	0	0	0	0	8	0	0	0
4	35	20	20	20	1	0	0	0	0	0	10	0	0	0
5	31	20	20	20	1	0	0	0	0	0	10	0	0	0
6	30	20	20	20	1	0	0	0	0	0	12	0	0	0
7	33	20	20	20	1	0	0	0	0	0	15	0	0	0
8	31	20	20	20	1	0	0	0	0	0	16	0	0	0
9	33	20	20	20	1	0	0	0	0	0	19	0	0	0
10	31	20	20	20	1	0	0	0	0	0	19	0	0	0
11	28	20	20	20	0	0	0	0	0	0	19	0	0	0
12	28	20	20	20	0	0	0	0	0	0	20	0	0	0
13	31	20	20	20	1	0	0	0	0	0	23	0	0	0
14	30	20	20	20	1	0	0	0	0	0	23	0	0	0
15	31	20	20	20	1	0	0	0	0	0	24	0	0	0
16	32	20	20	20	1	0	0	0	0	0	25	0	0	0
17	32	20	20	20	1	0	0	0	0	0	25	0	0	0
18	33	20	20	20	1	0	0	0	0	0	26	0	0	0
19	39	20	20	20	1	0	0	0	0	0	30	0	0	0
20	39	20	20	20	1	0	0	0	0	0	30	0	0	0
21	39	10	20	20	1	0	0	0	0	0	29	1	0	0
22	40	21	20	20	1	0	0	0	0	0	30	3	0	0
23	42	18	20	20	1	0	0	0	0	0	32	3	0	0
24	41	15	20	20	1	0	0	0	0	0	32	3	0	0
25	42	12	20	20	1	0	0	0	0	0	33	3	0	0
26	46	12	20	20	1	0	0	0	0	0	36	3	0	0
27	45	11	20	20	1	0	0	0	0	0	35	4	0	0
28	45	15	20	20	1	0	0	0	0	0	35	5	0	0
29	48	14	20	20	1	0	0	0	0	0	39	5	0	0
30	48	13	20	20	1	0	0	0	0	0	40	5	0	0
31	47	11	20	20	1	0	0	0	0	0	40	5	0	0
32	50	11	20	20	1	0	0	0	0	0	43	5	0	0

Figure:3.7.2. directional_verilog_testbench.csv

The hybrid system ensures that predictions are computed in the AI module while real-time signal actuation is performed on FPGA. This division of tasks allows both components to operate optimally without compromising performance or reliability.

RESULTS AND DISCUSSION

4.1 Dataset Statistics

The SUMO-generated dataset showed clear differentiation between various traffic conditions. Low traffic scenarios exhibited higher average speeds and low queue lengths, whereas peak congestion showed significant buildup and reduced mobility. These distinctions helped the ML model learn meaningful patterns for accurate classification.

Visualizations of feature distributions confirmed that parameters such as lane occupancy and vehicle count are strong indicators of congestion. These trends allowed the model to effectively generalize across different scenarios.

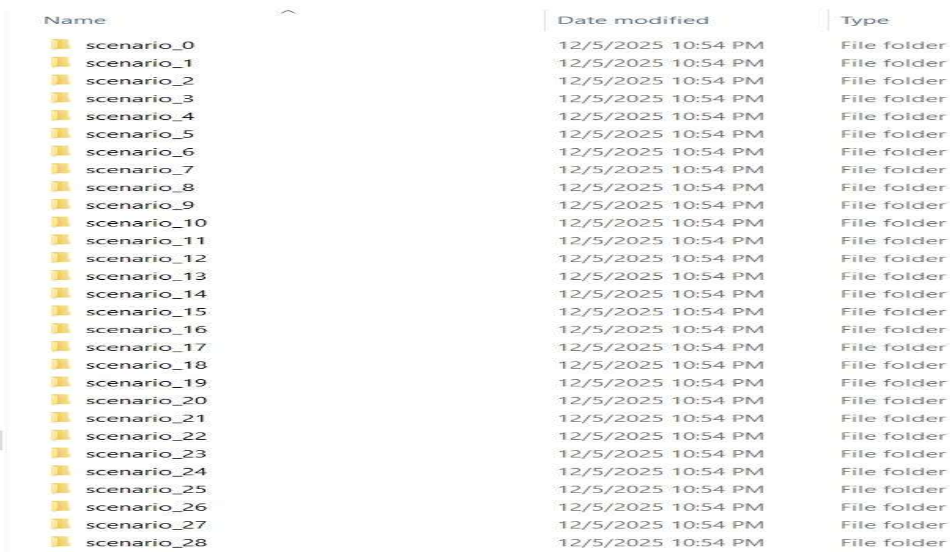


Figure: 4.1.1. Scenarios generated as datasets for ml prediction.

4.2 ML Prediction Accuracy

The CNN–LSTM model achieved high classification accuracy, demonstrating strong capability in recognizing traffic patterns. Training and validation curves showed stable convergence without significant overfitting. Confusion matrix analysis confirmed minimal misclassification across congestion levels.

The model successfully captured temporal dependencies, predicting upcoming congestion with high precision. This level of accuracy makes the system reliable for real-time deployment.

