



INTRODUCTION

Bengaluru, known as the Silicon Valley of India, is notorious for its traffic congestion, posing significant challenges for commuters and city planners alike. The city's rapid urbanization and exponential growth in vehicle numbers have strained the existing traffic management systems, necessitating innovative solutions to enhance traffic flow and reduce congestion.

PROBLEM STATEMENT

Problem 1: The goal is to count the vehicles of different classes and predict the counts for the future. Seven vehicle classes are considered, namely, 'Cars', 'Bus', 'Truck'.

Problem 2: Based on the data available at roads and junctions goal is to predict the traffic flow prediction.

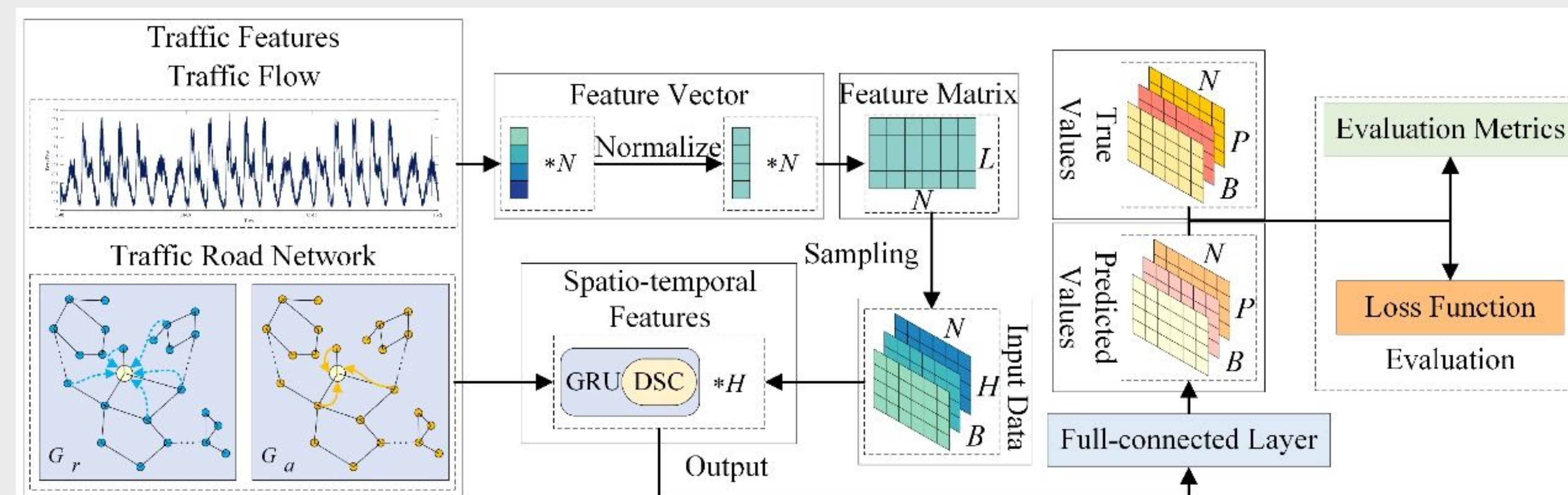
SOCIETAL RELEVANCE

- **Traffic Management:** Helps monitor and manage vehicle flow.
- **Urban Planning:** Provides data for designing roads, intersections, and traffic control systems.
- **Law Enforcement:** Assists in tracking vehicles involved in criminal activities across different locations.

TEAM MEMBERS

Suhas Gowda L 1RV22CS207
T Vinay 1RV22CS215
Shreya Chakote 1RV22CS189
Hanisha 1RV22CS244

