GROUP-70

Soft Computing

Traffic Flow forecasting using ANNs and meta heuristic algorithms.

G.V RITESH RAMAN (22135049)
ARPIT SINGHAL (22134010)
TANMANA MISHRA (22135133)

CONTEXT

Traffic congestion is rising in cities around the world. Contributing factors include **expanding urban populations**, **aging infrastructure**, inefficient and **uncoordinated traffic signal timing** and a lack of real-time data. The impacts are significant. Traffic data and analytics company **INRIX** estimates that traffic congestion cost U.S. commuters **$305 billion in 2017 due to wasted fuel**, **lost time and the increased cost of transporting goods** through congested areas. Given the physical and financial limitations around building additional roads, cities must use new strategies and technologies to improve traffic conditions.

DATASET

This dataset contains **48.1k (48120)** observations of the number of vehicles each hour in four different junctions:

1) DateTime
2) Junction
3) Vehicles
4) ID

The sensors on each of these junctions were collecting data at different times, hence you will see traffic data from different time periods. Some of the junctions have provided limited or sparse data requiring thoughtfulness when creating future projections.

OBJECTIVE

Forecast hourly vehicle count at 4 junctions.

- **Approach:**
  LSTM Model:
  -Single-step prediction: Next day's traffic
  -Repeated single-step: Multiple future days
- **Evaluation**: Mean Squared Error (MSE)
- **Optimization**:

- Used Particle Swarm Optimization (PSO) to tune LSTM hyperparameters. and retrained LSTM with optimized parameters and trained with them.

- **Result:**

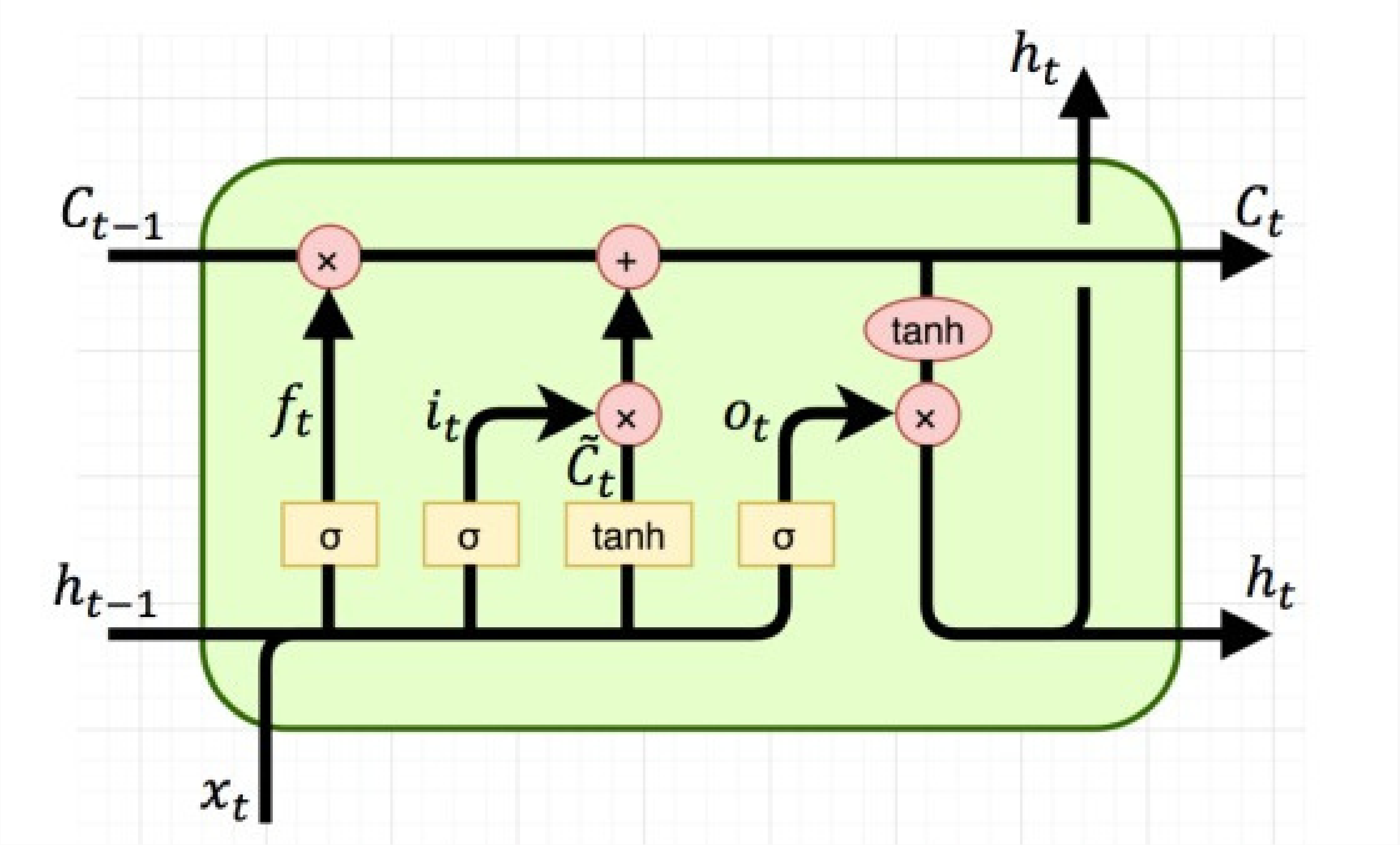
-Compared performance of base LSTM vs PSO-optimized LSTM

