Collaborative Filtering Recommender System for Timely Arrival Problem in Road Transport Networks Using Viterbi and the Hidden Markov Algorithms

Ofem Ajah Ofem, University of Calabar, Nigeria
Moses Adah Agana, University of Calabar, Nigeria*
Elemue Oromena Felix, University of Calabar, Nigeria

ABSTRACT
In this study, a timely arrival recommender system (TARS) using Viterbi and hidden Markov Model (HMM) was developed. Ratings from current road users were used as inputs and trained to provide recommendations to prospective road users on the best routes to follow to get to their destinations from any source in time. The system was deployed on Android devices and iPhones with Google map. Real time data on current road conditions were collected from twenty-one (21) bolt drivers in Calabar Metropolis traversing various routes from Unical to Watt Market. The system could calculate arrival time in km/h, generate nearest nodes on each route, detect routes with free or congested traffic flow, and then recommend the best route in real time to users for timely arrival. The application, if adopted, can aid road users to save time, cost, and reduce stress on both humans and the vehicles used.

KEYWORDS
Arrival, Collaborative, Networks, Ratings, Recommendation, Road, Timely, Transportation

INTRODUCTION
One of the major problems affecting global transportation system is timely arrival of travelers at their destinations. Transportation, whether public or private, plays an important role in peoples’ lives by helping them to move from one place to another, providing them an easy means of transporting their goods and services for optimum business transactions. There are different means of transportation; some people choose to travel by air, water or by land/road.

Road transportation involves the movement of people and goods from one point to another on roads. On roads, motor vehicles and animals are used to move humans, goods and services. A road is a land route that traverses between two destinations, constructed to enable transportation by way of motorized and non-motorized carriages (Nanzip, 2020). People engage in transportation for various reasons such as business, tourism or socialization. In most cases, many people travel via a given
route at the same time of the day resulting in congestion on such routes. Consequently, commuters often get disgruntled with time wastage on the road, stress, and untimely arrival at their destinations.

The quality of a road transport system can be improved upon with the help of modern technology and intelligent transportation systems (ITS), resulting in more reliable and convenient road transport services. Accordingly, collaborative filtering recommender systems (CFRSs) can be suitable for optimum road transport services (Alexander, et al., 2017). Recommender systems are algorithms designed to make suggestions about items that are relevant to users. Such systems are designed to help users filter out irrelevant targets and recommend targets that may be of help to them. A recommender system helps a user to make choices following recommendations by the system when there is no sufficient personal knowledge or experience of the available alternatives (Isinkaye, Folajimi & Ojokoh, 2015).

Recommender systems are software tools and techniques for suggesting items to users by considering their preferences in an automated fashion. The suggestions provided are aimed at supporting users in various decision-making processes. Technically, recommender systems have their origins in different fields such as Information Retrieval (IR), text classification, machine learning and Decision Support Systems (DSS). They have proven to be worthy tools for online users to deal with information overload and have become one of the most popular and powerful tools in E-commerce (Mehrbakhsh et al., 2013).

Recommender Systems (RSs) are used in a variety of areas, with commonly recognized examples taking the form of playlist generators for various services desired by users. These systems can operate using a single input or multiple inputs within and across platforms. Since recommendations are usually personalized, different users or user groups receive diverse suggestions. In addition, there are also non-personalized recommendations (Jannach, 2016). Recommender systems development is initiated from quite a simple observation: individuals often rely on recommendations provided by others in making routine, daily decisions (McSherry & Mironov, 2019). In simple form, RSs try to predict what is most suitable based on the user’s preferences and constraints. In completing such a computational task, RSs collect from users their preferences, which are either explicitly expressed, (e.g. ratings for path), or are inferred by interpreting user actions. In the case of road transportation, path recommender systems assist road users to consider a path to take before traversing such path.

A path or route recommender system can be used to identify multiple optional paths available in a given location, influenced by user rating or preferences using a subclass of information filtering system that seeks to predict the alternative desired route to the destination. The Collaborative Filtering Recommender System (CFRS) can provide such guidance on what path or route to choose among the vast available options that can lead to timely arrival (via rating or preference). The system can be applied for the provision of dependable traffic information to help road users including pedestrians to make better route choice decisions.

Timely arrival is a priority to every road user, saves transportation cost, promotes productivity and optimizes profit maximization. Timely arrival is measured by a road user’s ability to arrive at the desired destination from a particular source on time and is stochastic due to the unprecedented dynamic nature of both travel time and the waiting time during transit services. In the research community, however, the problem of finding good alternative routes to arrive on time has received far less attention than the classical shortest path problem. Most of the techniques that accelerate shortest path computations in a specific setting often rely on a unique representation of distance between any given locations. Given this distance, they operate by pruning as much of the road networks from a search as possible, thus discarding anything that does not describe an optimal path. As a result, these techniques complicate the process of finding good alternative routes. Each of the possible routes may only be optimal in a setting described by the personal preferences of a group of users (Moritz, 2015).

Almost all transportation systems have timetables that designate vehicular arrival at scheduled stops. Road transport systems are highly utilized where services run on time, as anyone planning on making use of the service can program his/her activities with that of the transport system (Moritz,
The proficiency of a transport model to solve any timely arrival problem depends largely on the degree affected by external factors (like road congestion, length of road, etc.). For road users to therefore arrive their destinations on time, the transport system should be able to provide accurate estimated time of arrival (ETA), generate alternative routes (paths) to the traveler’s destination, and predict which routes or road segments are congested or smooth for timely arrival.