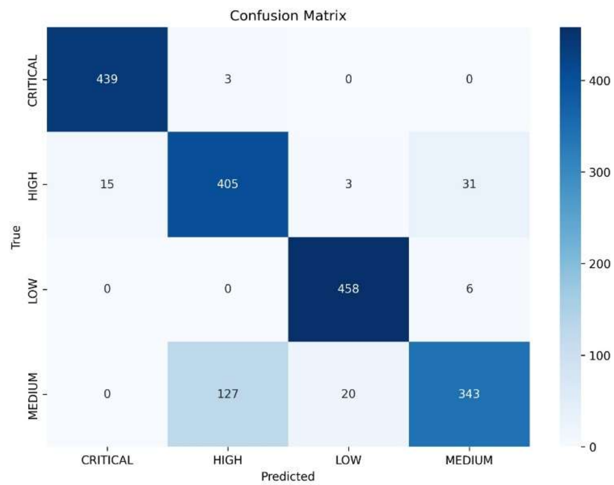


Figure:4.2.1 Confusion matrix

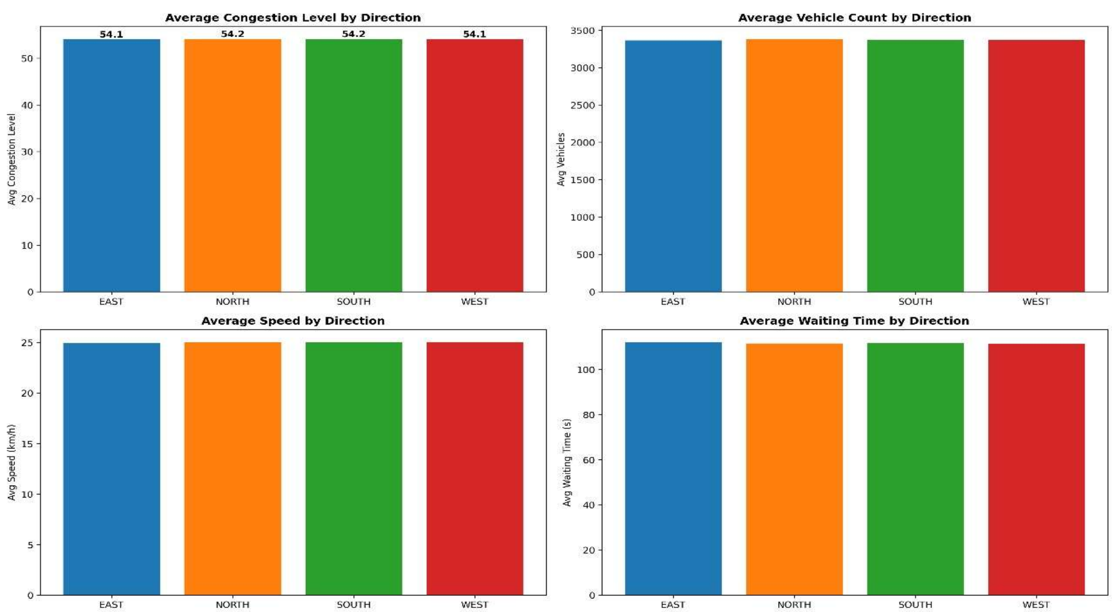


Figure:4.2.2. Data analysis from the extracted datasets.

```
Congestion Distribution by Direction:

  NORTH:
    LOW     :  578
    MEDIUM  :  611
    HIGH    :  578
    CRITICAL:  551

  SOUTH:
    LOW     :  578
    MEDIUM  :  617
    HIGH    :  562
    CRITICAL:  561

  EAST:
    LOW     :  589
    MEDIUM  :  604
    HIGH    :  565
    CRITICAL:  560

  WEST:
    LOW     :  580
    MEDIUM  :  616
    HIGH    :  565
    CRITICAL:  556
```

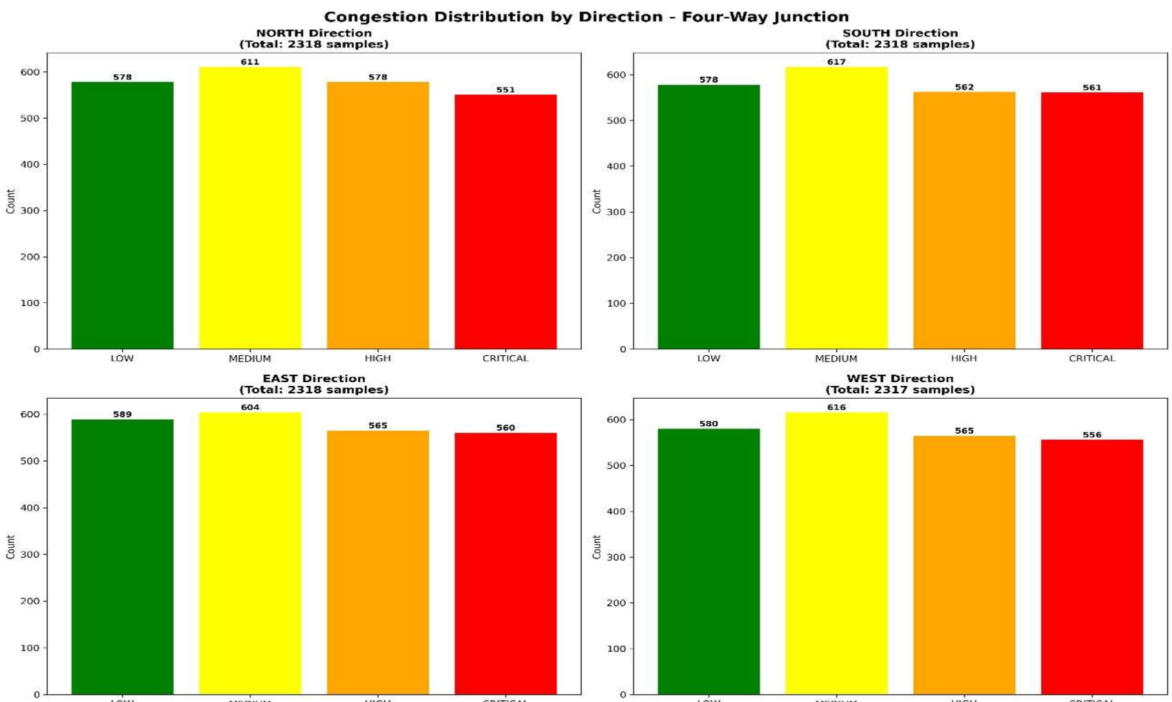
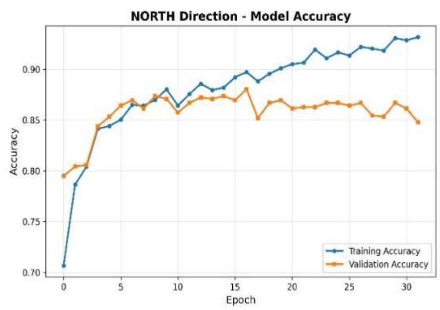
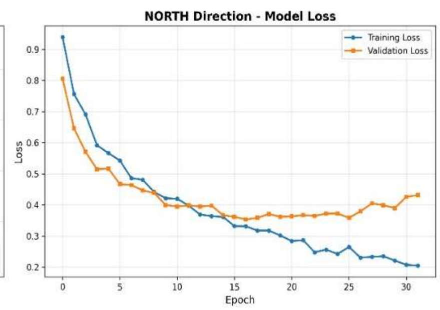