ANN

LSTM-Based Traffic Forecasting

- Forecasted hourly vehicle counts at four junctions using LSTM
- **Why LSTM?**
  - Captures long-term patterns in time series data.
  - Solves RNN’s vanishing gradient issues.
- **Components:**
  - Cell state (Ct): Captures long term memory.
  - Hidden state(ht): Captures short term memory.
- **Architecture:**
  - Forget Gate: Discards irrelevant info from cell state Ct
  - Input Gate: Adds useful data to cell state Ct
  - Output Gate: To create the hidden state ht.

LSTM




```python
class SimpleLSTM(nn.Module):
    def __init__(self, output_size, input_size, hidden_size, num_layers, seq_length, dropout=0.2):
        super(SimpleLSTM, self).__init__()
        self.output_size = output_size
        self.num_layers = num_layers
        self.input_size = input_size
        self.hidden_size = hidden_size
        self.seq_length = seq_length
        self.lstm = nn.LSTM(input_size=input_size, hidden_size=hidden_size,
                            num_layers=num_layers, batch_first=True, dropout=dropout)
        self.fc_1 = nn.Linear(hidden_size, 32)
        self.fc = nn.Linear(32, output_size)
        self.relu = nn.ReLU()

    def forward(self, x):
        device = x.device
        h_0 = torch.zeros(self.num_layers, x.size(0), self.hidden_size).to(device)
        c_0 = torch.zeros(self.num_layers, x.size(0), self.hidden_size).to(device)
        output, (hn, cn) = self.lstm(x, (h_0, c_0))
        out = self.relu(output[:, -1, :])
        out = self.fc_1(out)
        out = self.relu(out)
        out = self.fc(out)
        return out.unsqueeze(1)
```

Architecture of LSTM

Our function definition(LSTM)

META-HEURISTIC

PSO-Tuned LSTM For Traffic Forecasting

PSO is inspired by the social behavior of birds flocking or fish schooling. It's used to find optimal or near-optimal solutions to complex problems.

- **Why PSO(Particle Swarm Optimization)?**
  - No gradient needed – works on black-box functions
  - Efficient for hyperparameter tuning and faster convergence.
- **Working:**
  - Swarm of particles explore hyperparameter space
  - Each particle remembers: Personal Best(pBest) & Global Best(gBest)
  - Velocity updates guided by inertia, personal, and global experience
  - Used to Optimize: LSTM, Learning rate, Batch Size, No. of Epochs

NOVELTY

1. Sequential Split for LSTM

- **Why Sequential Split?**
  - LSTM captures temporal patterns — order matters
  - Sequential split mimics real-world forecasting scenarios
- **Advantages of Sequential Split:**
  - Preserves time order — no data leakage
  - Ensures training on past, testing on future
  - Crucial for time series tasks like traffic prediction
  - Avoid Breaking of Temporal Dependencies
- Data is sequentially split along with random split and their MSEs are compared.

NOVELTY

2. Forward Chaining for Time Series Forecasting(similar to K- fold)

What is Forward Chaining Validation?

A time-aware validation method that:

- Trains on past data
- Tests on future data
- Expands training window step by step

Why Use It?