Different models and algorithms have been developed to calculate travelers’ estimated time of arrival (ETA) and generate alternative routes to a traveler’s destination, and predict which routes are most suitable for timely arrival of travelers. For example, Jiang and Fei (2015) proposed a vehicle speed prediction method on road segments using Neural Networks (NNs) based on historical traffic data. The forward-backward algorithm was applied to extract vehicular speed on each road segment along the driving route. The model is not however efficient in determining early time of arrival. Woodard, et al. (2017) developed a method known as TRIP to predict travel time on an arbitrary route in a road network at an arbitrary time. This does not however have the capability of providing alternative routes for early arrival.

One of the most effective models for timely arrival problem is the Hidden Markov Model (HMM). The HMM is a time series model based on the Markov model. It establishes the state transition mechanism of two types of related variables to achieve a prediction. The HMM can be represented as a directed graph with \( N \) states where each state can emit either a discrete character or a continuous value drawn from a Probability Density Function (PDF). The model is widely used to solve scientific problems including road transportation (Miller et al., 2015).

Collaborative filtering (CF) is one of the most used approaches for providing recommendations in various online environments. Collaborative recommendation methods have been widely utilized due to their simplicity and ease of use (Ben et al. 2019; Ekstrand et al., 2011; Shi et al., 2014). This method relies on a database of ratings submitted by each user for products or services, then the ratings are compared to each other with the use of suitable similarity methods in order to provide recommendations to the user who makes the request. The main two functions of such systems are to identify a pre-specified number of neighbors according to similar ratings and then provide the recommendations.

Cheng, et al. (2014) opined that the most important link in the recommendation system is the recommendation algorithm, among which the most well-known and widely used algorithm is the CF algorithm, whose ideas are scientific and operable, and its recommendation results are also more accurate.

CF algorithm is used for recommending other items that are of interest to other users with similar hobby interests to the target user. The first step in the algorithm is to construct a user-item matrix, which is based on whether the user is interested in a certain item. The second step is to calculate the set of similar users of the target by cosine similarity according to the item matrix (Ben et al., 2019); finally, the user-item matrix is calculated by \( K \), the nearest neighbor.

Another recommender algorithm is content-based filtering algorithm. Content-based filtering and collaborative filtering have long been viewed as complementary (Adomavicus & Tuzhilin, 2005). Content-based filtering can predict relevance for items without ratings (e.g., new items, high-turnover items like news articles, huge item spaces like web pages). Collaborative filtering on the other hand needs ratings for an item in order to predict for it, while content-based filtering needs content to analyze, and content can be scarce in some domains (e.g., movies, music, restaurants, and books without text reviews available) (Balabanovic & Shoham, 2007).

Researchers generally believe that collaborative filtering leads to more unexpected or different items that are equally valuable. Some people call this property of recommendations novelty or serendipity. However, collaborative filtering has also been shown to over-specialize in some cases. Content-based filtering (CBF) and collaborative filtering may be manually combined by the end-user specifying particular features, essentially constraining recommendations to have certain content
features. More often, they are automatically combined, sometimes called a hybrid approach. There are many ways to combine them, and no consensus exists among researchers (Basu, Hirsh, & Cohen, 2018).

People navigate over 1 billion kilometers a day using mobile routing services, and many of these services provide travelers with information on transit networks. In most of such applications, given an origin-destination pair and a desired departure or arrival time, the route associated with the minimum expected trip time is provided to the user (Pankaj et al., 2018; Baran, Dziech, & Zeja, 2018; Townsend, 2017). However, the uncertainty associated with these recommendations, either due to variability in travel time or the transit service headway, is rarely accounted for. The degree of risk aversion of transit passengers highly affects their route choices (Szeto et al., 2016).

In the course of carrying out their day-to-day activities as people move from one location to another, being able to arrive at various places or deliver different items on time is vital for overall productivity and sustainable development, and this requires a good road network that can facilitate easy movement devoid of traffic jam.

A good number of techniques that accelerate shortest path computations in a specific setting exist. Most of these techniques, however, rely on a unique representation of distance between any given locations. Given this distance, they operate by pruning as much of the road network from a search as possible. This process discards anything that does not describe an optimal path. A stochastic time-dependent (STD) network will therefore be a more realistic representation of an actual road network compared with the deterministic one. There is no unique definition of an optimal path under uncertainty. One possible and common solution to this problem is to find a path with the minimum expected travel time. Due to time variability, some researchers use a distribution of travel time instead of expected value of it so they find the probability distribution of total travelling time using different optimization methods such as dynamic programming and Dijkstra’s algorithm (Olya, 2014).

Road users often face adverse road conditions before they arrive at their destinations. They experience delays on the road due to traffic congestions and jam which may result from several factors including: accidents, checkpoints by law enforcement agencies, flood as a result of heavy rainfall, pot holes, failed bridges and failed vehicles on the road, etc. Consequently, these contribute to late arrival at their destinations.

In order to arrive at their destinations on time, road users may have to make complex decisions regarding which route to take in case of adverse road traffic situations. These complex decisions arise from the uncertainty about available routes to navigate in order to arrive at the destination on time (Szeto et al., 2016).

Traditional collaborative filtering algorithms suffer from the lack of available ratings. In this situation, road users may not have the real time information of the traffic situation on the route they are plying. This could lead them to taking routes that are currently experiencing adverse traffic situations, thereby affecting their arrival time.

