PUBLIC TRANSPORTATION OPTIMIZATION USING IOT

Team Member

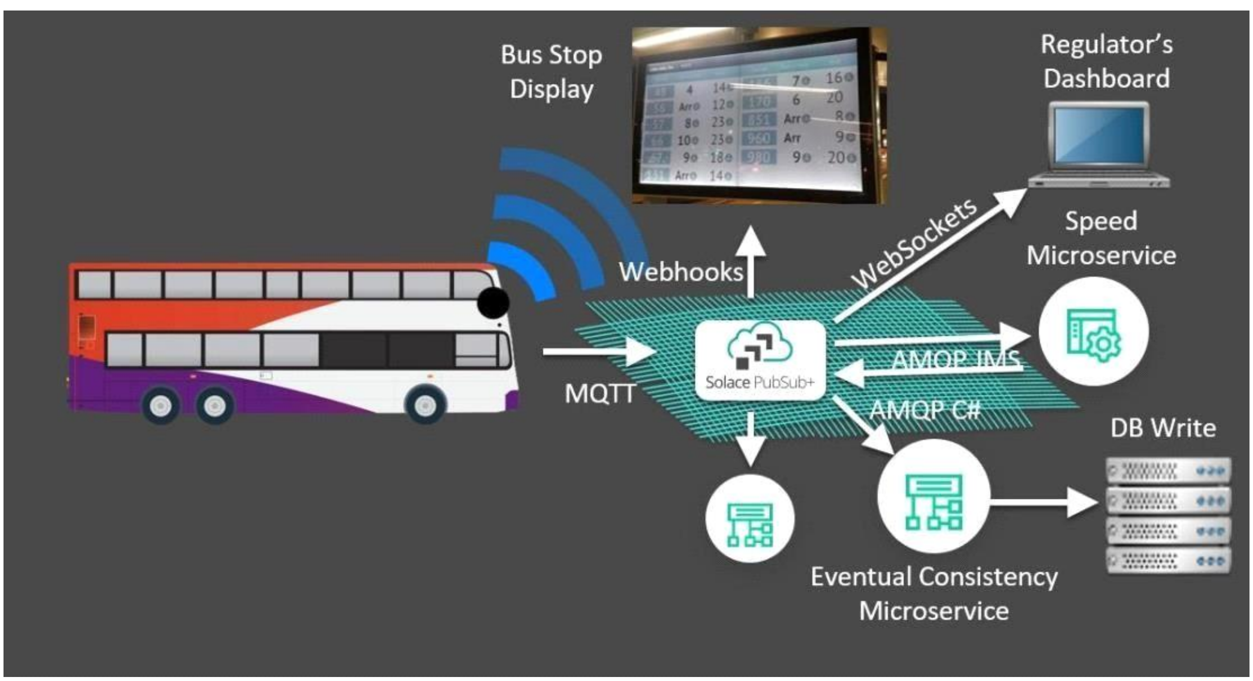
GOKUL SELVAM A P – (912221106005)

Phase 5 submission document

Project Title : PUBLIC TRANSPORTATION OPTIMIZATION

Phase 5: Project Document & Submission

Topic : In this section we will document the complete project and prepare it if submission.



PUBLIC TRANSPORTATION OPTIMIZATION

Introduction :

- The progress of information technology in IoT development is very influential on the various aspects of human activities. The paradigm of the IoT provides a reference to connect all physical objects in the global Internet base as well as the existing infrastructure for information and communication exchange. IoT aims to support rapid and precise identification such as location tracking, monitoring and management. Therefore, IoT is based on multiple integration of communications solutions, technology identification and tracking, sensor networks and actuators, and sharing of other information distribution .
- According to , IoT Architecture network consists of several layers such as layers of sensing, access layer, network layer, middleware layer and application layer. The application layer integrates the functions of the lower system, and builds practical applications from various industries, such as smart grid, smart logistics, intelligent transportation, precision farming, disaster monitoring and remote medical care.
- IoT's main function is to collect data to be measured by a sensor where it is integrated into a shortrange wireless network such as Bluetooth, ZigBee or Wi-Fi, and then transmit data to a larger network such as an internet network gateway . IOT sensors use low cost, highly scalable, efficient, lowpower, and integrated data across all sub-networks. The more sensors combined and with the increase of data collection time, the data will become larger and known as the "Big Data". Big Data was introduced by Gartner Report in 2001 and has three dimensions covering 3Vs: Volume, Velocity, and Variety. This definition has been rewritten and reasserted by others to include the fourth V: Veracity . In short, IoT provides a means of data collection, detection and monitoring of an event, an algorithm for acting on an activity, storage of data and a considerable analysis.

- ❖ From several papers, the authors observed and found that most researchers tend to use and utilize IoT on passenger safety, so they focus more on features that help control drivers in driving the bus, monitoring bus lines and utilizing radio signals and LAN networks and other applications to maximizing IoT functionality on transport. The goal is to minimize the occurrence of accidents.
- ❖ But what about passenger comfort in choosing public transportation? Inspired from the public transport navigation system , The purpose of this research is to know the opportunities that can be used to maximize IoT function on public transportation. if previous researchers have made a monitoring system for bus travel, bus scheduling as well as early detector of the accident then this research tries to explore opportunities that can be obtained and used to produce a better public transportation system. Based on this, the research question is what the IoT function can be usedfor smart public transportation?

Here’s a list of tools and software commonly used in the process:

1. Automated Fare Collection (AFC) Systems: Used for ticketing and fare collection in buses, trains, and other public transport systems.

2. Real-time Passenger Information (RTPI) Systems: Provide real-time updates on bus/train schedules, routes, and any service disruptions.

