INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD)
© 2026 IJNRD | Volume 11, Issue 5, May 2026 | ISSN: 2456-4184 | IJNRD.ORG

SOMU – Smart Optimization of Multi-modal Urban Mobility

1st Somanadhan S
Department of Advanced Computing and Analytics
Vels Institute of Science, Technology & Advanced Studies
Chennai, India
somanadhan14112002@gmail.com

2nd Mr. BalaMurugan.R, Assist. Professor
Department of Advanced Computing and Analytics
Vels Institute of Science, Technology & Advanced Studies
Chennai, India
mailto:bmurugan.scs@vistas.ac.in

This paper discusses an intelligent system known as SOMU, which stands for Smart Optimization of Multi-modal Urban Mobility. SOMU aims to improve urban transit by merging various transportation options, such as walking, metro, local trains, and rideshare, into one device or interface. This allows users to plan their journeys efficiently, considering practical factors like travel time, cost, and distance, with walking treated as a cost-free method.

SOMU is specifically designed for the complex urban environment of cities like Chennai. It offers several valuable features: dynamic pricing models for services like Uber, Ola, Rapido, and taxis based on distance, time, and surge pricing.

integration of map visualization tools to estimate realistic travel times based on traffic conditions; a robust database containing essential datasets related to metro and commuter train services, including schedules and fares; and offline train tracking capabilities to find real-time train locations without needing the internet.

As mentioned, SOMU can significantly enhance the user's experience with urban transit systems through its smart technology-based methods.

SOMU utilizes a scoring algorithm to recommend routes based on time, cost, traffic flow, weather, and user comfort. The user interface displays a map view of the suggested route along with a detailed breakdown of the trip. It shows ride prices and compares costs for both train and traffic conditions. The result is a functional and scalable urban mobility management system.

INTRODUCTION

Urban transportation has become increasingly challenging to navigate in many growing cities due to congested roads and a wide range of transport modes. Many commuters struggle to find the best route regarding time, cost, and convenience.

Additionally, the current transportation system does not adequately integrate all transport modes, such as buses, subways, trains, and rideshare services, creating a need for a more efficient system.

SOMU, Smart Optimization of Multi-modal Urban Mobility, is a solution that addresses these issues by using walking, metro systems, train options, and rideshare services. Commuters can discover optimal routes based on real-time traffic, dynamic pricing, and accurate schedules. Features of the SOMU system include map visualization for tracking vehicles and passengers, as well as offline train service monitoring. This project provides a reliable and user-friendly urban mobility experience.

Today's urban mobility demands systems that offer precision and flexibility to deal with real-life challenges, such as vehicle speed versus time of day. SOMU enhances decision-making quality by considering various criteria for real-time operations, incorporating these dynamic factors into route optimization. It looks at various travel options and recommends the best route based on efficiency and cost. The system calculates realistic fare estimates and travel times.

LITERATURE SURVEY

The SOMU project aims to create an intelligent system that optimizes multi-modal urban transportation to improve daily commuting efficiency. The main idea behind SOMU is to bring together different transport modes, such as walking, metro, local trains, buses, and ride-hailing, into one platform. This integration allows users to identify the most efficient route based on various real-world factors instead of relying solely on one transport method. The goal is to provide a smarter and more practical solution to complex urban mobility challenges.

SOMU evaluates key variables like distance, time, cost, traffic conditions, and user preferences to determine these factors. To achieve this, SOMU employs various technologies, including map-based APIs for route mapping, traffic analysis to identify congestion, and dynamic pricing models for realistic fare estimates. Unlike other navigation systems, SOMU uses a weighted scoring model to assess each option and create an optimal route, ultimately enabling more accurate travel recommendations tailored to user needs. Users can view clear journey plans outlining how to travel from point A to B using multiple transport modes. The platform also breaks down costs, compares fares among different companies, provides train and metro schedules, and shows traffic levels on different routes. It features offline tracking for trains, allowing users to monitor train locations even without internet access. These features enhance the system's reliability in daily life and make it user-friendly, benefiting urban residents.