For this reason, this study has focused on developing a more efficient recommender system capable of providing an estimated time of arrival (ETA) to facilitate timely arrival for road users based on Viterbi Algorithm and the hidden Markov model (HMM). The several existing algorithms for timely arrival problem appear less efficient, hence the combination of two models in this research to provide a more robust and efficient recommendation on timely arrival to road users. The model has been designed to generate alternative routes to a traveler’s destination, as well as predict which routes or road segments (nodes) that are congested, less congested or have free traffic flow for timely arrival of travelers to their destinations.

The choice of the combination of the Viterbi algorithm with the hidden Markov model (HMM) is to apply them to investigate the collaborative filtering for timely arrival due to their dynamic programming capability for obtaining sequential estimate in solving stochastic path problems with an additional probabilistic weight on each node. This will help to provide answers to frequently asked questions (FAQ) such as: What is the traffic condition? Which path has the fastest travel time between two points on the road? And how fast is the path? The implementation of a collaborative
filtering recommender system can be used to guide people’s journeys by deciding the time space dependent on ratings given by immediate users and will also help to educate the road users on the state of various routes in a city for easy traversal. This can boost the economy via enhanced mobility, thus reclaiming lost productive hours (due to traffic jam), restore quality of life (caused by stress from long stay on the road and air pollution from vehicles’ exhaust pipes), increase safety on the road and reduce global warming.

RELATED WORKS

In a study by Yang, Sebastien and Samitha (2018), network structure was proposed to the online decision-making of a passenger, including boarding, waiting, and transferring and considering the stochastic on-time arrival problem in transit networks where both the travel time and the waiting time for transit services are stochastic. A specific challenge of this system was the combinatorial solution space due to the unknown ordering of transit line arrivals. Furthermore, the design of the framework was based on a dynamic programming algorithm that is pseudo-polynomial in the number of transit stations and travel time budget, and exponential in the number of transit lines at a station, which is a small number in practice.

Fan, Kalaba, and Moore (2015) applied Bellman principle of optimality to formulate the mathematical model for identifying dynamic routing policies in stochastic transportation networks. This did not however place emphasis on timing. Similarly, Dijkstra (1959) developed an algorithm to solve the shortest path problem. This has been applied with limitations in determination of shortest routes in a road network because timing is not critical in the algorithm (Rohila & Gouthami, 2014).

Tingting et al. (2020) designed a congestion pattern prediction model for a busy traffic zone based on the hidden Markov model (HMM). The model was designed to establish a correlation between the external road traffic state (observation state) and internal road traffic state (hidden state) of a busy traffic zone in the Chinese Ningbo City. Traffic states were acquired by cleaning and mining floating vehicle trajectory data, which were used to calibrate the HMM in order to predict the zone congestion pattern. This study demonstrated the validity and rationality of the model as results showed that prediction accuracy can reach 83.4%, which is 5.8% higher than that of the autoregressive moving average model. This result illustrated the feasibility and effectiveness of the approach in the field of congestion pattern prediction for busy traffic zones. The model focused more on congestion prediction than on recommending alternative routes and early time of arrival.

Qi and Ishak (2014) proposed a stochastic approach Hidden Markov Model (HMM), for short-term freeway traffic prediction during peak periods. Data used for the study were obtained from a six-year real-time traffic monitoring device on a 60.8-km corridor of Interstate-4 in Orlando, Florida. The HMM was used to estimate the most likely sequence of traffic states. The model performance was evaluated using prediction errors, which are measured by the relative length of the distance between the predicted state and the observed state in the two-dimensional space. Results showed that minimal prediction errors lower than or equal to (£) 10% were obtained from HMMs. Also, the model performance was not remarkably affected by location, travel direction, and peak period time. The HMMs were compared to two naïve predication methods. The results showed that HMMs perform better and are more robust than the naïve methods.

Wang et al. (2015) developed a hidden Markov model (HMM) based on traffic estimation model, in which the traffic condition on a road segment is considered as a hidden state that can be estimated according to the conditions of road segments having similar traffic characteristics. Data used for the study were floating car data (FCD). An algorithm based on clustering and pattern mining rather than on adjacency relationships was used to find clusters with road segments having similar traffic characteristics. Results of experiments based on real FCD confirmed the applicability, accuracy, and efficiency of the model in estimating urban traffic congestion.
Relationship Between Viterbi Algorithm (VA) and Hidden Markov Model (HMM)

The Viterbi Algorithm (VA) can be simply described as an algorithm which finds the most likely path through a trellis, i.e. shortest path, given a set of observations. The trellis in this case represents a graph of a finite set of states from a Finite State Machine (FSM). Each node in this graph represents a state and each edge a possible transition between two states at consecutive discrete time intervals. The FSM referred to here is commonly used in digital electronics and is often referred to in the literature as a Markov Model (MM). For each of the possible transitions within a given FSM there is a corresponding output symbol produced by the FSM. The outputs of the FSM are viewed by the VA as a set of observation symbols with some of the original data symbols corrupted by some form of noise. This noise is usually inherent in the observation channel that the data symbols from the FSM have been transmitted along (Ben et al., 2019).

Another type of FSM is the Hidden Markov Model (HMM). As the name suggests, the actual FSM is hidden from the VA and has to be viewed through the observations produced by the HMM. In this case the trellis’s states and transitions are estimates of the underlying HMM. In either type of model, MM or HMM, the VA uses a set of metrics associated with the observation symbols and the transitions within the FSM. These metrics are used to cost the various paths through the trellis, and are used by the VA to decide which path is the most likely path to have been followed, given the set of observation symbols.