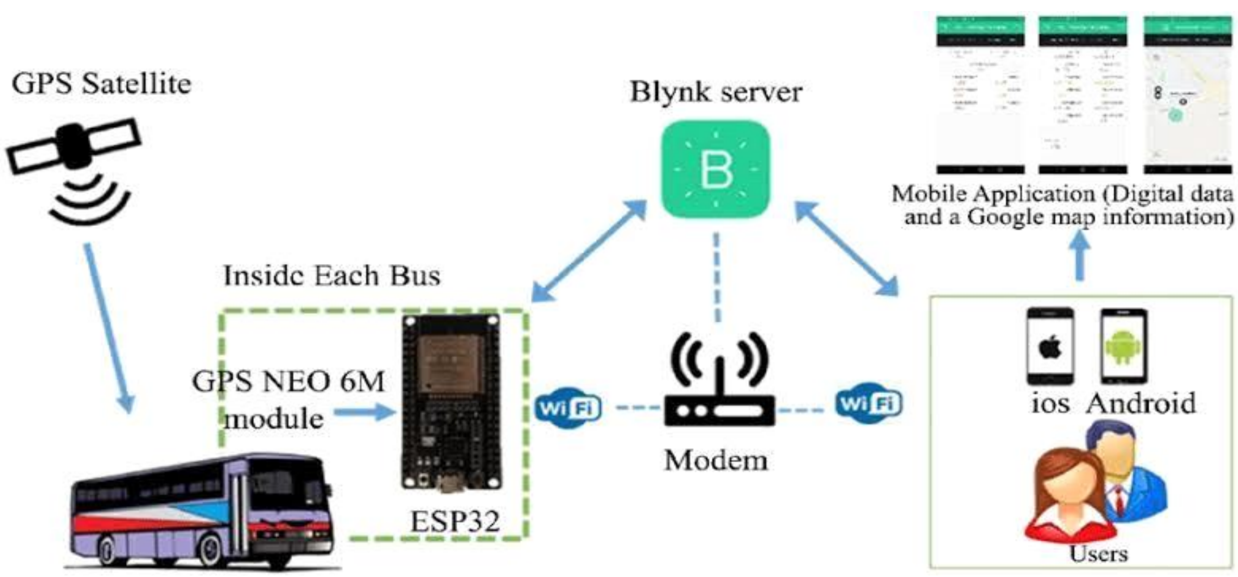
3. Transportation Management Systems (TMS): Used to manage and optimize operations, including scheduling, vehicle tracking, and maintenance.

4. Automatic Vehicle Location (AVL) Systems: Utilized for tracking and managing the location of vehicles in real-time.

5. Fleet Management Software: Helps monitor and manage the fleet of vehicles, including maintenance schedules and fuel consumption.

6. Traffic Management Software: Aids in controlling and optimizing traffic flow, particularly for bus and tram networks.

These are just a few examples, as there may be various other specific tolls and software used depending on the type and scale of the public transport system.



DESIGN THINKING AND PRESENT IN FORM OF DOCUMENT

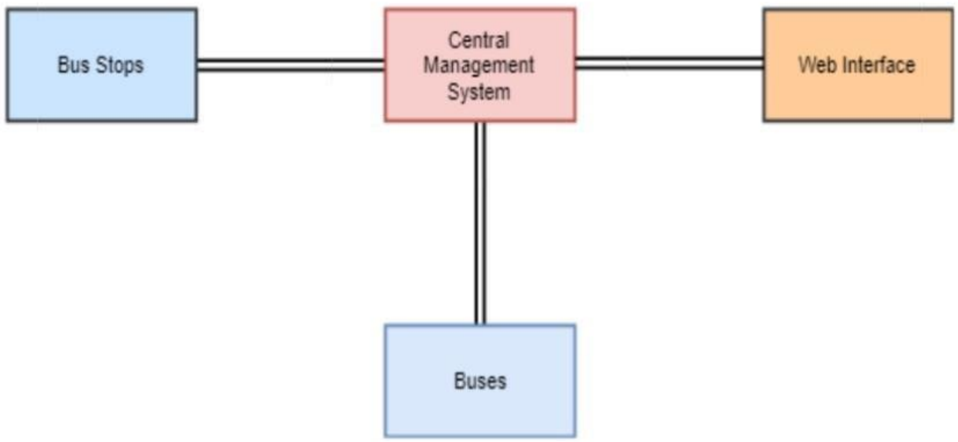
This paper's focus is on the system design and implementation for the Intelligent Urban Transportation system IUTS to tackle the mentioned challenges detailed in the previous sections. The proposed design needs to satisfy the following functional requirements and design principles as detailed in the below sections:

.1. Vehicle Tracking To provide the user with the most accurate information regarding the current location of the requested bus, a tracking module must be installed in all the vehicles. This module is crucial to the system as it collects real-time data of the bus's exact location.

.2. Fare Payment A cashless fare payment module should be used as the main ticketing system. As the public bus system is still reliant on paper tickets which are easy to defraud and are difficult to keep track of. Not only to get a ticket, but also as a user identification and driver authentication module.

3. Vehicle Health Monitoring The module collects real-time vehicle specific data like the speed and engine related status. These data give some insights which help in scheduling maintenance and monitor the health of the vehicle and the driving behavior.

.4. Server Communication All the data collected from the implemented modules must be broadcasted to a cloud network where it is processed and analyzed. The high-level design of the entire system is shown in Fig. 1. The system comprises of three integral parts, the hardware layer which is implemented onboard the buses and on the bus stops through which the data is collected, the Central Management System (CMS) [15] which includes the server, database and the application layer which interacts with both the users and operators in the form of a mobile application and a dashboard. The hardware consists of an OBU, which is installed onboard the buses. This acts as a data collector, collecting the data from the various modules. The temporal and spatial data which allow live tracking, the cashless ticketing module, and finally the health monitoring through the OBD to achieve the function requirements. A processing unit, processing and analyzing the collected data and a communication unit, communicating with the server sending the data packets and receiving the final route [20].