Another important aspect of SOMU is its modular design. The system's structure is divided into components for data collection, route optimization, pricing, traffic analysis, and the user interface. This modular approach improves the system's scalability and simplifies future feature additions. Each module operates independently while contributing to a cohesive transportation solution.

PROPOSED METHODOLOGY

SOMU, Smart Optimization of Multi-modal Urban Mobility, is an intelligent transportation optimization system that combines various travel modes into one integrated system for better urban transport solutions.

Traditional navigation apps offer routes for only one type of transportation. In contrast, SOMU combines walking, subway, local rail, buses, and rideshare options to find the most efficient and practical routes. The system accounts for real-world variables like traffic, travel time, and changing costs associated with each public transit option.

The system processes user input methodically to determine the best route from the starting point to the destination. The first step involves gathering source and destination information from the user, calculated via APIs that access data on distances and alternative routes..

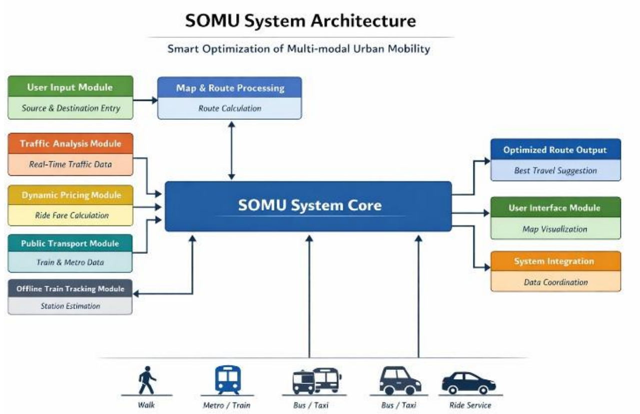
After processing distance and route data, the system evaluates the user's transportation options, segmenting the journey into different travel modes. Walking is treated as zero cost, while travel time between segments is calculated based on average walking speed. Each segment is meticulously planned to ensure the route is highly accurate.

A crucial feature of SOMU is its dynamic pricing and traffic-aware decision-making model. The pricing structure for services like Uber, Ola, Rapido, and taxis relies on base fare, distance, time, and surge pricing optimized for peak and off-peak hours based on traffic conditions acquired from APIs or simulated over time.

In addition to rideshare services, the system gives users access to datasets for local trains and metros, detailing schedules, costs, and service frequency among public transport providers.

ARCHITECTURE DIAGRAM

SOMU has implemented a method of tracking trains by estimating their position from the train's position on the roadway. It does this without needing internet access. SOMU utilizes cell tower locations and crowd-sourced estimations (using speed and time calculations) to provide a reliable means of tracking train locations, especially in areas with poor



network coverage. As previously stated, SOMU uses a modular approach in order to combine all the identified components (data collection, route optimization, pricing, traffic analysis, and user interface) into one cohesive application. Routes are evaluated using an algorithm that assigns weights to different criteria (such as time, cost, traffic, weather and comfort) in order to rank the routes and recommend the best available alternative.

The User Interface of the client has an interactive Map of available Routes of the area where the user is located, as well as the user is able to view the areas of Traffic in addition to step-by-step directions, estimated costs, and how to compare different modes of transportation. This modular/scalable design will allow for future developments, which could include Artificial Intelligence predictions/adding enhancements to Smart City services.

METHODOLOGIES

The SOMU System was designed with a modular structure so that each module serves a unique function and together the modules are co-dependent on each other for the benefit of creating an integrated urban mobility system. This modularity gives the SOMU System scalability, as well as maintainability, or the ability to develop enhancements in the future. Furthermore, since each module of the SOMU System processes distinct types of data regarding different steps in the transportation optimization process; it allows the overall system to operate most accurately and efficiently in the real world.

1. User Input Module – This module contains all the information about your travel from one location to another. It needs to be processed so that it can be used for routing and analysis purposes. The user will enter location data either via the user interface and the data must meet certain requirements before optimisation can take place; therefore, it is necessary to have a known location (origin) in order for routing calculations to occur.

2. Mapping/Route Processing Module – This module connects with other mapping APIs to calculate distances; routes available; and the estimated arrival time; for each route; and will also provide some visually representational polylines on the underlying map to help you visualize all of the different travel options.