Viterbi algorithm may be viewed as a solution to the problem of maximum a posteriori probability (MAP) estimation of the state sequence of a finite-state discrete-time Markov process observed in memoryless noise. In the Markov process, at any time \( t \), the probability of being in any given state is the product of being in a state at \( t - 1 \) and the probability of changing state.

METHODOLOGY

The hidden Markov model (HMM) based on Viterbi algorithm was applied to a collaborative filtering recommender system for timely arrival in road transportation networks. The Viterbi’s algorithm was used due to its dynamic programming capability for obtaining sequential estimate in solving stochastic path problems with an additional probabilistic weight on each node. Similarly, the Hidden Markov Model (HMM) was used because it provides discrete-time stochastic processes whose behaviors are not directly observed but whose link cost can be derived from the probability of the corresponding transition-observation pair.

The map of various routes in Calabar Metropolis, Cross River State, Nigeria was adopted as shown in Figure 1.

A recommender system was designed and analyzed to determine the shortest possible path from a certain start node during congestion based on rating or preference; and to evaluate and compare traveling time for various routes; then to find the most suitable path considering traffic jam dependent on time. The road network recommender system was designed to serve as solution to the timely arrival problem.

Stage 1 – Problem identification: This stage was concerned with identification of the research problem; which is timely arrival of road users from a known source (University of Calabar) to their target destination (Egerton roundabout at Watt Market), with a view to providing alternative routes and predicting traffic congested as well as free segments of the road.

Stage 2 – Problem analysis: At this stage the problem was analyzed to determine its effects on human and vehicular movement as well as socio-economic activities in the study area.

Stage 3 – Feasibility study: This stage considered availability and feasibility of resources to aid the development of the recommender system for timely arrival.

Stage 4 – System requirements: This stage considered the important parameters required to design the system.
Stage 5 – System analysis: At this stage an analysis of the system was carried out to determine its viability and sustainability before resources were committed to design and implement the recommender system. The cost and benefits analysis of the system were taken into consideration.

Stage 6 – System design: This stage is when the actual design of the system was done.

Stage 7 – System implementation: The system was implemented to achieve desired results.

Stage 8 – System evaluation: This is the stage at which the performance of the system is evaluated. The system evaluation is continuous as new routes and complexities arise. Each route has to be tested for timely arrival and congestion separately. Any observed failure at this stage could result in redesign or modification of the system.

Stage 9 – Solution to the problem: This stage is the final stage when the objective of designing the recommender system was finally achieved.

Three major routes were considered, and these are:

1. Unical to Egerton roundabout (Watt market) via Mary Slessor – Bogobiri and Calabar road.
2. Unical to Egerton roundabout (Watt market) via Goldie street through OrokOrok roundabout.
3. Unical to Egerton roundabout (Watt market) via Mount Zion, Palm Street, and Calabar road.

Data Collection

A total of twenty-one (21) volunteer road users (mainly Bolt drivers) in Calabar Metropolis were selected for the purpose of collecting experimental data used to develop TARS application.

Research Design Steps

The following steps were adopted for the research design:

1. Model formation/specification
2. Parameter Definitions of HMM for Route Predictions
3. Algorithm formation
4. Architecture of Timely Arrival Recommender System (TARS)
5. Experimental design
Model Formation

The system was modeled to identify congested road segments as well as provide alternative routes and nodes for timely arrival. Road-users’ ratings based on observations of different routes traffic information were collected to determine timely arrival from source (UNICAL) to the destination (Egerton Roundabout in Watt Market). The model was designed to predict the available nodes, in a given route, for road users to traverse at different times of the day in order to arrive at their destinations on time.

The traffic congestion prediction sub module was mainly used to predict congested segments (nodes) on each route at different times of the day. At any time $T_k$, the congestion level $R(T_k, R_i)$ of each road $R_i$ is denoted by the total number of possible driving routes on the road $R_i$ in a time period. The higher the value $R(T_k, R_i)$ is, the more congested the road $R_i$ is.

The vehicle route recommendation sub module collects information about a driver’s location, just-driven road segments and destination, and then introduces better alternative routes for drivers based on traffic congestion situations at each given time, following existing recommendations from similar road users.

The Hidden Markov Model (HMM) deals with both observed events and hidden events as casual factors in probabilistic model. It is applied depending on new input driving routes. The given corpus of training samples may not fully include all of possible driving routes. With the increase of inputting driving routes, the amount of training data for the HMM will also grow, which could improve the prediction accuracy.

To achieve this model, the following tools and equipment were required:

1. A map of Calabar Metropolis showing all the road networks leading from University of Calabar to Egerton Round-about in Watt Market.
2. The length (distance) and condition of each route and each node from the source to destination using measuring wheel.
3. GPS enabled mobile devices (phones) to track a drivers’ location from the source to destination.

With the help of these set of tools and system requirements, it was possible to obtain the physical distances between all the available nodes in the road networks leading from the source (Unical) to the destination (Egerton roundabout). Also, it was possible to determine the time taken for a road user to traverse the driving route from source to destination through an interconnected network of nodes. From the simulated network, the free routes for a road user to traverse from a source node to a destination node is determined using the recommender system.

Parameter Definitions of HMM for Route Predictions

To predict the various available nodes leading to a road user’s destination, it is necessary to input a driver’s just-driven path represented by coordinate points into HMM and then output the entire paths leading to the destination. Just-driven path can be regarded as an observation sequence and the corresponding sequence, which is composed of different nodes that are leading to the destination. The process of the HMM construction, based on the number of routes ($R_1$, $R_2$, $R_3$, … $R_n$) and the number of nodes (A, B, C,…K) from the source to the destination is as illustrated in Figure 2.