.5. Design approach The detailed design process went through two phases as follows: A. Design Phase I The onboard unit (OBU) detailed in Fig. 3, is an integral part of the system as it will generate the necessary data required for the software component of the overall proposed system. Given the lack of accurate, publicly available data about the bus transport system, aspired a need for an accurate GPS module. This module is activated continuously and transmits a series of latitude-longitude coordinates to the CMS using the GSM transceiver. The second function of the onboard sub-system is to monitor the health of the vehicle, i.e. check for fault codes and alert the operator before any major issues. For this function, the output of the vehicle's OBD socket must be read and interpreted. The proposed system is planned to pilot on the a University bus fleet, therefore the OBD socket must be adapted such that it uses the standard Controller Area Network (CAN) protocol as the K-Line protocol [19] is outdated and cannot be adapted to the Universal Asynchronous Receiver-Transmitter (UART) format. From there the Raspberry Pi (RPi) can interface with the OBD using UART and can translate the continuous output of the OBD, using a look-up table, to a set of alerts. The third function of the on-board unit is to enable cashless payment through a Radio-

Frequency Identification (RFID) reader module. A typical user interaction with the RFID reader, whereby a passenger boarding the bus must swipe their RFID card [16] to trigger a set of validation and verification procedures which ensure a passenger is using a valid card and has sufficient funds in their virtual wallet to pay for the trip. Finally, the system receives its route from the CMS and is capable of displaying that route the driver with instructions; this is possible through the Raspberry Pi's interface with the LCD screen .

DESIGN INTO INNOVATION

This paper uses the methodology systematic literature review [16] to review existing literature related to IoT for public transportation, security in public transportation, and time efficiency. This study conducts a thorough literature review of studies on IOT utilization on public transport. This is the process of determining the source of the research used, which determines the keyword pattern for the paper search process, initiates inclusion and exclusion criteria, data mining, and analysis of findings for answer research questions. A. Search Process The first process is to determine the literature source to find the appropriate articles / journals. Sources selected for systematic literature review are as follows:

1. IEEEXplore Digital Library (http: /ieeexplore.ieee.org)
2. Direct Science (www.sciencedirect.com)
3. Springer Link (link.springer.com)
4. Emerald Insight (www.emeraldinsight.com)
5. Google Scholars (https://scholar.google.co.id)
6. Wiley Online Library (onlinelibrary.wiley.com)
7. ACM Digital Library (dl.acm.org)

8. Elsevier (https://www.elsevier.com/)

The search mechanism inclusion criteria consists of three filter processes. The first is the search process. All documents we find from source publications related to the specified keywords will be saved as Founded Studies. After that, the next step we filter the paper according to title and abstract. If the title and abstract are free and suitable for determining research questions, then this paper will be stored as a —Candidate Study‖.

Then the final section to filter these writings is that all candidate documents will be read thoroughly to answer research questions. If the letters are appropriate to answer the research question, the paper will be defined as —Selected Studies‖. The applied keyword pattern for finding research papers relating to this research was made using the Boolean operator to filter the data, so it can be specified priority to search data based on the symbol used. Boolean symbols and operators used in this paper, such as OR, AND. Combination of keywords are as follows: (Internet of thing OR IoT) AND (public Transportation OR intelligence transportation system)

- AND (smart transportation OR public transportation) (internet of thing OR (IoT) OR (public AND transportation) OR (smart AND transport)) AND

- (IoT OR RFID) (public AND transport) OR (RFID AND key IoT)) AND (intelligent OR transportation)

- (internet of thing OR smart transportation) AND (smart cities OR transportation)

- (IoT AND transportation) OR (smart transportation OR Sensor RFID)

- The inclusion criteria of the search mechanism in this study consisted of three filter processes. The first is the process of "Founded Studies". All papers that we find from the publication sources related to the specified keywords will be stored in Founded Studies. The next step we filter the paper according to title and abstract. If the title and abstract have a correlation and are suitable for determining research questions, then this paper will be stored as a "Candidate

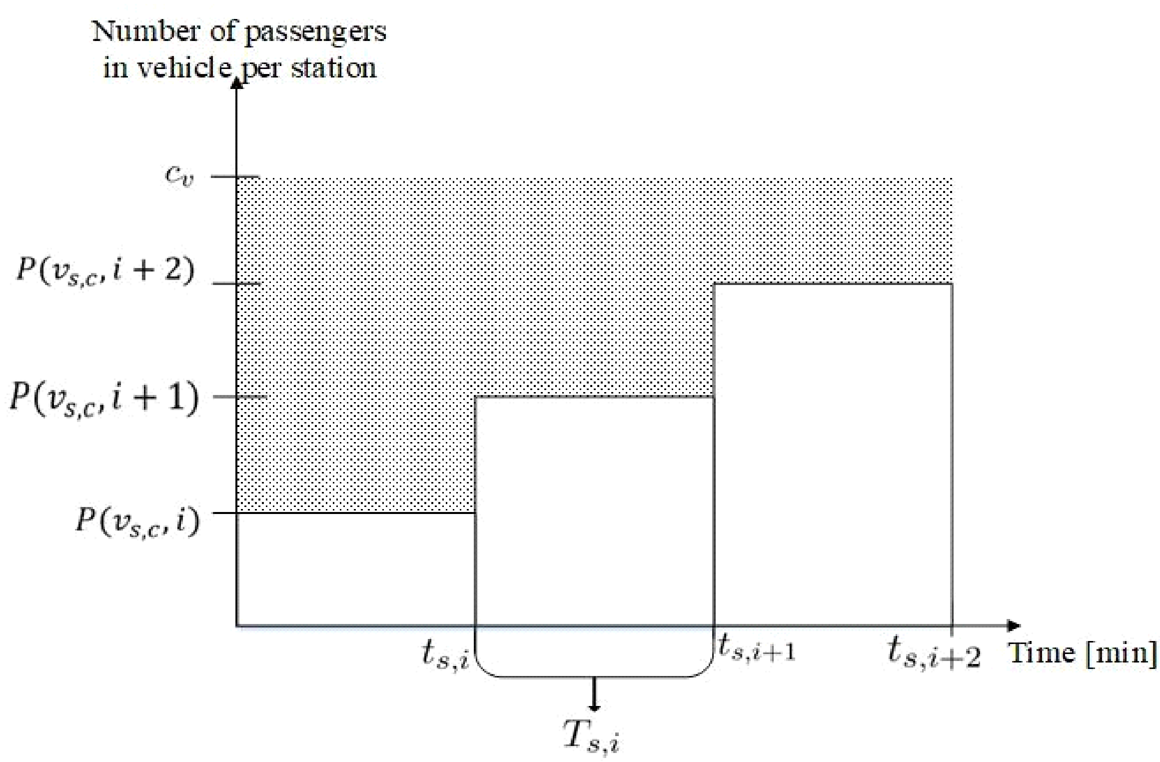
Study". Then the last part is filtering out these writings and will be read thoroughly to answer research questions. If the paper tends to be defined as "Selected Studies". While to clarify the validity of the literature, the search exclusion criteria are defined in several procedures, namely:

a. This paper is based on the date of their publication between 2010-2018

b. The complete paper structure, which means all identities (journal / conference, author's identity, etc.) are mentioned on paper.