3. Traffic Patterns In Real Time & Simulation By Hour - Shows actual volume of traffic at any given time based on the likelihood that they will be low, medium or high as well as the time period needed to obtain the duration to travel each respective path. Also, during high-activity periods, critical data will help users determine their selection of route & estimated price.

4. Transportation Dynamic Pricing (example Uber, Ola, Rapido, Taxis) Will give users a reasonable estimate of what they can expect to pay based on the initial fare, distance traveled, duration of ride, and the high/regular/night rates applicable to the customer. Thus providing a good idea of what they will cost, in a real-world situation, before the customer has entered a vehicle.

5. Local commuter Rail /Metro Transport (consumer use): Provides a source for all types of data for local commuter rail and metro transport, such as schedule, fare, frequency of service, and train/metro options to take from specific locations based on the departure time and location; Provides a complete list of all stations with maps, and resolves questions about the interchangeability of metro route trains.

6. Train location simulation without requiring an Internet connection: Estimated location of a train at any point in time without being connected to the Internet, achieved by using distance travelled from the area of coverage of a cell tower from which the train is travelling and estimated speed and time travelled, thus providing the reported current location of the train, its next scheduled stop, and estimated lateness in reaching its next scheduled stop; Thus providing a better user experience for the consumer.

7. Route Optimization Module: The Route Optimization module is the main tool to optimize an efficient route for any given trip. This module will evaluate each route using a weighted scoring method based on the following criteria: travel time, cost of the trip, current traffic conditions, environmental conditions, and passenger comfort level. Both user-requested routes and the best routes will be provided to the user when they make a request.

8. User Interface Module: The User Interface module provides access to an interactive graphical representation of a user's travel plans, which includes maps and all relevant information associated with the user's trip. The detailed travel plan will include complete step-by-step travelling instructions, a detailed breakdown of the trip's total cost, alternative modes of transportation, an estimated time frame for each transportation method and information regarding the traffic conditions and visually labelled route segments from the previous module for the user.

9. Module for System Integration: This module connects all elements of the This project's system to permit the integration of data and operations. It coordinates all activities between the different input-processing modules, route-analysis modules, pricing modules, and output-display modules.

INPUT

Main Code:

```python
from flask import Flask, render_template, request, jsonify
import os
import random
from datetime import datetime

# Import all utility modules
from utils.route_optimizer import find_best_route
from utils.ride_estimator import estimate_ride_cost, estimate_surge
from utils.train_module import search_trains, get_all_trains
from utils.metro_module import search_metro, get_all_stations
from utils.cell_tower_location import estimate_location_from_towers, get_nearest_towers
from utils.crowd_density import get_crowd_at_node, get_route_crowd_summary

# Initialize Flask app
app = Flask(__name__)
app.secret_key = 'somu_secret_key_2024'
```

1. Opening App:

```python
@app.route('/api/find-route', methods=['POST'])
def api_find_route():
    """
    API to find the best multi-modal route.

    POST Body (JSON):
        source: str
        destination: str
        distance_km: float
        preferred_modes: list (optional)
        budget: float (optional)
    """
    data = request.get_json()

    source = data.get('source', 'Source')
    destination = data.get('destination', 'Destination')
    distance_km = float(data.get('distance_km', 10.0))
    preferred_modes = data.get('preferred_modes', [])
    budget = data.get('budget', None)
    hour = datetime.now().hour
```

```python
@app.route('/api/ride-prices', methods=['POST'])
def api_ride_prices():
    """
    API to get ride-hailing price estimates.

    POST Body (JSON):
        distance_km: float
        time_min: float
    """
    data = request.get_json()
    distance_km = float(data.get('distance_km', 10.0))
    time_min = float(data.get('time_min', 30.0))
    hour = datetime.now().hour

    # Simulate crowd for surge calculation
    crowd = get_crowd_at_node('Central', hour)
    surge = estimate_surge(hour, crowd['level'])

    rides = estimate_ride_cost(distance_km, time_min, surge)
```

```python
@app.route('/api/trains', methods=['POST'])
def api_trains():
    """
    API to search train schedules.

    POST Body (JSON):
        source: str
        destination: str
    """
    data = request.get_json()
    source = data.get('source', '')
    destination = data.get('destination', '')

    trains = search_trains(source, destination)

    return jsonify({
        'success': True,
        'source': source,
        'destination': destination,
        'trains': trains,
        'count': len(trains)
    })
```