From Figure 2:

$R_3 = \langle A \rightarrow B \rightarrow C \rightarrow F \rightarrow I \rightarrow L \rightarrow K \rangle$

$R_2 = \langle A \rightarrow B \rightarrow E \rightarrow H \rightarrow K \rangle$

$R_1 = \langle A \rightarrow D \rightarrow G \rightarrow J \rightarrow K \rangle$
Assuming that R₁, R₂, and R₃ are routes, in order to distinguish different vehicle routes leading from the source point (A) to the destination point (K), the observation set V includes the starting symbol (<), the end symbol (>), and different coordinate points (nodes) (A→K). Each observation is defined by \( p_{ij} \), where \( i \) is the number of routes \( R_i \) in the node set and \( j \) is the number of coordinate points (nodes) in each route \( R_i \). For instance, assuming that the nodes n = (A, B, C, D…K) are represented by integers (1, 2, 3, 4, …11), the observation sequence of the routes R₁, R₂, and R₃ is illustrated in Figure 3.

From Figure 3:

\[
\begin{align*}
R_3 & \rightarrow <A \rightarrow (0, 1), B \rightarrow (1, 2), C \rightarrow (2, 3), F \rightarrow (3, 8), I \rightarrow (8, 9), L \rightarrow (9, 10), \text{ and } K \rightarrow > \\
R_2 & \rightarrow < A \rightarrow (0, 1), B \rightarrow (1, 4), E \rightarrow (4, 7), H \rightarrow (7, 10), \text{ and } K \rightarrow > \\
R_1 & \rightarrow < A \rightarrow (0, 5), D \rightarrow (5, 6), G \rightarrow (6, 11), J \rightarrow (11, 10), \text{ and } K \rightarrow >
\end{align*}
\]

Algorithm Formation

The solution to the timely arrival problem is corresponding to a HMM decoding which is to discover the hidden state sequence that was most likely to have produced a given observation sequence. Here, the Viterbi algorithm was used to find the best hidden state sequence composed of different symbols for an observation sequence (a given vehicle route). The process of a vehicle route prediction using the HMM and Viterbi algorithms is shown Figure 4.
Three algorithms were developed for the Timely Arrival Recommender System (TARS).

Algorithm 1: Context based Recommender System

**Input:** Traveler’s Position \((x_{gps}, y_{gps})\), D(destination) // X and Y are coordinates identified by the GPS.

**Output:** Recommended node.

**START**
1. \(C_n\) → Current node/Traveler’s Position \((x_{gps}, y_{gps})\)
2. \(D_n\) → Destination node,
3. \(db\): check_route\((C_n, D_n)\)
4. \(i = search(db) = 1_n\)
5. \(filter(i) \rightarrow treq.\)
6. \(j = search\) (nearest_node).
7. Push \(j \rightarrow recommendation\) (String) \(\rightarrow D_n.\)
8. Generate:- \([n_1, n_2, n_3, ... n_n]\)
9. Show optional node.

**End**

Algorithm 1 takes the GPS co-ordinate of the user from the user device and GPS co-ordinate of the vehicle from the database. The output of the Algorithm is the suggested route for the user based on his queries (current position and destination). Algorithm 1 recommends alternative nodes on the route for travelers to get to their destinations based on their current node (n) and destination. The algorithm initially collects the user’s location GPS coordinates (current node and destination), and then checks the database (db) for available nodes from current user location. It then provides the user with available nodes to his destination, details of the nodes (distance and time required); finds the nearest node from current user location to the user’s specified destination, and then recommends the nearest node from the current user location.

Algorithm 2: Estimation of Arrival Time

**Input:** Traveler’s Position \((x_{gps}, y_{gps})\), D(destination), GPS co-ordinates

**Output:** User’s estimated time of arrival.

**Start**
1. Initialize the Application
2. Collect the user GPS co-ordinate
3. Collect the vehicle GPS co-ordinate
4. Compute vehicle speed using GPS co-ordinates
5. Using speed calculate the distance
6. Using distance, calculate the time

**End**

Algorithm 2 is explained as follows:

- Initial stage is application initialization.
- After the application is initialized, the algorithm collects the user’s location based on GPS co-ordinates from the user’s device.
- The algorithm computes the speed of the vehicle using GPS coordinates.
• It calculates the distance.
• Time for arrival of the vehicle is estimated based on the distance.

Algorithm 3: Recommended Node Based on Current User Rating

Input: Traveler’s Position \((x_{gps}, y_{gps})\), \(D\) (destination), GPS co-ordinates
Output: Users’ rating of congested road segments (nodes).
Decision: Users’ rating of congested nodes.

Start
1. Initialize the Application
2. \(C_n\) → Current node/Traveler’s Position \((x_{gps}, y_{gps})\)
3. \(D_n\) → Destination node,
4. Obtain information on current users’ rating of the node
3. Take decision (based on users’ rating) whether to traverse the node or look for an alternative node
End

Algorithm 1 is used to generate recommended nodes leading to user destination based on the user query and context. Algorithm 2 plays a crucial role in TARS because it is used to generate estimated time of arrival for each of the available nodes and the recommended node. The user’s arrival time is computed on the basis of GPS coordinates with respect to the user’s current location. The timing computation is performed by first evaluating the vehicle’s speed (km/hr). The calculated speed is used to find the distance from the user’s current location to the user’s destination, which is received by the user through his android-based TARS application.