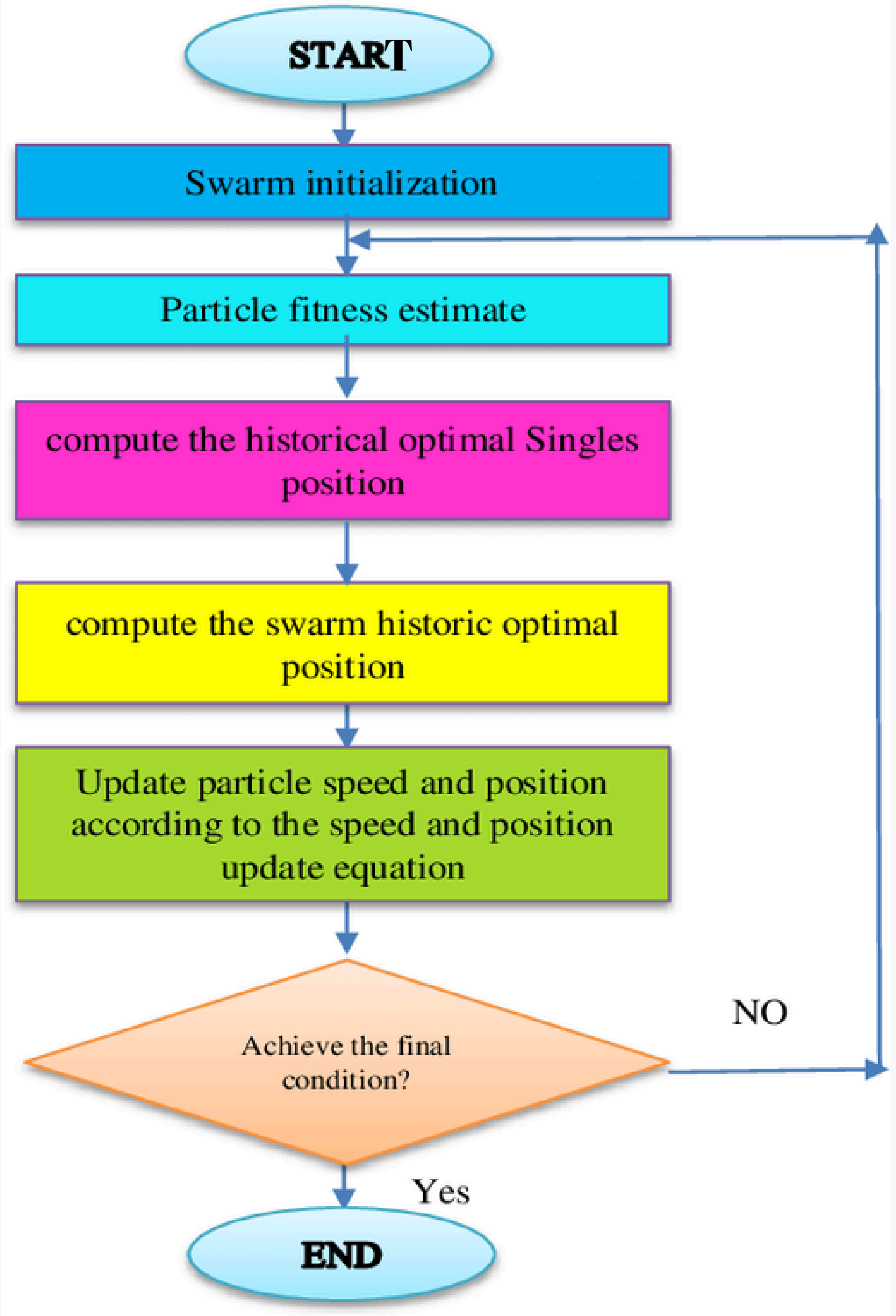
- Maintains temporal order
- Prevents data leakage
- Mimics real-world forecasting
- Adapts to growing datasets