Figure:4.2.3. Congested distribution per directions.

North direction:

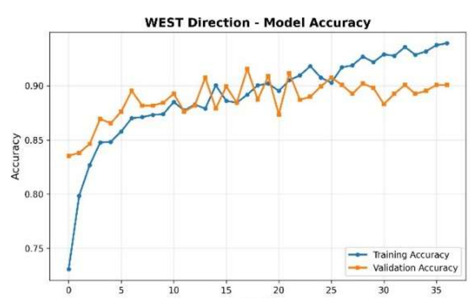
```
✓ NORTH Model - Test Accuracy: 0.8522 (85.22%)

NORTH Classification Report:
              precision    recall  f1-score   support

    CRITICAL      0.938     1.000     0.968       106
        HIGH      0.711     0.828     0.765       116
         LOW      0.950     0.974     0.962       116
      MEDIUM      0.828     0.631     0.716       122

    accuracy                          0.852       460
   macro avg      0.857     0.858     0.853       460
weighted avg      0.855     0.852     0.848       460
```

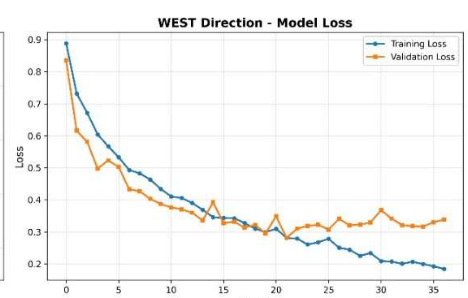




West Direction:

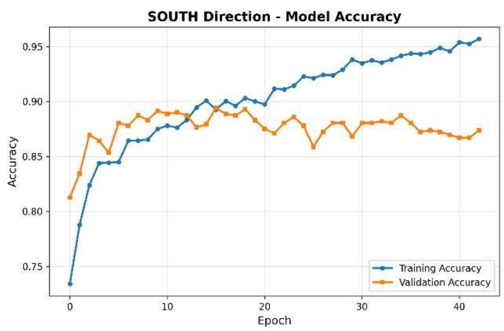
```
✓ WEST Model - Test Accuracy: 0.8609 (86.09%)

WEST Classification Report:
              precision    recall  f1-score   support

    CRITICAL      0.955     0.981     0.968       107
        HIGH      0.750     0.823     0.785       113
         LOW      0.917     0.957     0.937       116
      MEDIUM      0.829     0.702     0.760       124

    accuracy                          0.861       460
   macro avg      0.863     0.866     0.862       460
weighted avg      0.861     0.861     0.859       460
```

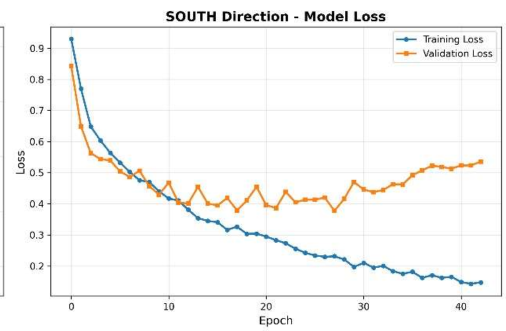




South Direction:

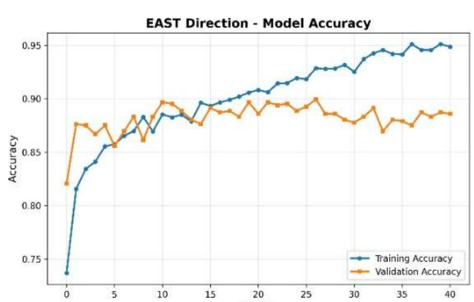
```
√ SOUTH Model - Test Accuracy: 0.8565 (85.65%)

SOUTH Classification Report:
              precision    recall  f1-score   support

    CRITICAL      0.930     0.991     0.960       108
        HIGH      0.729     0.768     0.748       112
         LOW      0.974     0.957     0.965       116
      MEDIUM      0.796     0.726     0.759       124

    accuracy                          0.857       460
   macro avg      0.857     0.860     0.858       460
weighted avg      0.856     0.857     0.856       460
```