Algorithm 3 is used to predict road segments (nodes) that are traffic congested based on other current road users’ ratings. The road user can then make the decision of traversing other available suggested nodes that are leading to the destination and are traffic free.

Architecture of Timely Arrival Recommender System (TARS)

Figure 5 shows the architecture of the Collaborative Filtering Recommender System (TARS). The architecture of the TARS rating shows rating from current road users based on different time intervals. The system searches the current road user’s node with the use of GPS co-ordinate and the user will input destination node. The recommender system will then suggest nearest alternative nodes and recommend nodes that are free (based on user rating) that can lead to timely arrival at the user’s destination.

Figure 5. Architecture of the Collaborative Filtering Recommender System
The recommender system provides users with a list of options based on the user inputs. In the collaborative filtering (CF), if there is constraint on any route, there will be a signal (a rating) to coming users in the transcending nodes (vertices). For instance, when User1 in node A is transcending through node F and there is a constraint in the route (edge) from A→F, a signal (a rating) sent at that point in time leads User1 to change direction to a recommended route leading to his destination.

**Experimental Design**

Three experiments were conducted to determine the efficacy of the model. The first experiment which is known as Estimated Time of Arrival Test (ETAT) was used to calculate estimated time of arrival. The second experiment which is known as alternative route test (ART) was used to choose among alternative routes and nodes for a user to travel from the source to the destination. The third experiment, which is known as route congestion test (RCT) was used to determine segments of the road that were congested at different times of the day, leading to recommendations on the more feasible routes to follow.

**RESULTS**

Users are to sign-in when registration is completed and sign-up for registration. Figure 6 shows a user’s current location when during sign-up and the current location during sign-in.

The timely arrival recommender system (TARS) was tested using a modeled weighted road network graph showing two routes (R₁, R₂) and six nodes (A, B, C, D, E, F) as follows:

\[ R₂ = \langle A \rightarrow B \rightarrow D \rightarrow F \rangle \]
\[ R₁ = \langle A \rightarrow C \rightarrow E \rightarrow F \rangle \]

**Figure 6. User’s respective locations during sign-up and during sign-in**
The estimated time of arrival test (ETAT) was carried out on the three routes at different times of the day (7.30am-9.30am, 11am – 1pm, and 3.30pm-5pm). From the results, it was observed that during easy going hours (11am-1pm), estimated time of arrival (ETA) for each of the routes (R₁, R₂) was accurate with a time difference of ±30 seconds. This was possible because the nodes along the routes were not congested. But during rush hours (7.30am-9.30am and 3.30pm-5pm), estimated time of arrival test (ETA) was inaccurate with a time difference doubling or even tripling the actual estimated time of arrival. Also, it was observed in some cases that the shortest distance from source point to destination point turned out to be the longest due to traffic congestions on different nodes along the route.

The alternative route test (ART) was therefore used to generate alternative nodes on the route segments and recommend the node that the road user can possibly traverse to get to his destination on time. The application (TARS) achieves this by calculating the distance of each node from the users’ current position to the destination node, and then recommends the shortest node to the destination. Another problem the research was designed to solve is to predict road congestion so that road users can traverse through nodes that are traffic free. The reliability test result for road congestion (RCT) was dependable. Current ratings by road users that are stuck in congested road segments notified incoming users of the current traffic situation at the road segment where the rater was stocked.

From the analysis using the TARS, it was observed that the shortest route for timely arrival based on the test without any adverse traffic situation was Node A(0)→Node F(5)→Node G(6)→Node L(11)→Node K(10) as shown in Figure 7. The dotted line shows the recommended route.

The test was achieved by allowing three vehicles to traverse the three different routes (R₁, R₂, R₃). The time recorded for each of the three vehicles to traverse the routes at the same speed (30km/h) during easy going hours are as follows:

\[
\begin{align*}
R₁ & = < A(0) → F(5) → G(6) → L(11) → K(10)> = 2.8\text{km}, = 5.6\text{mim} \\
R₂ & = < A(0) → B(1) → E(4) → H(7) → K(10)> = 3.3\text{km}, = 6.6\text{mim} \\
R₃ & = < A(0) → B(1) → C(2) → D(3) → I(8) → J(9) → K(10)> = 4.8\text{km}, = 9.2\text{mim}
\end{align*}
\]

Following the above result, the shortest route for timely arrival at a speed of 30km/h is route one (R₁). But with adverse traffic situation on R₁, any of route two (R₂) and route three (R₃) can become the shortest path for timely arrival for road users to traverse from source to destination, thereby validating the need for and the efficiency of the timely arrival recommender system (TARS).

**Figure 7. Shortest Route as determined by TARS**
Integers 0 – 11 were used to denote nodes on the road network from source A(0) to destination node K(10). Each observation is defined by $p_{ij}$, where $i$ is the number of routes $R_i$ in the node set and $j$ is the number of coordinate points (nodes) in each route $R_i$. For instance, assume that the nodes (n) (A, B, C, D…K) are represented by integers (1, 2, 3, 4,…11), the observation sequence of the routes $R_1$, $R_2$, and $R_3$ is:

$R_3\rightarrow < A\rightarrow (0, 1), B\rightarrow (1, 2), C\rightarrow (2, 3), F\rightarrow (3, 8), I\rightarrow (8, 9), L\rightarrow (9, 10), and K\rightarrow >$