c. Duplicate paper from the same study will not be included and removed from the SLRData Extractions

The research literature uses 105 papers from all sources and criteria. Of the 105 papers International Journal of Pure and Applied Mathematics Special Issue 3593 examined, there are 53 papers that study candidates based on titles and abstracts related to research questions. After further study, there are only 32 papers that can be used because it has a very strong correlation with this research.



PYTHON PROGRAM

Mini code:

```
import random
import time

# Simulated IoT data for vehicle locations
vehicle_data = {
    "bus_1": {"latitude": 40.7128, "longitude": -74.0060},
    "bus_2": {"latitude": 40.730610, "longitude": -73.935242},
    # Add more vehicles here
}

def simulate_vehicle_data():
    for vehicle_id, data in vehicle_data.items():
        # Simulate vehicle movement (randomly)
        data["latitude"] += random.uniform(-0.001, 0.001)
        data["longitude"] += random.uniform(-0.001, 0.001)
        print(f"{vehicle_id}: Latitude {data['latitude']}, Longitude {data['longitude']}")

while True:
    simulate_vehicle_data()
    # Implement route optimization and traffic monitoring logic here
```

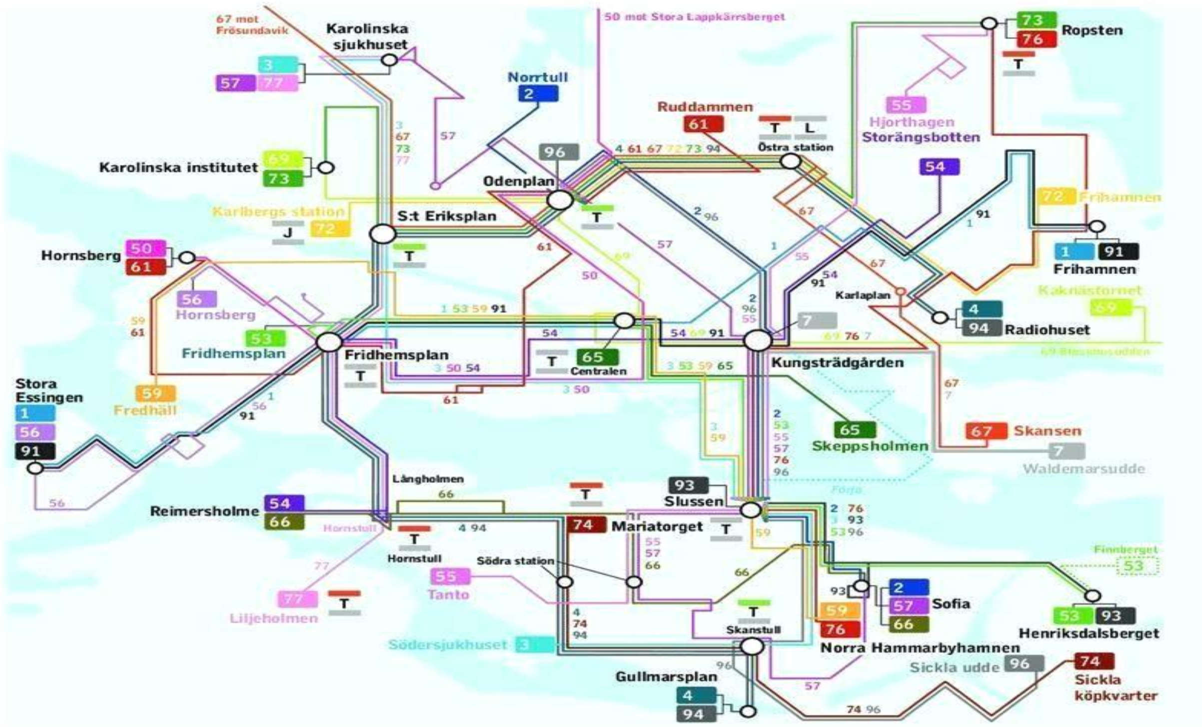
```
    time.sleep(10)  # Simulate data updates every 10 seconds
```

Python script for public transport systems

```
Import random

Import time

# Simulated IoT data for buses

Bus_data = [

    {“bus_id”: 1, “location”: (37.7749, -122.4194)},  # San Francisco

    {“bus_id”: 2, “location”: (34.0522, -118.2437)},  # Los Angeles

    # Add more buses and their locations

]

# Simulated passenger demand

Passenger_demand = [

    {“location”: (37.7749, -122.4194), “destination”: (34.0522, -118.2437)},  # SF to LA

    # Add more passenger demand routes

]

Def optimize_routes(buses, demand):

    # Add your optimization logic here

    # For demonstration, we’ll assign buses to the nearest passenger demand

    Optimized_routes = []

    For passenger in demand:

        Min_distance = float(“inf”)
```