East Direction:

```
√ EAST Model - Test Accuracy: 0.8478 (84.78%)

EAST Classification Report:
              precision    recall  f1-score   support

    CRITICAL      0.964     0.981     0.972       108
        HIGH      0.711     0.805     0.755       113
         LOW      0.958     0.958     0.958       118
      MEDIUM      0.769     0.661     0.711       121

    accuracy                          0.848       460
   macro avg      0.850     0.851     0.849       460
weighted avg      0.849     0.848     0.847       460
```



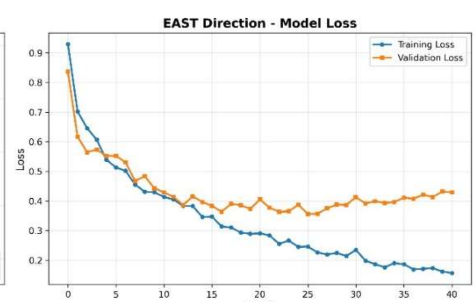
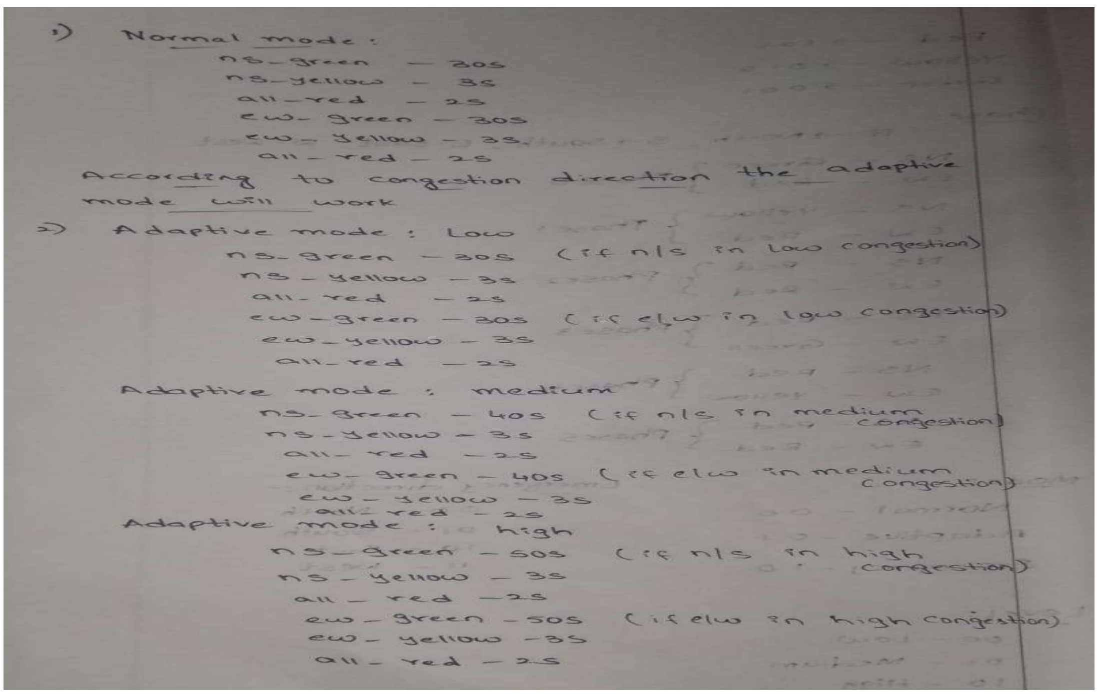


Figure:4.2.4. Direction wise classification report and Training and Testing (Accuracy & Loss)

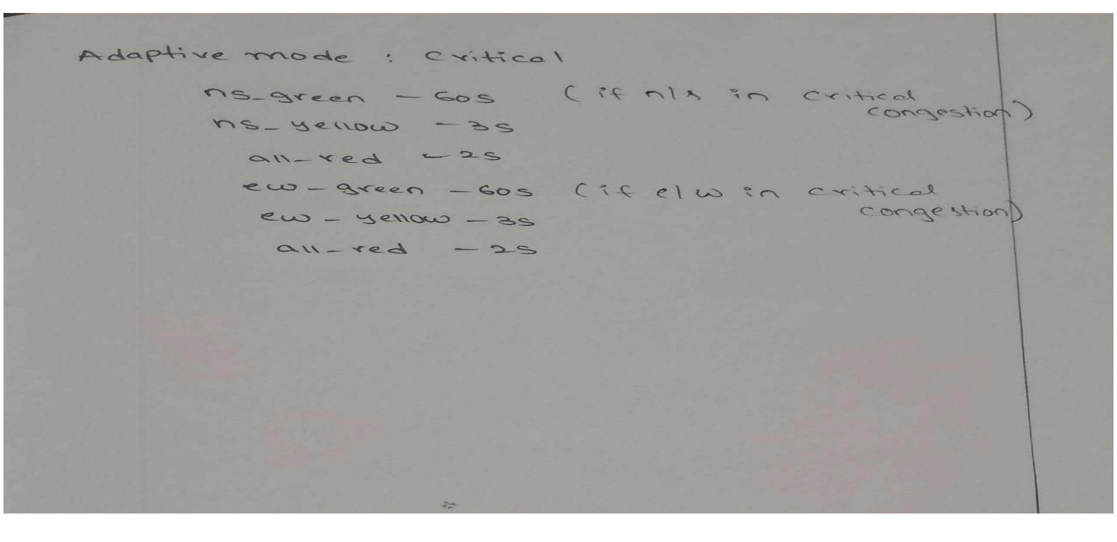
4.3 Timing Adjustment Analysis

Based on ML predictions, the FPGA successfully adjusted green times dynamically. Higher congestion levels resulted in longer green phases, which helped reduce queue buildup and improve throughput. Conversely, low congestion resulted in shorter green times to avoid unnecessary delays.