$R_2\rightarrow < A\rightarrow (0, 1), B\rightarrow (1, 4), E\rightarrow (4, 7), H\rightarrow (7, 10), and K\rightarrow >$

$R_1\rightarrow < A\rightarrow (0, 5), D\rightarrow (5, 6), G\rightarrow (6, 11), J\rightarrow (11, 10), and K\rightarrow >$

For each of the routes, the distance between nodes are:

$R_3\rightarrow < A\rightarrow (0.5\text{km}), \rightarrow(0.5\text{km}), \rightarrow(1.6\text{km}),\rightarrow(1.3\text{km}),\rightarrow(0.8\text{km}),\rightarrow(0.5\text{km})\text{and K\rightarrow >}$

$R_2\rightarrow < A\rightarrow (0.5\text{km}), \rightarrow(0.9\text{km}), \rightarrow(0.7\text{km}), \rightarrow(1.2\text{km}), \text{and K\rightarrow >}$

$R_1\rightarrow < A\rightarrow (0.8\text{km}), \rightarrow(0.4\text{km}), \rightarrow(1.1\text{km}), \rightarrow(0.5\text{km}), \text{and K\rightarrow >}$

For each of the routes ($R_i$), the distance between the observed given nodes ($R_1$, $R_2$, and $R_3$) is as follows:

$R_30\rightarrow1\rightarrow2\rightarrow3\rightarrow8\rightarrow9\rightarrow10 = \text{km} (0.5+0.5+1.4+1.1+0.8+0.5) = 4.8$

$R_20\rightarrow1\rightarrow4\rightarrow7\rightarrow10 = \text{km} (0.5+0.9+0.7+1.2) = 3.3$

$R_10\rightarrow5\rightarrow6\rightarrow11\rightarrow10 = \text{km} (0.8\text{km}+0.4\text{km}+1.1\text{km}+0.5\text{km}) = 2.8$

For each of the routes ($R_i$), we calculate the estimated time of arrival using the observed sequence of the route. Since the distance between each node is constant and already known, estimated time of arrival depends on the speed at which the vehicle is travelling. For instance, if a vehicle is travelling at a speed of 30km/h, the estimated time on each of the three given routes ($R_1$, $R_2$, and $R_3$) is as $D/S$, where $D$ denotes distance and $S$ denotes speed:

$R_30\rightarrow1\rightarrow2\rightarrow3\rightarrow8\rightarrow9\rightarrow10 = \text{hour} (0.017+0.017+0.047+0.037+0.027+0.017) = 0.162$

$R_20\rightarrow1\rightarrow4\rightarrow7\rightarrow10 = \text{hour} (0.017+0.03+0.023+0.04) = 0.11$

$R_10\rightarrow5\rightarrow6\rightarrow11\rightarrow10 = \text{hour} (0.027 + 0.013 + 0.037 + 0.017) = 0.093$

Then converting the time from hours to minutes, we have:

$R_30\rightarrow1\rightarrow2\rightarrow3\rightarrow8\rightarrow9\rightarrow10 = \text{min} (1.02 + 1.02 + 2.82 + 2.22+ 1.62+ 1.02) = 9.7$

$R_20\rightarrow1\rightarrow4\rightarrow7\rightarrow10 = \text{min} (1.02+ 1.8 + 1.38 + 2.4) = 6.6$

$R_10\rightarrow5\rightarrow6\rightarrow11\rightarrow10 = \text{min} (1.62 + 0.78 + 2.22 + 1.02) = 5.6$

From the results, it can be observed that the arrival time at 30km/h through route 1 ($R_1$) is 5.6 minutes; route 2 ($R_2$) is 6.6 minutes; while through route 3 ($R_3$) is 9.7 minutes. Thus, the route with the shortest distance for timely arrival is route 1 ($R_1$) which traverses through nodes $0\rightarrow5\rightarrow6\rightarrow11\rightarrow10$ at 30km/h in 0.093 hour (5.6 minutes). Table 1 shows the physical distance and time required to traverse from the source to the destination.

The graph showing the arrival time on each route at 30km/h ($R_i$ ($T_i$)) is illustrated in Figure 8.

To generate alternative nodes for a road user to traverse in case of adverse road situation, the road network model from source to destination points in Figure 9 was used as a test. Consider the road network from source point A(0) to destination point K(10).
It is assumed that any node that cannot be reached directly from a given node is hidden to the node in question. For instance, node (1) can be reached directly from the source node (0), but node (2) cannot be reached directly from (0). It therefore implies that node (2) is observable from node (1), while node (2) is hidden from node (0). Although the distance between any two nodes (d_{ij}) in the network of nodes is known and constant, the TARS application does not calculate the distance of a hidden (non-observable) node, rather it only calculates the distance of an observable or directly reachable node. The reason for this is to prevent any future error which may occur at hidden (non-observable) nodes due to occurrence of adverse traffic conditions that may lead to traffic congestion. Therefore, it is assumed

<table>
<thead>
<tr>
<th>Route</th>
<th>No. of Nodes (N)</th>
<th>Distance (km)</th>
<th>Speed (km/h)</th>
<th>Arrival time (min.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>R₁</td>
<td>5</td>
<td>2.8</td>
<td>30</td>
<td>5.6</td>
</tr>
<tr>
<td>R₂</td>
<td>5</td>
<td>3.3</td>
<td>30</td>
<td>6.6</td>
</tr>
<tr>
<td>R₃</td>
<td>7</td>
<td>4.8</td>
<td>30</td>
<td>9.7</td>
</tr>
</tbody>
</table>

Figure 8. Graph showing Arrival time on each Route at 30km/h (Rₜ)

Figure 9. Showing from source to destination for alternative route generation
that from the source node (0), the distance from node (1) to node (2) is not given, until when a road user travels from node (0) to node (1), that is when the distance from node (1) to node (2) is given.