NOVELTY

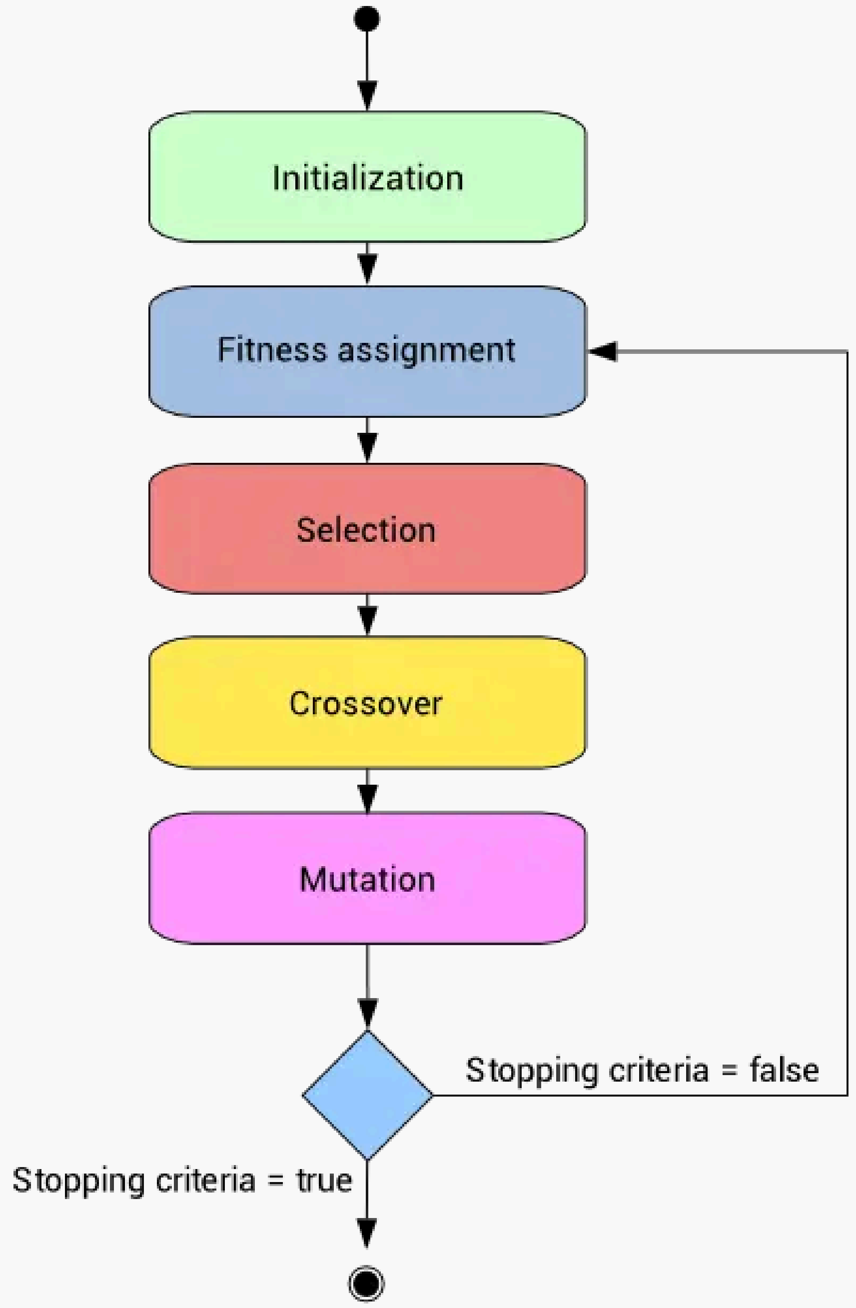
3. Some more meta-heuristic algorithms into picture:

- **Genetic Algorithm (GA):**
  - Mimics evolution: selection, crossover, mutation
  - Efficiently explores high-dimensional parameter space
  - No gradients needed, works on complex search landscapes
- **Bayesian Optimization (BO):**
  - Probabilistic surrogate modeling of loss function
  - Smart exploration vs. exploitation strategy
  - Fewer iterations needed to find optimal parameters
- **Why It's Novel:**
  - Avoids brute-force (grid/random) search
  - Boosts model performance while reducing compute cost