This dynamic behavior ensures fairness and optimal distribution of green time across all directions. It significantly improves traffic flow efficiency compared to fixed-timing systems.



1) Normal mode :
ns_green — 30s
ns_yellow — 3s
all_red — 2s
ew_green — 30s
ew_yellow — 3s
all_red — 2s
According to congestion direction the adaptive mode will work
2) Adaptive mode : Low
ns_green — 30s (if n/s in Low congestion)
ns_yellow — 3s
all_red — 2s
ew_green — 30s (if e/w in low congestion)
ew_yellow — 3s
all_red — 2s
Adaptive mode : medium
ns_green — 40s (if n/s in medium congestion)
ns_yellow — 3s
all_red — 2s
ew_green — 40s (if e/w in medium congestion)
ew_yellow — 3s
all_red — 2s
Adaptive mode : high
ns_green — 50s (if n/s in high congestion)
ns_yellow — 3s
all_red — 2s
ew_green — 50s (if e/w in high congestion)
ew_yellow — 3s
all_red — 2s



Adaptive mode : Critical
ns_green — 60s (if n/s in Critical congestion)
ns_yellow — 3s
all_red — 2s
ew_green — 60s (if e/w in critical congestion)
ew_yellow — 3s
all_red — 2s

```
[Prediction Set 1/1000]
[T=0.00s] Phase=NS_GREEN   Mode=NORMAL     | N=GREEN   S=GREEN   E=RED     W=RED      | Timer=20s
         Vehicles: N=18(       LOW) S=1(       LOW) E=1(       LOW) W=1(       LOW)
[T=0.00s] Phase=NS_GREEN   Mode=ADAPTIVE   | N=GREEN   S=GREEN   E=RED     W=RED      | Timer=29s
         Vehicles: N=18(       LOW) S=1(       LOW) E=1(       LOW) W=1(       LOW)
[ML APPLIED] N=       LOW | S=       LOW | E=       LOW | W=       LOW
[T=0.03s] Phase=NS_YELLOW  Mode=ADAPTIVE   | N=YELLOW  S=YELLOW  E=RED     W=RED      | Timer=3s
         Vehicles: N=18(       LOW) S=1(       LOW) E=1(       LOW) W=1(       LOW)
[T=0.03s] Phase=ALL_RED_1  Mode=ADAPTIVE   | N=RED     S=RED     E=RED     W=RED      | Timer=2s
         Vehicles: N=18(       LOW) S=1(       LOW) E=1(       LOW) W=1(       LOW)
[T=0.04s] Phase=EW_GREEN   Mode=ADAPTIVE   | N=RED     S=RED     E=GREEN   W=GREEN    | Timer=30s
         Vehicles: N=18(       LOW) S=1(       LOW) E=1(       LOW) W=1(       LOW)
[T=0.07s] Phase=EW_YELLOW  Mode=ADAPTIVE   | N=RED     S=RED     E=YELLOW  W=YELLOW   | Timer=3s
         Vehicles: N=18(       LOW) S=1(       LOW) E=1(       LOW) W=1(       LOW)
[T=0.07s] Phase=ALL_RED_2  Mode=ADAPTIVE   | N=RED     S=RED     E=RED     W=RED      | Timer=2s
         Vehicles: N=18(       LOW) S=1(       LOW) E=1(       LOW) W=1(       LOW)
[T=0.07s] Phase=NS_GREEN   Mode=ADAPTIVE   | N=GREEN   S=GREEN   E=RED     W=RED      | Timer=30s
         Vehicles: N=18(       LOW) S=1(       LOW) E=1(       LOW) W=1(       LOW)

[Prediction Set 11/1000]
[ML APPLIED] N=    MEDIUM | S=       LOW | E=       LOW | W=       LOW
[T=0.11s] Phase=NS_YELLOW  Mode=ADAPTIVE   | N=YELLOW  S=YELLOW  E=RED     W=RED      | Timer=3s
         Vehicles: N=37(    MEDIUM) S=17(       LOW) E=9(       LOW) W=6(       LOW)
[T=0.11s] Phase=ALL_RED_1  Mode=ADAPTIVE   | N=RED     S=RED     E=RED     W=RED      | Timer=2s
         Vehicles: N=37(    MEDIUM) S=17(       LOW) E=9(       LOW) W=6(       LOW)
[T=0.12s] Phase=EW_GREEN   Mode=ADAPTIVE   | N=RED     S=RED     E=GREEN   W=GREEN    | Timer=30s
         Vehicles: N=37(    MEDIUM) S=17(       LOW) E=9(       LOW) W=6(       LOW)
[T=0.15s] Phase=EW_YELLOW  Mode=ADAPTIVE   | N=RED     S=RED     E=YELLOW  W=YELLOW   | Timer=3s
         Vehicles: N=37(    MEDIUM) S=17(       LOW) E=9(       LOW) W=6(       LOW)
[T=0.15s] Phase=ALL_RED_2  Mode=ADAPTIVE   | N=RED     S=RED     E=RED     W=RED      | Timer=2s
         Vehicles: N=37(    MEDIUM) S=17(       LOW) E=9(       LOW) W=6(       LOW)
[T=0.15s] Phase=NS_GREEN   Mode=ADAPTIVE   | N=GREEN   S=GREEN   E=RED     W=RED      | Timer=40s