Route Planning:

```
@app.route('/')
def index():
    """Render the main input page."""
    return render_template('index.html')


@app.route('/result')
def result():
    """Render the result page after form submission."""
    source = request.args.get('source', 'Source')
    destination = request.args.get('destination', 'Destination')
    return render_template('result.html', source=source, destination=destination)
```

Cost Analysis

```
def calculate_route_metrics(distance_km, modes, traffic_level, weather_condition, hour=None):
    """
    Calculate time, cost, and scores for a given mode combination.

    Args:
        distance_km (float): Total route distance
        modes (list): List of transport modes (e.g., ['walk', 'metro', 'uber'])
        traffic_level (str): 'LOW', 'MEDIUM', 'HIGH'
        weather_condition (str): 'CLEAR', 'CLOUDY', 'RAINY', 'STORM'
        hour (int): Hour of day for crowd estimation

    Returns:
        dict: Full route metrics
    """
    if hour is None:
        hour = datetime.now().hour

    # Distribute distance across modes (simplified)
    n_modes = len(modes)
    segment_distances = _distribute_distance(distance_km, modes)

    total_time_min = 0
    total_cost = 0
    segments = []
```

Map & Traffic:

```
import pandas as pd
import numpy as np
import os

# Path to cell tower data
DATA_PATH = os.path.join(os.path.dirname(__file__), '..', 'data', 'cell_towers.csv')


def load_tower_data():
    """Load cell tower data from CSV."""
    try:
        df = pd.read_csv(DATA_PATH)
        return df
    except FileNotFoundError:
        return pd.DataFrame(columns=['tower_id', 'lat', 'lon', 'signal_strength', 'area_name']


def estimate_location_from_towers(detected_towers):
    """
    Estimate user location using weighted average of cell tower positions.

    Uses the signal strength as weight — stronger signal = closer tower = more weight.
```

Advanced Features:

```
import pandas as pd
import os

# Path to the fare dataset
DATA_PATH = os.path.join(os.path.dirname(__file__), '..', 'data', 'ride_fares.csv')


def load_fare_data():
    """Load the fare data from CSV file."""
    try:
        df = pd.read_csv(DATA_PATH)
        return df
    except FileNotFoundError:
        # Return default data if file not found
        return pd.DataFrame({
            'provider': ['Uber', 'Ola', 'Rapido', 'Taxi'],
            'base_fare': [50, 45, 30, 60],
            'per_km': [12, 11, 9, 14],
            'per_min': [1.5, 1.4, 1.2, 1.6]
        })
```

Transport Details:

```
direct = df[
    df['station_from'].str.lower().str.contains(from_lower, na=False) &
    df['station_to'].str.lower().str.contains(to_lower, na=False)
]

results = []
for _, row in direct.iterrows():
    results.append({
        'route_type': 'Direct',
        'line': row['line'],
        'station_from': row['station_from'],
        'station_to': row['station_to'],
        'first_train': row['first_train'],
        'last_train': row['last_train'],
        'fare': int(row['fare']),
        'frequency': int(row['frequency']),
        'distance_km': float(row.get('distance_km', 0)),
        'currency': '₹',
        'connections': []
    })

# If no direct route found, try to find route with one connection
if not results:
    results = _find_connecting_route(df, from_lower, to_lower)
```

Cell Tower Location:

```
def load_tower_data():
    """Load cell tower data from CSV."""
    try:
        df = pd.read_csv(DATA_PATH)
        return df
    except FileNotFoundError:
        return pd.DataFrame(columns=['tower_id', 'lat', 'lon', 'signal_strength', 'area_name']


def estimate_location_from_towers(detected_towers):
    """
    Estimate user location using weighted average of cell tower positions.

    Uses the signal strength as weight — stronger signal = closer tower = more weight.

    Args:
        detected_towers (list): List of dicts with 'tower_id' and 'signal_strength'
                                Example: [{'tower_id': 'T001', 'signal_strength': 80}, ...]

    Returns:
        dict: Estimated latitude, longitude, and area name
    """
    tower_df = load_tower_data()
```