Methodology

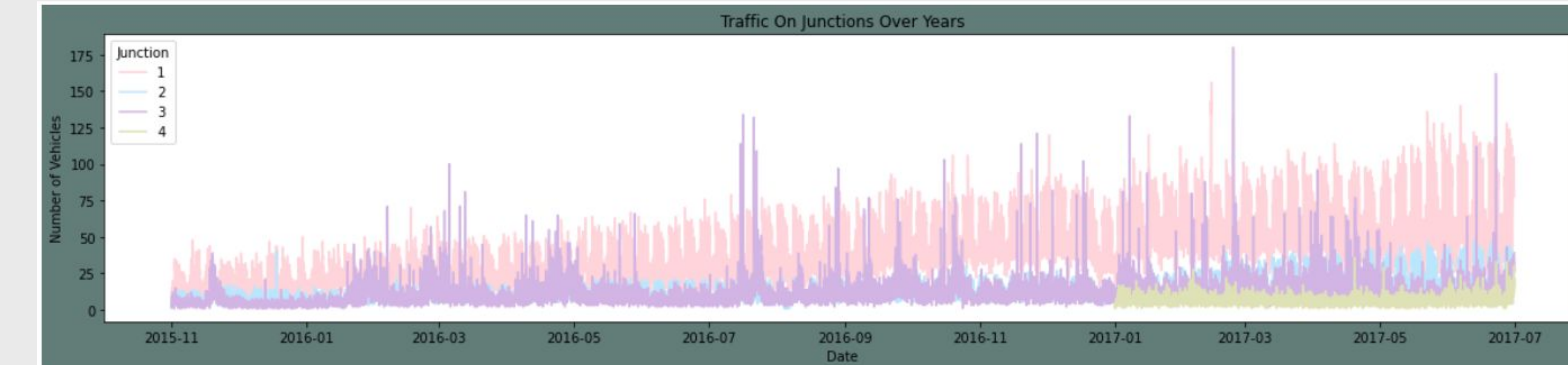


Data Analysis

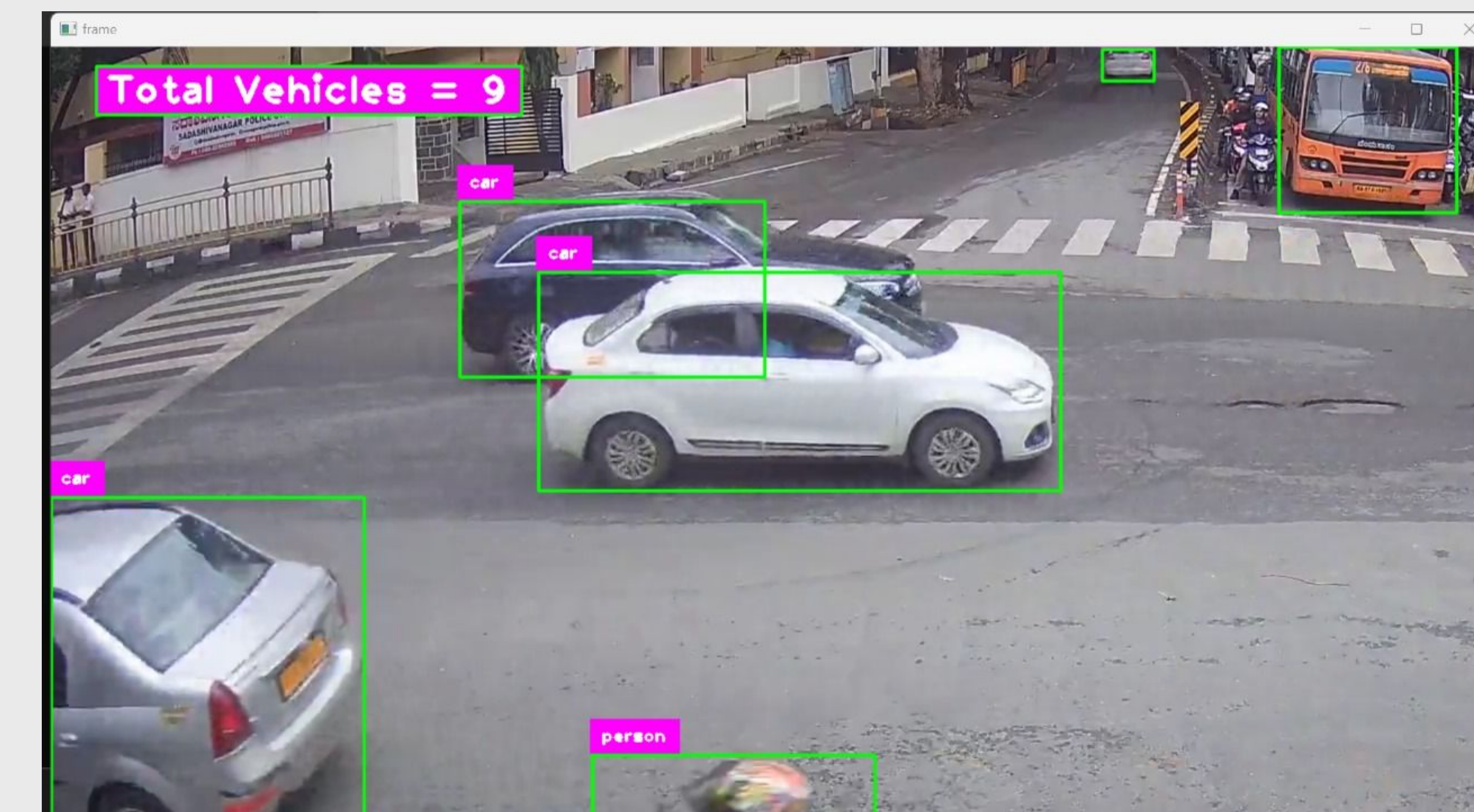
Exploratory Data Analysis involves visualizing and analyzing the data to understand its characteristics and identify patterns:

- **Trend Analysis:** Plot vehicle counts over time to observe long-term trends and seasonal patterns.
- **Seasonality:** Use heat maps or seasonal plots to identify recurring patterns at different times (e.g., daily or weekly cycles).
- **Correlation Analysis:** Examine the relationships between different features and the target variable (traffic count) to understand their impact.