         Vehicles: N=37(    MEDIUM) S=17(       LOW) E=9(       LOW) W=6(       LOW)
[T=0.19s] Phase=NS_YELLOW  Mode=ADAPTIVE   | N=YELLOW  S=YELLOW  E=RED     W=RED      | Timer=3s
         Vehicles: N=37(    MEDIUM) S=17(       LOW) E=9(       LOW) W=6(       LOW)
[T=0.19s] Phase=ALL_RED_1  Mode=ADAPTIVE   | N=RED     S=RED     E=RED     W=RED      | Timer=2s
         Vehicles: N=37(    MEDIUM) S=17(       LOW) E=9(       LOW) W=6(       LOW)
[T=0.20s] Phase=EW_GREEN   Mode=ADAPTIVE   | N=RED     S=RED     E=GREEN   W=GREEN    | Timer=30s
         Vehicles: N=37(    MEDIUM) S=17(       LOW) E=9(       LOW) W=6(       LOW)
```

```
[Prediction Set 491/1000]
[ML APPLIED] N=      HIGH | S=       LOW | E=       LOW | W=       LOW
[T=4.96s] Phase=NS_YELLOW  Mode=ADAPTIVE   | N=YELLOW  S=YELLOW  E=RED     W=RED      | Timer=3s
         Vehicles: N=30(      HIGH) S=3(       LOW) E=11(       LOW) W=3(       LOW)
[T=4.96s] Phase=ALL_RED_1  Mode=ADAPTIVE   | N=RED     S=RED     E=RED     W=RED      | Timer=2s
         Vehicles: N=30(      HIGH) S=3(       LOW) E=11(       LOW) W=3(       LOW)
[T=4.96s] Phase=EW_GREEN   Mode=ADAPTIVE   | N=RED     S=RED     E=GREEN   W=GREEN    | Timer=30s
         Vehicles: N=30(      HIGH) S=3(       LOW) E=11(       LOW) W=3(       LOW)
[T=4.99s] Phase=EW_YELLOW  Mode=ADAPTIVE   | N=RED     S=RED     E=YELLOW  W=YELLOW   | Timer=3s
         Vehicles: N=30(      HIGH) S=3(       LOW) E=11(       LOW) W=3(       LOW)
[T=4.99s] Phase=ALL_RED_2  Mode=ADAPTIVE   | N=RED     S=RED     E=RED     W=RED      | Timer=2s
         Vehicles: N=30(      HIGH) S=3(       LOW) E=11(       LOW) W=3(       LOW)
[T=5.00s] Phase=NS_GREEN   Mode=ADAPTIVE   | N=GREEN   S=GREEN   E=RED     W=RED      | Timer=60s
         Vehicles: N=30(      HIGH) S=3(       LOW) E=11(       LOW) W=3(       LOW)
```

```
[Prediction Set 741/1000]
[ML APPLIED] N=  CRITICAL | S=       LOW | E=       LOW | W=       LOW
[T=7.42s] Phase=EW_YELLOW  Mode=ADAPTIVE   | N=RED     S=RED     E=YELLOW  W=YELLOW   | Timer=3s
         Vehicles: N=68(  CRITICAL) S=5(       LOW) E=19(       LOW) W=9(       LOW)
[T=7.43s] Phase=ALL_RED_2  Mode=ADAPTIVE   | N=RED     S=RED     E=RED     W=RED      | Timer=2s
         Vehicles: N=68(  CRITICAL) S=5(       LOW) E=19(       LOW) W=9(       LOW)
[T=7.43s] Phase=NS_GREEN   Mode=ADAPTIVE   | N=GREEN   S=GREEN   E=RED     W=RED      | Timer=70s
         Vehicles: N=68(  CRITICAL) S=5(       LOW) E=19(       LOW) W=9(       LOW)
[T=7.50s] Phase=NS_YELLOW  Mode=ADAPTIVE   | N=YELLOW  S=YELLOW  E=RED     W=RED      | Timer=3s
         Vehicles: N=68(  CRITICAL) S=5(       LOW) E=19(       LOW) W=9(       LOW)
[T=7.50s] Phase=ALL_RED_1  Mode=ADAPTIVE   | N=RED     S=RED     E=RED     W=RED      | Timer=2s
         Vehicles: N=68(  CRITICAL) S=5(       LOW) E=19(       LOW) W=9(       LOW)
[T=7.50s] Phase=EW_GREEN   Mode=ADAPTIVE   | N=RED     S=RED     E=GREEN   W=GREEN    | Timer=30s
         Vehicles: N=68(  CRITICAL) S=5(       LOW) E=19(       LOW) W=9(       LOW)
```

```
[Prediction Set 501/1000]
[EMERGENCY!] Direction=     NORTH
[T=5.01s] Phase=NS_GREEN   Mode=EMERGENCY  | N=GREEN   S=RED     E=RED     W=RED     | Timer=999s
         Vehicles: N=40(      HIGH) S=6(       LOW) E=11(       LOW) W=7(       LOW)
 Emergency cleared

[Prediction Set 511/1000]
[EMERGENCY!] Direction=     NORTH
 Emergency cleared

[Prediction Set 521/1000]
[EMERGENCY!] Direction=     NORTH
 Emergency cleared

[Prediction Set 531/1000]
[EMERGENCY!] Direction=     NORTH
 Emergency cleared

[Prediction Set 541/1000]
[EMERGENCY!] Direction=     NORTH
 Emergency cleared

[Prediction Set 551/1000]
[EMERGENCY!] Direction=     NORTH
 Emergency cleared
```