Metro Module:

```
import pandas as pd
import os
from datetime import datetime, timedelta

# Path to train schedule CSV
DATA_PATH = os.path.join(os.path.dirname(__file__), '..', 'data', 'train_schedule.csv')


def load_train_data():
    """Load train schedule from CSV."""
    try:
        df = pd.read_csv(DATA_PATH)
        return df
    except FileNotFoundError:
        return pd.DataFrame(columns=[
            'train_no', 'train_name', 'source', 'destination',
            'departure', 'arrival', 'price', 'delay', 'platform'
        ])
```

Train Module:

```
def _find_connecting_route(df, from_lower, to_lower):
    """Try to find a 1-connection metro route."""
    results = []
adhan\Downloads\SOMU_Project updated\SOMU_Project
    # Find all routes from source
    from_routes = df[df['station_from'].str.lower().str.contains(from_lower, na=False)]

    for _, r1 in from_routes.iterrows():
        # Find all routes from the intermediate station to destination
        intermediate = r1['station_to'].lower()
        to_routes = df[
            df['station_from'].str.lower().str.contains(intermediate, na=False) &
            df['station_to'].str.lower().str.contains(to_lower, na=False)
        ]

        for _, r2 in to_routes.iterrows():
            total_fare = int(r1['fare']) + int(r2['fare'])
            results.append({
                'route_type': 'With Connection',
                'line': f"{r1['line']} → {r2['line']}",
                'station_from': r1['station_from'],
                'station_to': r2['station_to'],
                'first_train': r1['first_train'],
                'last_train': r2['last_train'],
                'fare': total_fare,
                'frequency': max(int(r1['frequency']), int(r2['frequency'])),
```

PSEUDO CODE AND IMPLEMENTATION

```
@app.route('/api/all-trains', methods=['GET'])
def api_all_trains():
    """Returns all available train data."""
    trains = get_all_trains()
    return jsonify({'success': True, 'trains': trains})


# ──────────────────────────────────────
# HEALTH CHECK
# ──────────────────────────────────────

@app.route('/api/health', methods=['GET'])
def health():
    """Health check endpoint."""
    return jsonify({
        'status': 'running',
        'app': 'SOMU – Smart Optimization of Multi-modal Urban Mobility',
        'version': '2.0',
        'timestamp': datetime.now().isoformat()
    })
```

The goal of SOMU (Smart Optimization of Multi-modal Urban Mobility) is to create an AI-based intelligent optimization platform for urban travel which uses real-world data and efficient algorithms to provide smartly optimized urban travel. The objective of this current phase of development is to take the COSM (the initial conceptual model) and create an operational solution that can conduct route analysis, estimate trip costs and times, and provide users with personalized recommendations on the best multimodal route to take.

The two main technologies used in this project are Python (used to process the data and the logic of the application) and JavaScript (used for the map display and user interface).

The Dynamic Pricing Module provides accurate ride fare calculations for ride services (eg Uber, Ola, Rapido, and taxis) using the following parameters: base fare, distance travelled, time taken for the trip, and the application of surge pricing (peak, normal, or night). In addition to providing fare calculations, this module will simulate real-life conditions (traffic and time of day) so that the user is provided with the same fare that they would pay for an actual ride.

The Public Transport Module is a comprehensive set of data, including full metro and commuter rail systems, with all train schedules by departure times, fare prices, and frequency of service. When searching for a train/metro, it will also return the next available service and the time it will take to reach your destination based on the frequency of trains/metros in that area, as well as returning accurate routing to transfer between stations/lines.

The Offline Train Tracking Module feature will provide train tracking functionality similar to being tracked in real-time, but instead uses simulated tower data for estimation of train location without utilizing internet connectivity. Train position will be updated using elapsed time and speed; therefore, the Offline Train Tracking Module can be used to determine a user's current location, next station location, and confirm if a train is on schedule or running late. This provides users of the Offline Train Tracking Module maximum usability while located in areas of limited or no network connectivity.