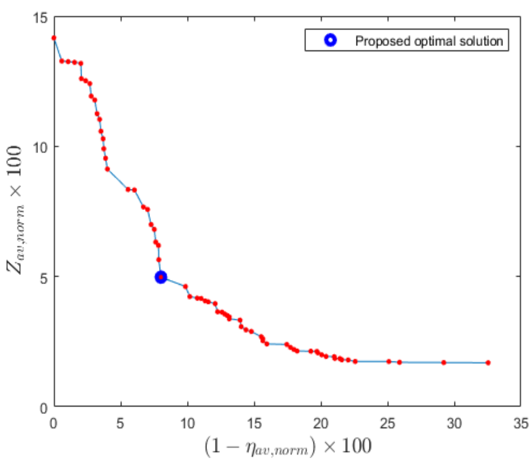
```
      Assigned_bus = None
      For bus in buses:
         Distance = haversine(bus[“location”], passenger[“location”])
         If distance < min_distance:
           Min_distance = distance
            Assigned_bus = bus[“bus_id”]
      Optimized_routes.append({“passenger”: passenger, “bus_id”: assigned_bus})
   Return optimized_routes
Def haversine(coord1, coord2):
   # Haversine formula to calculate distance between two coordinates
   Lat1, lon1 = coord1
   Lat2, lon2 = coord2
   # Calculate distance (for simplicity, this is not accurate for long distances)
   Return random.uniform(10, 200)  # Simulated distance
While True:
   Optimized_routes = optimize_routes(bus_data, passenger_demand)
   For route in optimized_routes:
      Print(f”Bus {route[‘bus_id’]} assigned to passenger route from
{route[‘passenger’][‘location’]} to {route[‘passenger’][‘destination’]}”)
   Time.sleep(300)  # Simulated time delay, e.g., 5 mminutes
```

OUTPUT ;

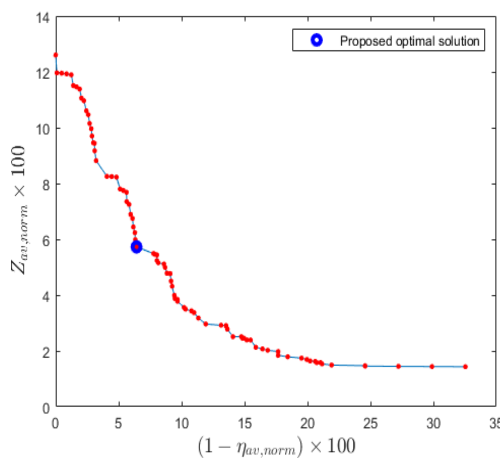


Node to Node Route Planning

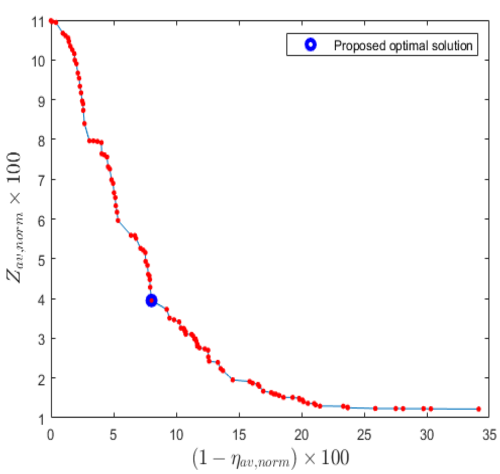
From Node	To Node					
	Entrance	Exit	Bus Stop	Fare Gate	Bus	MRT
Entrance			Start Trip	Start Trip		
Exit						
Bus Stop		End Trip	**Walking**	**Walking**	**Boarding**	
Fare Gate		End Trip	**Walking**			**Boarding**
Bus			Alighting		**Travelling**	
MRT				**Alighting**		**Travelling**



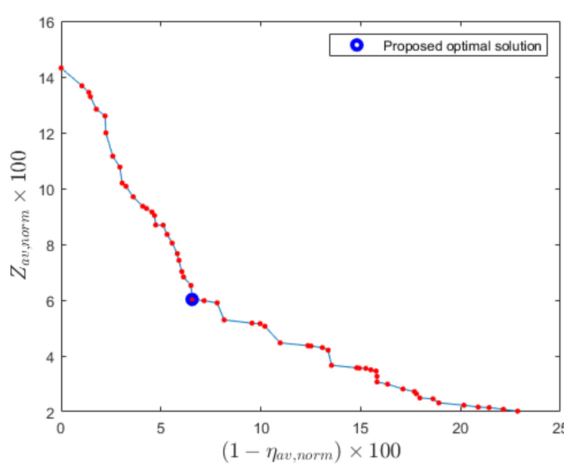
(a)



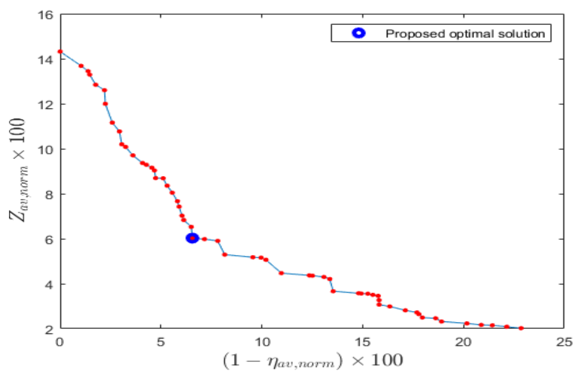
(b)



(c)