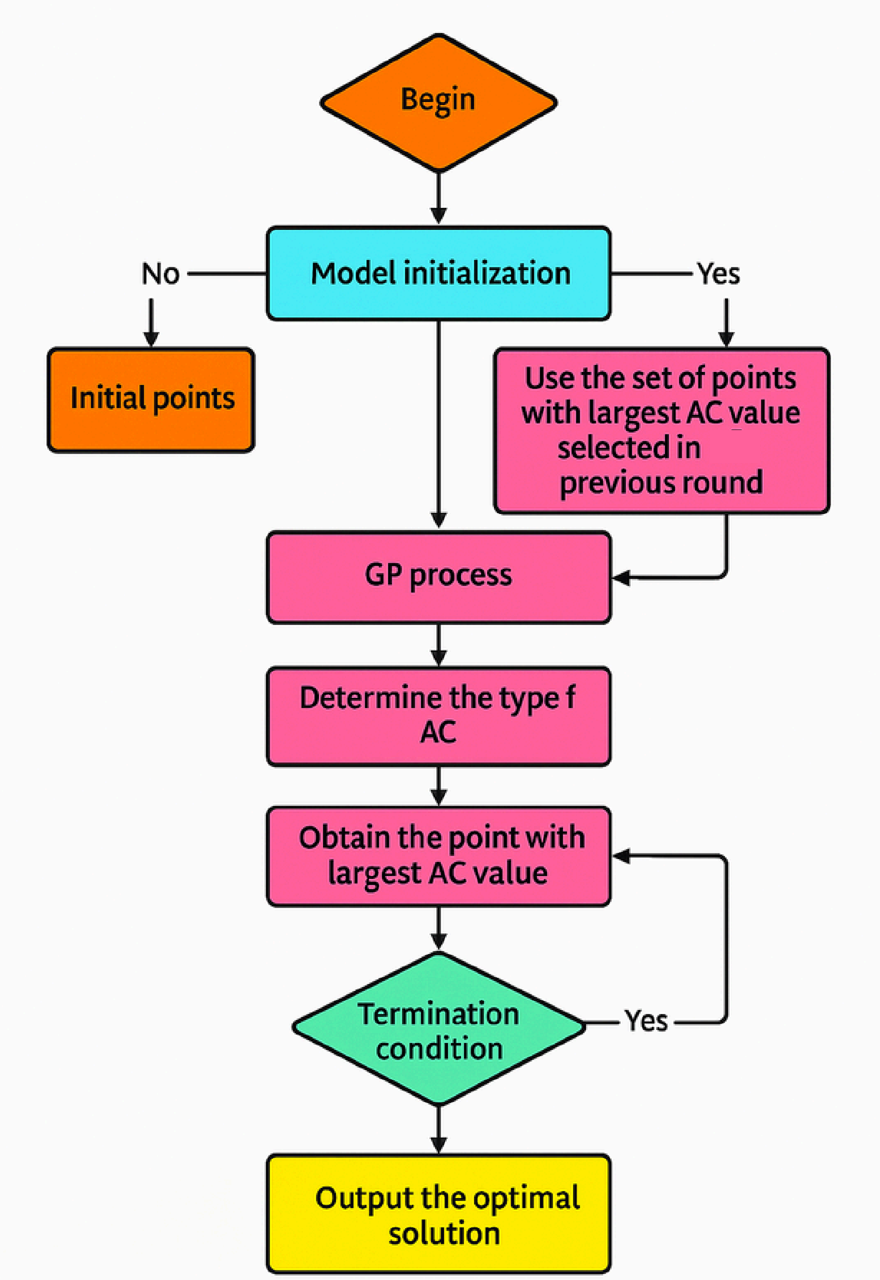
FLOWCHARTS



PSO ALGORITHM



GENETIC ALGORITHM

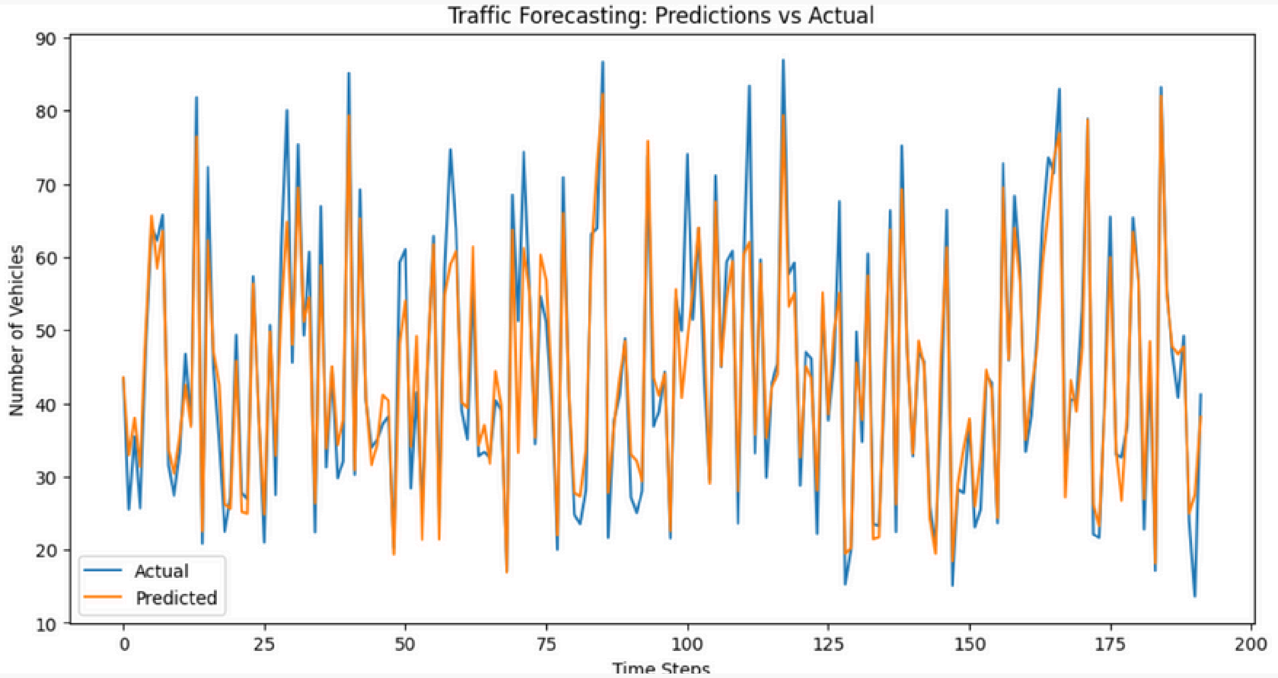


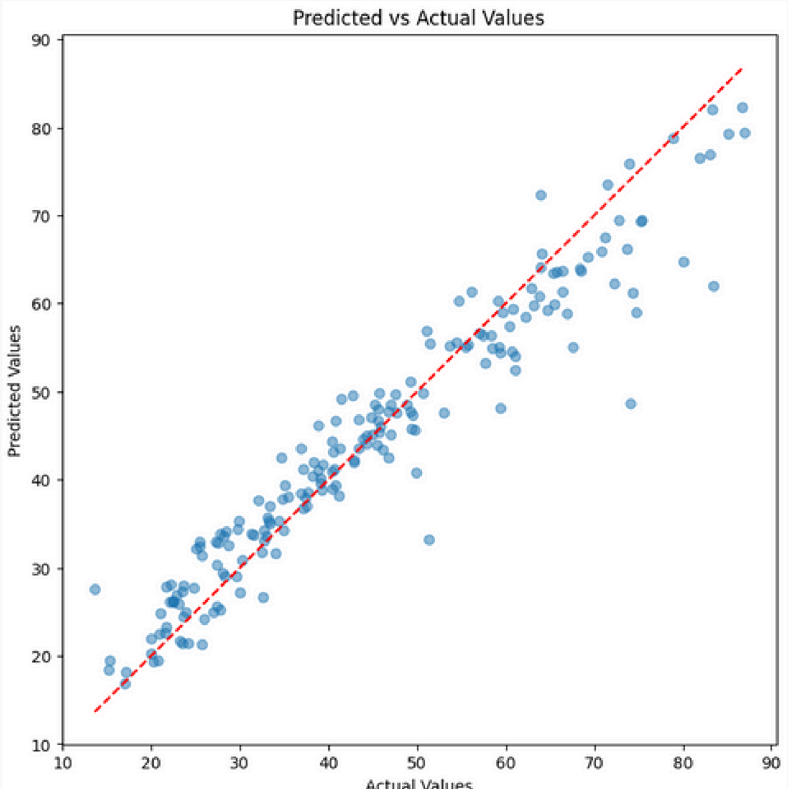
BAYESIAN OPTIMIZATION

RESULTS

LSTM

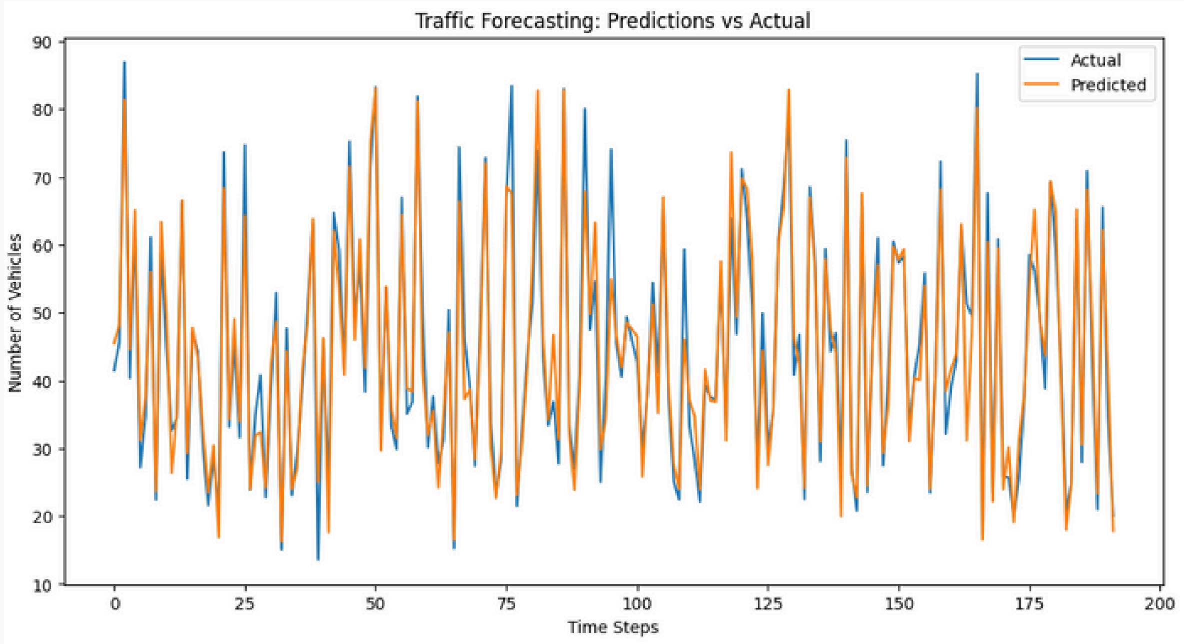
Test MSE: 27.3320
Test RMSE: 5.2280





PSO-LSTM

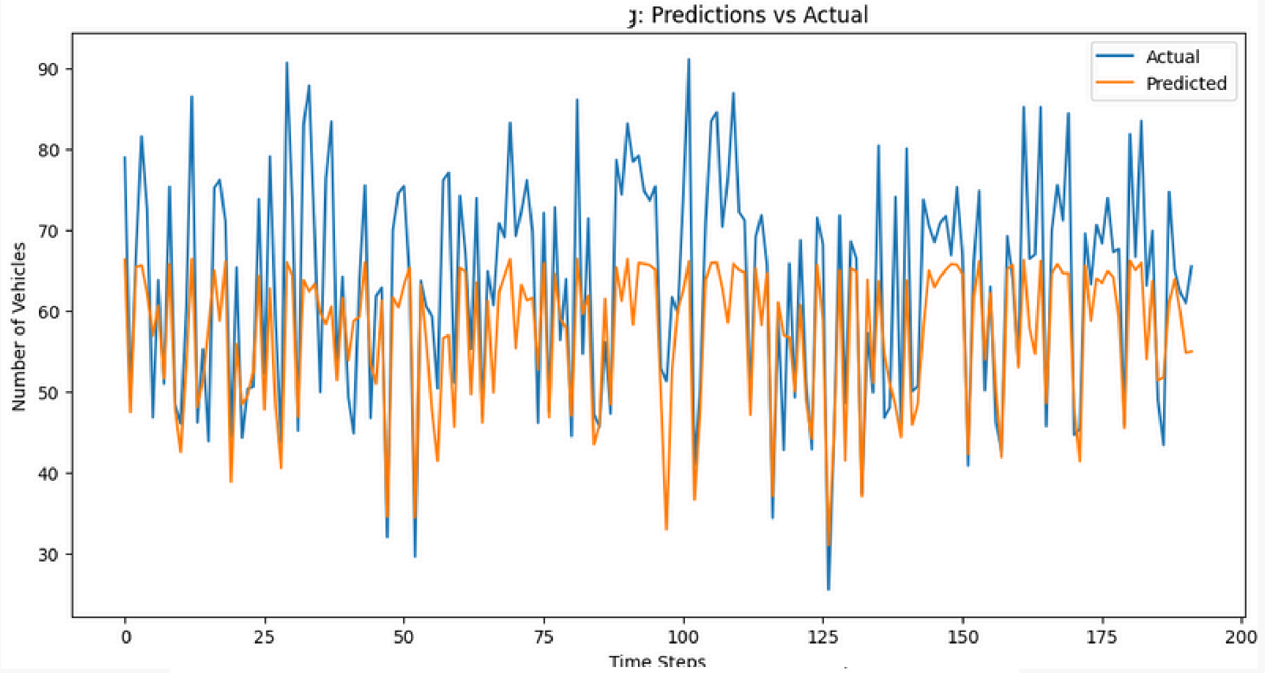
Test MSE: 20.7571
Test RMSE: 4.5560

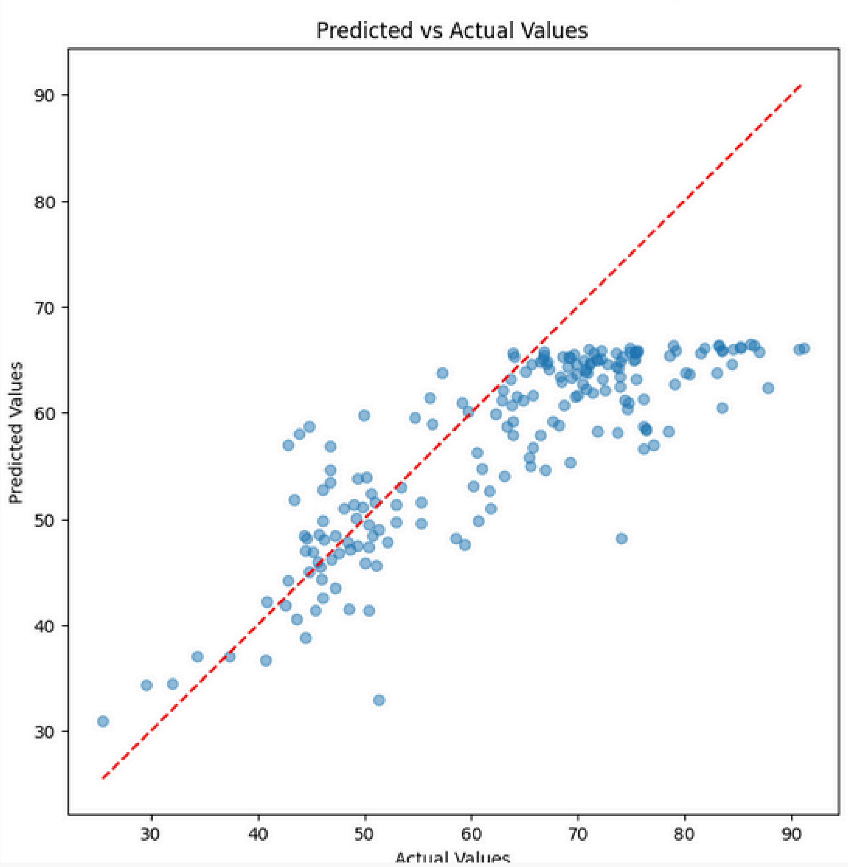


RESULTS

SEQUENTIALLY SPLIT LSTM