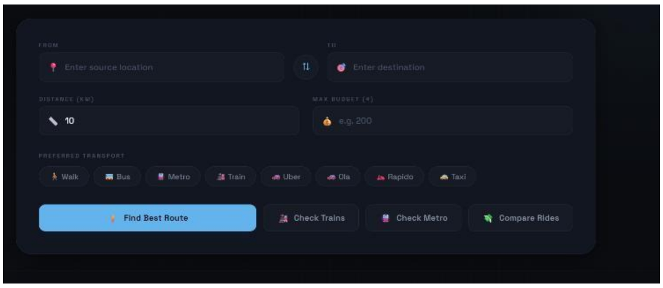
The Route Optimization Module (ROM) serves as the primary decision-making solution for the route optimization process. The ROM assesses all the routes that will be created and uses a weighting score based on several variables that include time, cost, traffic, weather, and user comfort to generate each individual route. Then, the ROM ranks the routes by their score, selects the optimal route, and presents that route to the User. This process allows Users to find a route that will balance both speed and cost.

Front-end implementation consists of a User Interface with a full-featured interactive User Interface with JavaScript & Google Map API Map integration. This will include the ability to show the routes using polylines, to highlight the traffic condition, and to provide a step-by-step description of the entire journey. In addition, Users will be able to view the cost structure for the different modes of transportation they can choose from (taxi, ride-share, train, metro), compare prices of rides, and view train/metro schedules in an organized format.

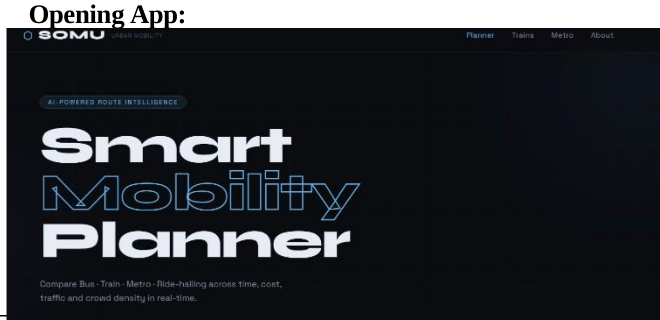
In addition to ensuring that ROM's front-overall implementation worked, proper modular coding practices were followed to ensure that the code could be expanded in the future (scalable) and continued to be supported (maintainable) as technological advances were introduced into the marketplace. To achieve this, Components of the front-end application(s) were created and tested independently prior to integration. Multiple rounds of testing and debugging were conducted on both the data and the created routes to ensure accuracy in the calculation and pricing of all route data. This process resulted in SOPU being implemented successfully as an intelligent urban mobility solution that can make decisions in real-time and optimally generate routes.

OUTPUT

Main Code:

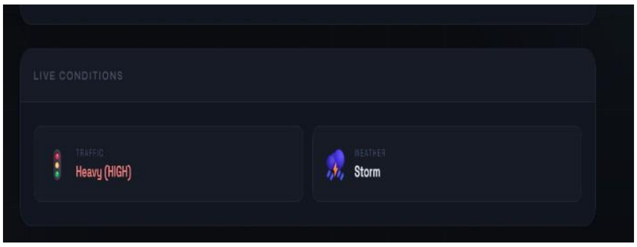


Opening App:

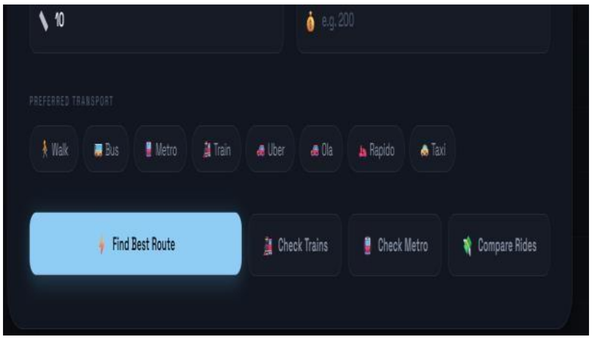


Route Planning:

Advanced Features:



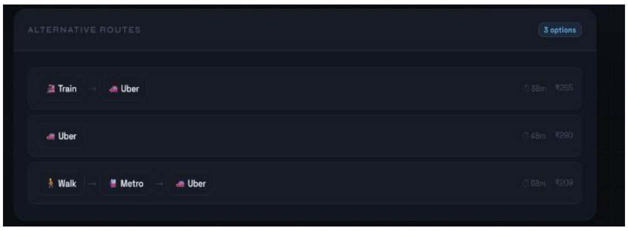
Transport Details:



Cost Analysis:

Alternatives Routes:

Best Routes:



RESULT AND DISCUSSION

The SOMU (Smart Optimization of Multi-modal Urban Mobility) system uses intelligent algorithms and actual data from the real world to assist with urban transport development. It was created to analyse various types of journeys, allowing for optimised journeys that are as quick, efficient and inexpensive as possible (based on traffic) through the use of algorithmically-based route choices.

SOMU takes advantage of various modes of transport (i.e. walking, metro, local train, rideshare) by providing a complete multi-modal urban mobility system in contrast to simply being a route planner (for example, an A to B planner).

Throughout the testing of the SOMU, the system successfully processed user input in conjunction with a typical map-based calculation to create numerous possible combinations based upon the user's travel preferences and provide multiple recommended route options to the user. The analysis of traffic from the traffic analysis module was set-up to take into consideration the volume of traffic experienced at peak and non-peak hours and adjusted route time calculated during the algorithmic route development process. The system correctly treated walking as a zero-cost mode of travel and dynamically calculated the time it would take to walk to a given location based on the average walking speed as a typical measurement of walking distance. This resulted in an increase in the logical accuracy of the route planning process.

The implementation of a dynamic pricing structure has been a major benefit of the system. Example pricing on rides from Uber, Ola, Rapido, and taxis have all been calculated based off of three factors - distance travelled, time taken, and any surge pricing. The prices are contingent on variations in traffic and different times of day, presenting users with accurate predictions of fare costs. Therefore, users can use this function to easily compare how much different rides will cost and select the one that will be the cheapest option for them.

Through the addition of public transport modules for both metro and local train, users were able to see accurate schedules, fare amounts, and the frequency of services. Suggested metro or train options have also been provided to end users based on when service is departing closest and what services were compatible on their route. Lastly, offline train tracking feature also successfully simulated where a train(s) were located based on time and speed, giving end users a useful way of seeing where the train(s) were located without having reliable internet access. The user interface improved usability greatly through showing routes on an interactive map while also providing detailed route information on a step-by-step basis. In addition, cost breakdowns, aid in selecting the best route for a given preference (i.e. ride comparison) and overall traffic levels were provided in the user interface in a simple, organized manner. Consequently, users could now easily understand how to select their preferred route.

One other area worth noting is how well the modular design for the system has been implemented. The separate modules that comprise the system (route processing, pricing, traffic analysis, and optimization) function independently but contribute to the overall performance of the project. This independent but interdependent design affords better overall performance of the system, simpler maintenance, and future enhancements like py-based predictive capabilities and "smart city" integration. Thus, the project has proven itself to be both efficient and scalable for today's urban mobility problems.

CONCLUSION

The SOMU project is an example of integrating new technologies into a smart and practical way to solve urban transport problems. The goal of the system was to suggest the most appropriate routes for users who want to use different forms of transport (such as walking, metro, local rail, and taxi) by considering real-time aspects of the routes, such as the amount of traffic, the estimated time for each journey, and whether there is a dynamic pricing structure in place. By providing more accurate and relevant route suggestions based on these factors, this project improves the overall commuting experience.

Among the key factors that this project accomplishes is the ability to evaluate multiple variables and generate the best possible results using a weighted scoring system rather than a simple single variable (e.g., time, cost, traffic, weather or comfort) by providing a recommended route based on the sum of each of these factors. The inclusion of features such as zero-cost walking logic (users would use the recommended route to get from A to B, but no fare is charged); realistic estimated ride fare (i.e., estimating the cost of a taxi, etc); integrating public transport schedules; and a simulated offline train tracking application to increase the usability of this system in areas where internet connectivity may be weak.