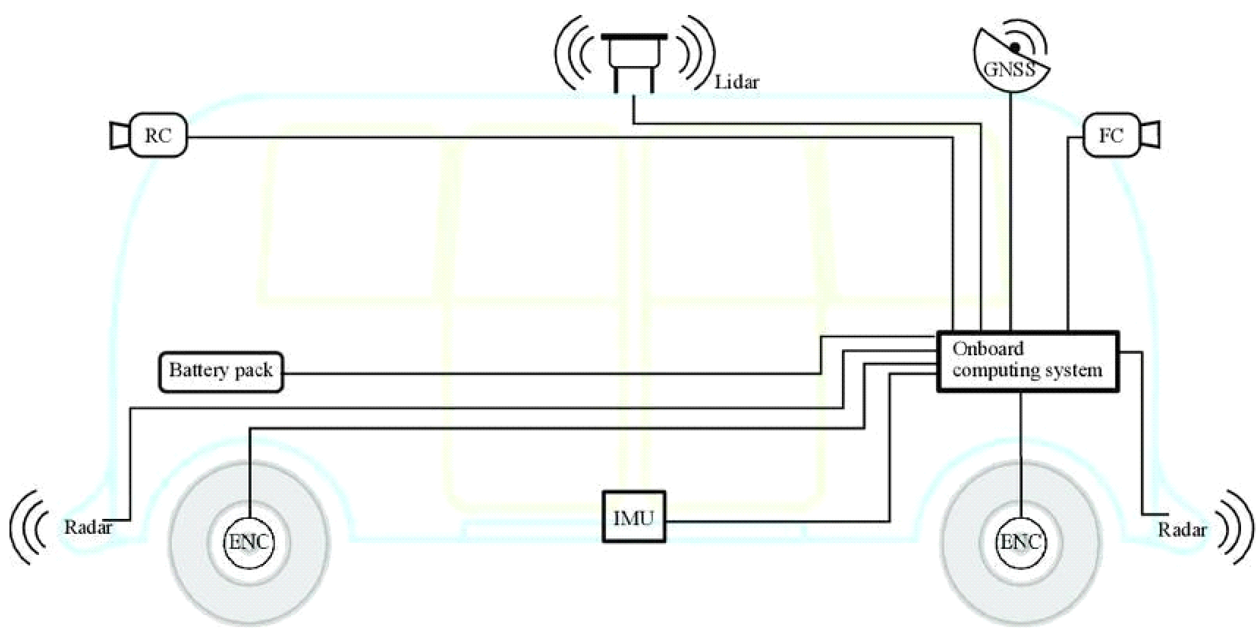
(d)



Proposed optimal solution
$Z_{av,norm} \times 100$
$(1 - \eta_{av,norm}) \times 100$

(e)

LAYOUT:



Lidar
GNSS
RC
FC
Battery pack
Onboard computing system
Radar
ENC
IMU
ENC
Radar

Public transport optimization including sensors and components:

In. Public transport optimization using Various sensors and components are usedin public transport systems to enhance efficiency, safety, and passenger experience. Some of these include:

- GPS (Global Positioning System) Sensors
- Passenger Counting Sensors
- CCTV Cameras
- Ticketing and Fare Collection Systems
- Vehicle Health Monitoring Systems
- Automated Announcements and Information Displays
- Wi-Fi and Connectivity Components
    - GPS (Global Positioning System) Sensors: Used for real-time tracking and monitoring of vehicles to provide accurate location information.

    - Passenger Counting Sensors: Employed to measure the number of passengers boarding and alighting at different stops or stations, aiding in demand analysis and resource allocation.

    - CCTV Cameras: Installed for security purposes to ensure passenger safety andmonitor any incidents that may occur on the vehicles or at the stations.

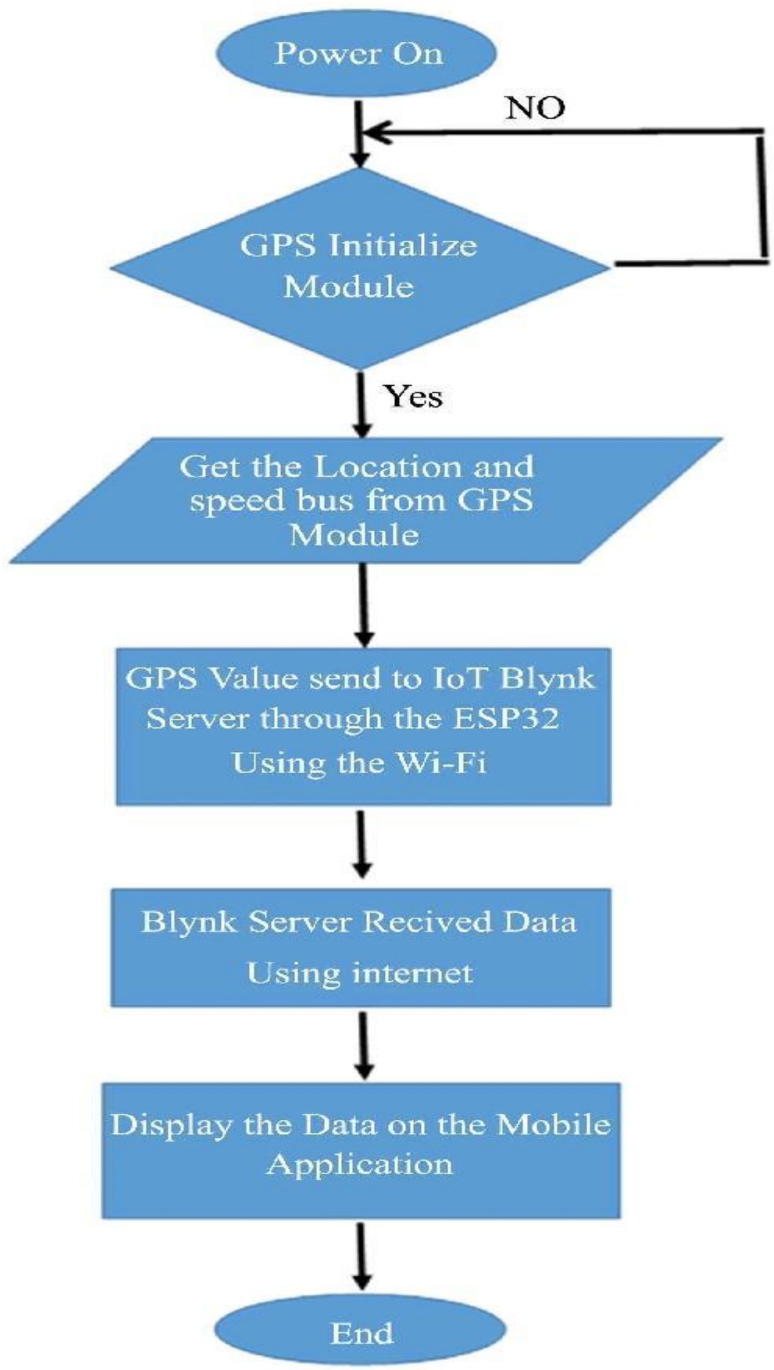
    - Ticketing and Fare Collection Systems: Components such as ticket machines, contactless smart cards, or mobile payment systems are utilized for convenientand efficient fare collection.