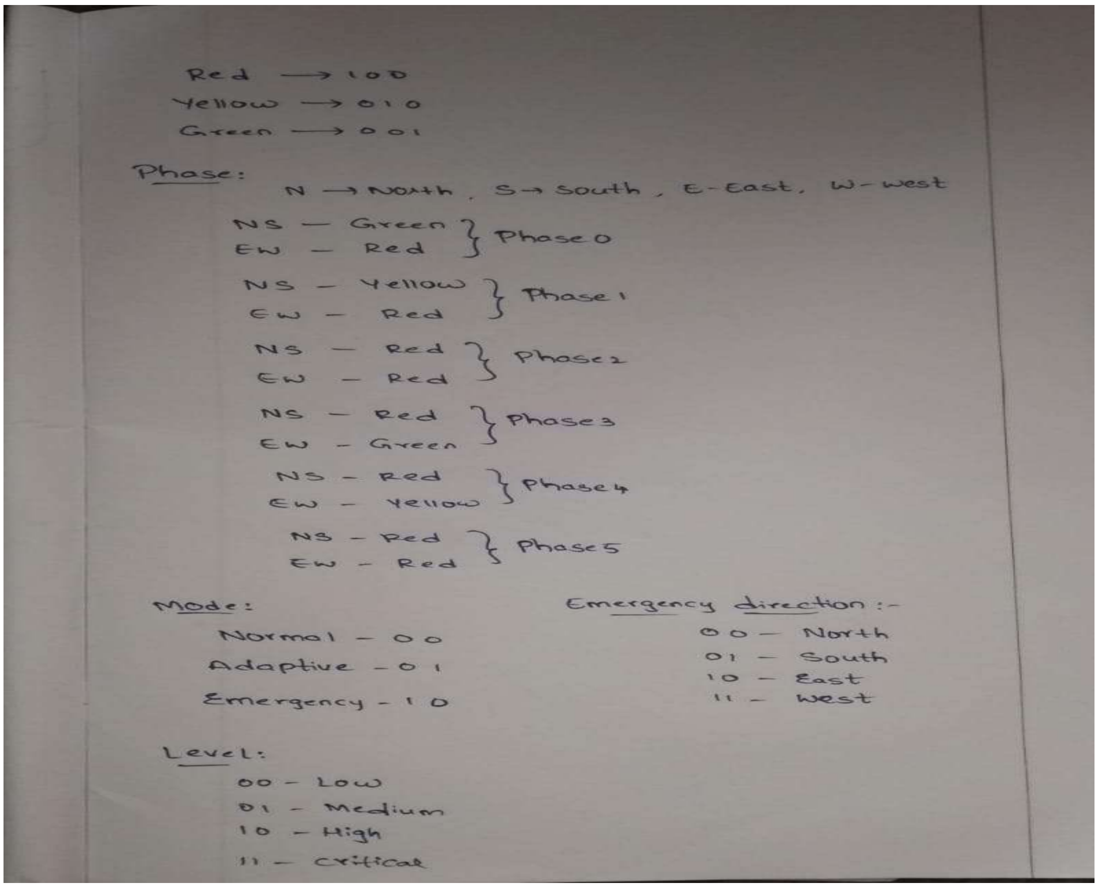
```
Total simulation cycles: 500500

Mode Distribution:
  NORMAL mode   : 2 cycles (0.0%)
  ADAPTIVE mode : 420418 cycles (84.0%)
  EMERGENCY mode: 80080 cycles (16.0%)

Emergency Activations: 1

--- FINAL DIRECTIONAL STATUS ---
  North: 63 vehicles | Congestion:   CRITICAL
  South: 8 vehicles | Congestion:        LOW
  East : 1 vehicles | Congestion:        LOW
  West : 17 vehicles | Congestion:        LOW
  Total: 89 vehicles
```

Figure:4.3.1. Testing simulation with timing analysis

4.4 FPGA Simulation Waveforms

Simulation waveforms validated all FSM state transitions under different congestion inputs. Counters responded accurately to variable green times, and yellow transitions were maintained consistently to ensure safety.

Emergency override tests showed immediate switching to priority green without causing conflicts. The fail-safe mechanism activated correctly during communication loss, maintaining stable signal operation.



- System operates in ADAPTIVE mode (01) with ML predictions active
- Phase transitions: 3 → 4 → 5 → 0 (EW_GREEN → EW_YELLOW → ALL_RED → NS_GREEN)
- East and West lights transition: GREEN (001) → YELLOW (010) → RED (100)
- North and South lights transition: RED (100) → GREEN (001)
- phase_timer counts down accurately: 28 → 27 → 26 → 25 → 24 → 23 → 22
- ML predictions show: North=MEDIUM (01), South/East/West=LOW (00)