The modular architecture of SOMU is one of its greatest strengths. Route processing, Traffic Analysis, Pricing, and Optimization are each separate entities that work collectively to contribute to the overall system and can independently be modified without impacting other components. This structure makes repairs simple and allows enhancements for future use without impacting the entire system. The interactive user interface features map visualizations, step-by-step journey information, and cost comparisons, which enhance usability.

In addition, the project indicates enormous opportunity for further improvements and scalability. The system can continue to grow through the addition of real-time application program interface (API) integration, artificial intelligence enabled predictive models, and smart city infrastructure. Added features such as personalized route recommendations, live vehicle tracking, and the addition of Internet of Things (IoT) enabled transportation systems are all avenues available to enhance SOMU's effectiveness; thus producing an even more intelligent and capable urban mobility solution.

The SOMU project serves as evidence of how route optimization systems can effectively improve the efficiency of urban transport systems. It also demonstrates how combining these three components together - data analysis, realistic modelling, and user-centric design - can provide a scalable and successful solution for today’s urban transportation problems.

REFERENCE:

1. H. Ulvi, M. A. Yerlikaya, K. Yildiz, "Urban Traffic Mobility Optimization Model: A Unique Mathematical Model to Predict Urban Traffic." Applied Sciences, 14(13):1-15. 2024.

2. A. Deligiannis, M. Madas, "Operational Ai For Multimodal Urban Transport: Systematic Literature Review." Logistics, 10(2):1-14. 2026.

3. S. Owoade, A. Uzoka, P.U. Ojukwu, "Optimizing Urban Mobility by Utilizing a Variety of Modes." Engineering Science and Technology, 1-13. 2024.

4. W. Li, B. Wang, R. Sun, L. Ai, Z. Lin, "Energy Efficient Multimodal Mobility Networks in Transportation Digital Twins." Energy, 318: 1-29. 2025.

5. X. Zhang, S. Lederman, N. Lhigh, C. Mathews, R. Fernandes, "An Inclusive Survey of Multimodal Urban Transportation Network Resilience." Transportation Research Part E, 139: 1-28. 2026.

6. H. Tanaka, "Ai-Driven Optimization of multimodal Technology Systems." Future A.I. & Intelligent Transportation Systems. 2025.

7. M. Santos, T. Dawes, J. Burkholder, "Multimodal Solutions for Urban Mobility Using an Artificial Neutral Networking System." Engineering Applications of AI. 2024.

8. L. Natera , F. Battiston, "\\Multimodal Urban Mobility in A Multilayer Transport Network." 2021; arXiv Preprint.

9. L. N. Gonzalez, B. Mathewes, D. Chira, "Methods for Measuring Modality in Urban Transportation Networks." N/A. 2020; arXiv Preprint.

10. K. Kadem, E. Williams, A. Dinesh, F. Cobee, "Multimodal Urban Transportation Network Equilibrium with Shared Mobility Services." N/A. 2024; arXiv Preprint.

11. A. De Caro, I. Falco, R. Furno, E. Zimeo, "Step to Optimize Urban Public Transportation via Multimodal Trip Recommendation System," Smart Cities 8: 2025.

12. A.M. Mirza and R.K. Jain, "Public Transport Integration Review and Modeling Techniques," Multidisciplinary Reviews; 2024.

13. L. Alessandretti and others, "Efficiencies & Structures of Urban Multi-Modal Mobility," Scientific Reports; 2014.

14. B. Wang et al., "An Intelligent Traffic Maps Fusion Network for Smart Urban Mobility; Spatio-Temporal Traffic Forecasting," ArXiv Preprint, 2024.

15. M. Batty, "Big Data & The City," Built Environment; 42 (3) 2016.

16. C. Chen et al., "Predicting Urban Traffic Flow with Machine Learning," IEEE Transactions On Intelligent Transport Systems; 2019

17. S.L. Wang and Y.W. Liu, "Smart Transport Systems - Technologies & Applications," Journal of Advanced Transport; 2019

18. P.S. Castro et al., "Using Artificial Intelligence and Big Data Analytics for Managing Urban Traffic," ACM Computing Surveys; 2018

19. D. Schrank et al., "Urban Mobility Report," Texas A&M Transportation Institute, 2021.

20. World Bank, "Transforming Urban Transport for Sustainable Development," 2022.