    - Vehicle Health Monitoring Systems: Sensors are employed to monitor the health and performance of the vehicles, including engine health, fuel efficiency,and maintenance requirements.

    - Automated Announcements and Information Displays: Components such as audio systems and digital displays are used to provide passengers with real-time information about routes, schedules, and any important updates.

- Wi-Fi and Connectivity Components: Equipped to provide passengers with internet access and connectivity during their commute, enhancing the overall passenger experience.

FLOW CHART ;



IoT module To establish the communication between urban bus and passenger's smartphone IoT

IOT MODULE :

Module is used. If we use Bluetooth it will provide only short range communication, so we are using IoT module.The Internet of Things (IoT) is internetworking of physical devices with electronics and network connectivity that control these objects to collect and exchange the data. IoT is not only used to sense the information but also to interact with the physical network.SIM800C is used here for the transmission of data and it is a complete quad-band GSM solution, which can be embedded in the customer applications. These modules are the sub-system of the internet-of –everything hardware. It Supports quad-band of 1900MHz and it can transmit voice, SMS and data information with low power consumption. It can smoothly fit into slim and compact demands of customer design.

By using the smart information system, the public bus usage can be improved and so private modes of transportation get reduced. It will play vital role in controlling traffic congestion and pollution .This system is created and developed using simple and cost-efficient components. It can be easily installed inside the bus because of its small size. We have tested our project in all the five stops that we have taken as reference location. All the other units in the system, including sensors, GPS module and power unit are tested and are found to be in working condition. The outputs taken at SRIT Parking when the bus moves in forward direction i.e., SRIT Parking to Perur is shown in the figure 4 and figure 5.For Internal passengers the current location is displayed as "SRIT Entrance", the next location is displayed as "SRIT parking" and the time taken to reach SRIT is displayed as "5 Minutes" in LCD Display . For External Passengers, the information will be displayed in the webpage www.iotclouddata.com/project/305/iotview.php as shown in the figure 6.The maximum capacity of the bus is fixed as 55 in our proposed model. Difference between the maximum seating availability of the bus and the number of passengers inside the bus is displayed as vacant position for external passing

Table 1. Bus route details when bus moves in forward direction S.No Source Destination Approximate time to reach the destination (in mins) 1 SRIT parking SRIT entrance 5 2 SRIT entrance Pachapalayam 10 3 Pachapalayam Chettipalayam 10 4 Chettipalayam Perur 7 The outputs are checked at each stop when the bus moves in reverse direction i.e., Perur to SRIT Parking also. Bus route details when bus moves in forward direction are shown in the table 1.

TESTS AND RESULT :

This section presents the tests and results of our proposed approach. A realistic scenario was selected to perform the experimentation of the mobility system on a bus route of a city in Mexico, specifically in the city of Cuernavaca, which is in the central region of the country. This city is a good example of most of the common cities in Mexico, where there are no schedules for public services and, therefore, the arrival times are completely chaotic. The experimentation considers a real route that crosses the city from south to north. The first phase for the experimentation was the design of the route (which is called Route 1) with 66 bus stops: 28 bus stops on the Guacamayas-University journey and 38 bus stops in the opposite direction (University-Guacamayas journey). Figure 18 shows both directions of Route 1. The two directions of the route are not symmetrical because some of the streets on the route are one-way streets. It is important to point out that the route needs to be designed with a software application because there are no GTFS specifications for most small Mexican cities. The GTFS specification is only available for big cities, such as Mexico City or Guadalajara. Eight trips were monitored on this bus route for experimentation; four trips for the Guacamayas–University journey and four trips for the University–Guacamayas journey. One relevant point in testing was having very similar traffic conditions in each journey in order to reduce the error rate in arrival time predictions. To do this, the test for the journeys in the Guacamayas–University direction were done starting at 18:00 h. and the journey for the

University–Guacamayas direction were done starting at 19:00 h. As mentioned above, the Guacamayas–University direction is composed of 28 bus stops. Ten of these bus stops were selected in that direction and another 10 bus stops were selected in the opposite route (University–Guacamayas) to be monitored in the tests. The arrival times were determined using the Passenger Transport app and were compared with the real time of the arrival of the bus to the bus stop. Both the predictions and the real times were registered in a database for further evaluation. In the evaluation, the proposed system was used to obtain the expected arrival time of the next bus. The time when the system is consulted and the time when the bus arrives to the bus stop are registered. The (Actual arrival time) Real-Time Arrival is the difference (in seconds) between the time when the software system gives the prediction and the time when the bus is currently arriving to bus stop. The Predicted Time (Predicted arrival time) represents the waiting time predicted by the proposed system. The results obtained for experimentation are presented in Tables 1–8. Each table represents one of the eight trips: 4 trips for the Guacamayas–University journey and 4 trips for the University–Guacamayas journey. The first column indicates the number of stops being monitored. The second column indicates the Actual Arrival Time which, as commented above, represents the difference (in seconds) between the time when the software system gives the prediction and the time when the bus is currently arriving at bus stop. For example, in the first row, the time when the system was consulted to predict the arrival time was 18:18:54 h, and the time when the bus actually arrived to the bus stop was 18:20:14 h. Therefore, the difference between the two times is 80 s. In order to simplify the visualization, only the difference in seconds is represented in the table. The third column indicates the Predicted Arrival Time, which represents the waiting time predicted by the proposed system. In the example of the first row, the time predicted (in our system) for the arrival time was 79 s. The fourth column indicates the difference between the Actual Arrival Time and the Predicted Arrival Time, which in the example was 1 s. The Fifth column represents the absolute percentage error, and the sixth column represents the precision of the proposed prediction algorithm, which in the example is 98.75%. The difference in seconds between the estimated and real arrival times was also registered together with the medium absolute error (MAE) absolute percentage error (MAPE), and the precision of the prediction for each stop. The average MAE, MAPE, and Precision are specified for each table, for example, for complete Trip 1, the average precision was 87.2%.