Test MSE: 100.1981
Test RMSE: 10.0099





HPO METHODS

Model Performance Comparison:
Base LSTM RMSE: 3.9587
GA-optimized LSTM RMSE: 3.9707
BO-optimized LSTM RMSE: 4.2835

K-FOLD CV

Validation MSE for fold 5: 216.1681
Average MSE across all folds: 206.3502
Standard deviation of MSE: 97.9568

OBSERVATIONS

1) PSO-LSTM performed better with lower MSE than the base LSTM model and hence we find that PSO is successfully able to optimize the model parameters.
2)Sequential split on the data gave a higher MSE compared to random split data and hence it is not suitable for out data.
3) GA and BO gave nearly similar errors but did not outperform base LSTM model.
4) K -Fold Cross Validation did not give satisfactory results for the 5 folds we tried them for.

Hence, PSO gave the optimized results and hence it is suitable for our data.

FUTURE SCOPE

Enhancing Model Utility & Deployment

- **Multimodal Data Fusion**
  - Integrate weather, GPS, and event data for richer context.
- **Real-Time Forecasting**
  - Deploy streaming models for live traffic updates.
- **Spatio-Temporal Modeling**
  - Leverage GNNs or 3D CNNs to capture spatial dependencies.
- **Cross-City Generalization**
  - Evaluate robustness on diverse geographic datasets.
- **Explainable Forecasting**
  - Use interpretability tools to uncover model reasoning.

Thank You...