Data Exploration



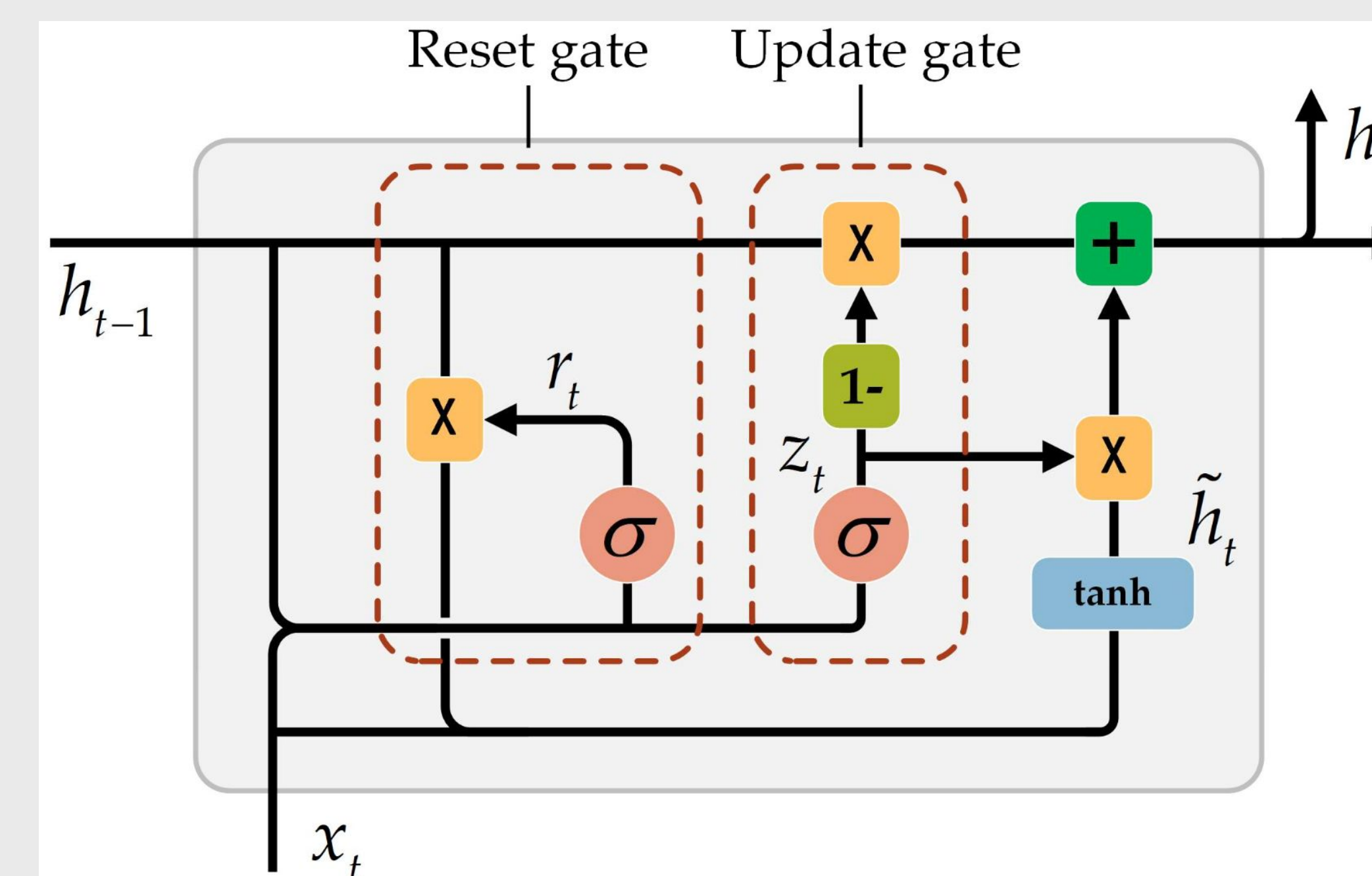
Output



Model Architecture

- **Model Architecture:** Define the structure of the GRU model, including:
 - **Input Layer:** Shape matching the time series data.
 - **GRU Layers:** One or more GRU layers with specified units. GRUs have gating mechanisms that control the flow of information, making them effective for capturing temporal dependencies.
 - **Dense Layers:** Fully connected layers to map the output of the GRU layers to the final prediction.
- **Hyperparameters:** Set the number of GRU units, learning rate, batch size, dropout rate, and number of epochs based on experimentation or domain knowledge.
- **Loss Function and Optimizer:** Choose an appropriate loss function (e.g., Mean Squared Error) and optimizer (e.g., Adam) for training the model.

GRU (Gated Recurrent Unit)



Model Building