Proposed software for simulations of transfer nodes as elements of a public transport system

Public transfer nodes should be simulated as elements of a public transport system. All the technological processes in transfer nodes are defined or influenced by processes of other elements of the public transport system. To model processes of the public transport systems functioning for solving scientific problems, we developed the specialized library of base classes. The classes' implementation was performed with the use of the Python programming language; it ensures compatibility of the developed software with the most popular environments for modeling of public transport systems (including Aimsun and PTV Visum). The developed code of the mentioned base classes is available in open access and could be downloaded from [13]. As the base classes, on the grounds of which simulation models of public transport systems should be implemented, we consider:

- Net: is used in order to develop the software implementation of a transport network model as an oriented weighted graph;
- Node: allows researchers to model points of the transport network as the graph nodes; the transport net points could be considered in a simulation model as software implementation of the public transport stops, transfer passenger nodes of intersections of the road network;
- Link: represents a software implementation of a link in a graph, which is used in order to describe a transport system model; the graph link could be used in

simulation models for modeling segments of the road network or spans of the public transport lines;

• Line: could be used in order to model a public transport line; is defined for the software implementation of a road network as an object of the Net class;

• Vehicle: allows to model a vehicle as an element of the transport system model; is used for developing simulation models of the public transport lines;

• Passenger: is an abstraction for implementation of passengers as transport system elements; an object of this type is a unit used for description of demand for services of the public transport within the framework of a simulation model of the transport system.

To simulate parameters which describe the environment influence on the transport system, in the proposed library the Stochastic class was developed. Implementation of the Stochastic class objects allows to model a random variable with the defined distribution and given numerical parameters.

Implementation And Testing Results;

The GSM proved to be troublesome in the originally proposed design. The main issue was using the GSM for the communication with the server, as multiple tests proved that we cannot use this feature. Furthermore, the module itself was not responsive to the AT commands regarding the TCP/IP communication when tried with various processors, controllers and even using a TTL to USB converter which is an adapter used to directly convert from the standard TTLUART serial communication to USB so that it could be connected to the computer directly and after running the commands using a terminal application. To mitigate these problems, a change to the original design is proposed [25]. The second module that was replaced was the LCD screen. As mentioned earlier, that one of the core objectives is the dynamic response to the demand, which means that turn-by-turn navigation is a must. But, google maps navigation is only supported by mobile devices. Alternatives to google maps were tested like “navit” and “mapbox, navit” uses street maps which most users are not familiar with and did not include Arabic street names which would be troublesome for the drivers besides “mapbox” as well, does not support progressive web applications (PWA) [26] which is essential as our application is built based on this technology. To solve the encountered problems during the first phase of implementation, a new system was proposed whereby the GSM module is replaced with a portable Wi-Fi module. This will allow the raspberry pi to access the internet and communicate with the server. Another addition is an android tablet, which can connect to the same network as the RPi. Adding the tablet will provide a sturdier, more familiar interface, as well help with meeting the requirement of turn-by-turn navigation [27]. The raspberry pi runs two concurrent threads, one which facilitates the communication with the main server to send the connected modules’ data and receives the specific trip Id and driver name. The other thread is a locally hosted server on the RPi, which takes the trip ID from the first and gets the trip data, then layoff the route to the tablet.

Results Analysis And Reflection;

As detailed in previous sec 4 around our proposed system problem formation, we adopted MDPDR VRP for this application, were leveraging two well-known LSTM-RNN implementations: Single-variable time series forecasting LSTM, and Tensorflow RNN. Start by visualizing the dataset in LSTM to gain some information about the data trend and seasonality. After loading the data into a data frame, it is normalized using a Min-Max Scaler preprocessing class and split into 80-20% for training-testing respectively. From there, the dataset is split again from its current NumPy array form into a sequential dataset with several previous time steps to be used as input variables to predict the next time period. After running the model for 100 epochs, we achieve at raining score of 22.93 RMSE and a test score of 47.53 RMSE Compared with the systems discussed in the literature review section, our proposed system provides the same livetracking functionality. But, unlike Pham, Micheal and Chi’s system [23], our proposed system is more reliable regarding data handling. While they use a GSM module to send SMS messages to the user’s phone, our proposed system constantly sends the tracking data to the server where then it is shown to both business operators through the dashboard and to the user via the application. This provides more transparency and reliability. Even though Petracca, et al [22] system provides similar functionality as our proposed one, still with our system there is no need for extra infrastructure components like the Road Side Units (RSUs), [35] as the on-board data are transmitted directly to the main server without the help of outside modules. The main edge is to use the machine learning and optimization techniques detailed in the problem formulation system to optimize the overall operating cost in terms of dynamic route distance formulated in terms of PLF as result from the MDPDR VRP adopted methodology

Conclusion;

The need for demand-responsive mass transit is apparent. The main challenges faced by the traditional bus system include lack of digital infrastructure, outdated distributions, and schedules, as well as outdated or inefficient routes. The proposed solution attempts to tackle these problems by operating in the fuzzy domain, making predictions about passenger demand, and preemptively customizing and optimizing routes and schedules to as many passengers as possible. This is in addition to the light-weight mobile application and the RFID payment system which will enhance the overall user experience. The benefits of such a system would extend across social, economic, and environmental improvements. While there have been similar attempts at solving the various sub-problems present within the traditional bus system, very few have attempted to re-imagine the overall experience and challenge the culturally established norms about the bus system. We still aim to improve the current results of cost optimization by adopting reinforcement learning and assess the implication on PLF calculations and bus route optimization and cost projections.

References ;

[1] Central Agency for Public Mobilization and Statistics," Egypt In Figures", 2015.

[2] SWVL," Egyptian start-up Swvl raises USD 8 million in series A funding round", 2018.

[3] K. Ltd," KPIT — On-Bus Intelligent Transport System", KPIT Technologies Ltd, 2018. Available at: https://www.kpit.com/what-wedo/products/on-bus-its.

[4] B. Hamilton, History of intelligent transportation systems, U.S. Department of Transportation, 2016.

[5] T. Matters, Facts & figures", Transport for London, 2018. Available at: https://tfl.gov.uk/corporate/about-tfl/what-wedo/londonunderground/