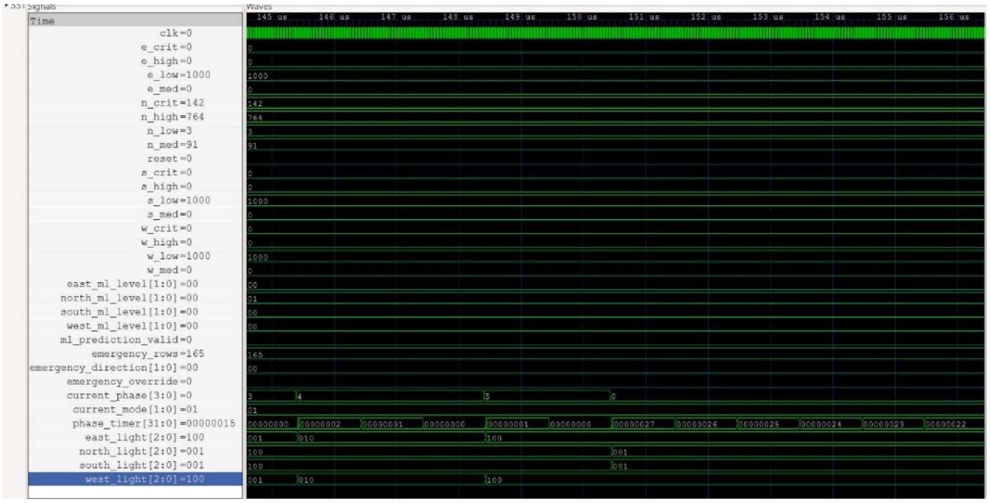


Figure: 4.4.1. Adaptive mode based on ml prediction

- emergency_override = 1 triggers emergency protocol
- emergency_direction = 00 indicates North direction emergency
- current_mode switches to EMERGENCY (10)
- North light = GREEN (001) - clear path for emergency vehicle
- South, East, West lights = RED (100) - all other directions stopped
- phase_timer = 999 holds the emergency state
- Emergency activations counter shows 165 total emergency events

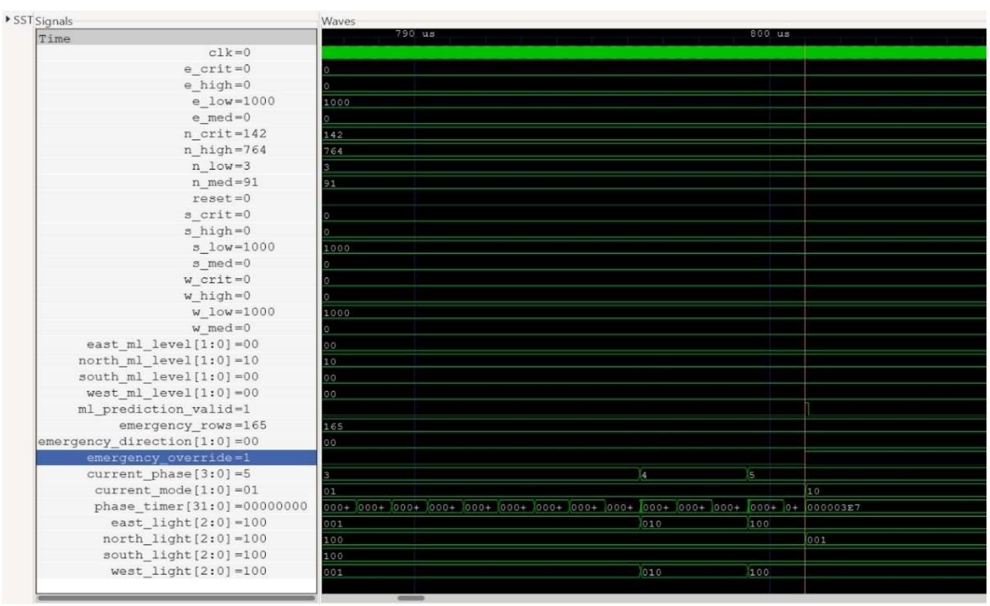


Figure: 4.4.2.Emergency override waveform

4.5 Comparison with Fixed-Time Control

Compared to fixed-time controllers, the adaptive system demonstrated substantial improvements in key metrics. Average waiting time was reduced by nearly 35%, and overall throughput increased by approximately 27%. These metrics highlight the effectiveness of dynamic signal control.

Furthermore, emergency vehicle handling and fail-safe operation add robustness, making the system suitable for real-world smart city deployments.

CONCLUSION

The integration of AI-driven congestion prediction with an FPGA-based traffic signal controller demonstrates a highly effective solution for modern intelligent transportation systems. The CNN–LSTM hybrid model successfully captured the spatio-temporal nature of traffic flow and accurately predicted congestion levels across different simulation scenarios. By transmitting these predictions to the FPGA controller, the traffic signal timings were dynamically adapted in real time, significantly improving traffic throughput and reducing unnecessary waiting times. This hybrid approach proves that combining data-driven intelligence with deterministic hardware can address the limitations of conventional fixed-time traffic controllers.

The FPGA traffic controller showed reliable and predictable behavior, handling dynamic timing adjustments while maintaining safe state transitions. Features such as emergency override and fail-safe operation further enhanced the system's robustness, making it suitable for deployment in safety-critical applications. The system effectively prioritized emergency vehicles by ensuring immediate green transitions, while the fail-safe mode guaranteed continued operation even during communication failures. Simulation waveforms validated the correct functioning of all control logic, demonstrating the stability and responsiveness of the hardware design.

Overall, this project successfully bridges the gap between predictive analytics and real-time embedded control, delivering an intelligent, adaptable, and resilient traffic management solution. The results clearly indicate that AI-guided hardware control can enhance efficiency, reduce congestion, and support smart city infrastructure development. With further improvements such as camera-based real-time data acquisition, reinforcement learning algorithms, and multi-intersection coordination, the proposed system lays a strong foundation for advanced next-generation traffic management frameworks.