The user inputs the following: (i) his current Position \((x_{gps}, y_{gps})\), (ii) destination \((D)\). The user is provided with real-time status of his vehicle based on his mobile phone’s GPS co-ordinates, since the application dynamically updates the location status of the user as he moves from one place to another.

The algorithm then generates two nodes on different routes that are nearest to the user’s current node on the user’s mobile device. For instance, assuming a user’s current location is node (4) and his destination is node (11) on our road network model, then node (4) automatically becomes the source node for the user, while node (11) is his destination. The application will generate only the nearest observable nodes (node 5 and node 7) that the user can traverse to get to his destination, and then suggest the node with the shortest distance to the next node as illustrated in Figure 10.

The screenshot of the output on the user’s mobile device (depicting the source, generated nodes and destination) is shown in Figure 11.

**Figure 10. Alternative nodes generated**

![Figure 10. Alternative nodes generated](image)

**Figure 11. Output of alternative nodes generated on a mobile device**

![Figure 11. Output of alternative nodes generated on a mobile device](image)
User ratings can be done on an entire route or on specific nodes. On an entire route, current users who traverse the three routes (R₁, R₂, R₃,) in our road network can rate each of the routes based on adverse traffic situation (Free Traffic Free [Ö] or Traffic Jam [x]). On request from a user whose currently location in our road network model is the source (node A(0)) getting to the destination point (node K(10), the output is indicated in Table 2.

Our result showed that from 13:22 (1:22pm) to 13:22 (1:22pm), road users rated route 1 (R₁) as having free traffic flow, while route 2 (R₂) and route 3 (R₃) were rated as having traffic jam. Six minutes later (13:26), route 1 (R₁) which was free, started experiencing traffic jam which lasted until 13:25 (1:35pm) when the road became free. At the time, route 2 (R₂) was rated as having free traffic flow, while the traffic jam on route 3 (R₃) lasted until 13:30 (1:30pm). From 13:35 (1:35pm) to 13:38 (1:38pm), route 1 (R₁) and route 3 (R₃) were having free traffic flow, while route 2 (R₂) was having traffic jam.

Recommendation is based on high rating of a route by current users at a specific time interval. The optimality of a route is based on user ratings at different time intervals.

On request from the road user (that is by clicking the recommend button), information on the current condition of the node was generated based on highest number of recent road users’ ratings. From the output in Figure 12, node (5) is blocked, while node (7) is free for the road users to get to their destination on time.

Table 2. Showing ratings by road users at different intervals

<table>
<thead>
<tr>
<th>Users</th>
<th>Route 1</th>
<th>Time</th>
<th>Users</th>
<th>Route 2</th>
<th>Time</th>
<th>Users</th>
<th>Route 3</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>User7</td>
<td>✓</td>
<td>13:26</td>
<td>User8</td>
<td>✓</td>
<td>13:26</td>
<td>User9</td>
<td>✓</td>
<td>13:27</td>
</tr>
</tbody>
</table>

Keys:
✓ = Free traffic flow
✗ = Traffic jam

Source: Field work sample in the study area. Recommendations generated by nine road users who rated and delivered recommendations to an active user looking for suggestions.

Figure 12. Output of recommended node generated on a mobile device
Table 3 shows the ratings or preferences of road users at different intervals. Following the results in Table 3, between 13:20 and 13:25, node 6 has free traffic flow based on the high rating by current users while node 5 is blocked.

Figure 13 shows the screenshot of recommended route and the ratings from various routes. Figure 14 shows the final destination reached by the user.

CONCLUSION

The outcome has shown that the Timely Arrival Recommender System (TARS) is able to generate nodes that lead to a user’s timely arrival at his destination from his source. Also, the application could differentiate between nodes that are blocked or congested and those that are free for easy traffic flow so that users can arrive their destinations on time.

<table>
<thead>
<tr>
<th>Users</th>
<th>Node 5</th>
<th>Time</th>
<th>Users</th>
<th>Node 6</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 5</td>
<td>✗</td>
<td>13:24</td>
<td>User 6</td>
<td>✓</td>
<td>13:25</td>
</tr>
<tr>
<td>User 3</td>
<td>✗</td>
<td>13:22</td>
<td>User 4</td>
<td>✓</td>
<td>13:23</td>
</tr>
<tr>
<td>User 1</td>
<td>✓</td>
<td>13:20</td>
<td>User 2</td>
<td>✗</td>
<td>13:20</td>
</tr>
</tbody>
</table>

Key:
 ✓ = Free traffic flow
 ✗ = Traffic jam

Figure 13. (a) Recommended Route; (b) Ratings from various routes
One of the interesting feature of the outcome is the ability of the system to calculate arrival time at a given speed (km/h), generate nearest nodes on each route, and recommend free traffic flow and blocked nodes based on users’ ratings. Another significant finding is the generation of time of arrival of the user. Most importantly, the system runs on Smartphones that are Android, Windows and iOS based, highlighting its high flexibility and supportability in real-time environments.

The fact that the results obtained from the research showed that the shortest observed route from source point to destination could be the longest hidden route when there are adverse traffic situations on the route justifies the need for a recommender system based on user ratings and related criteria.

It is suggested that since this application (TARS) was designed only for smartphones, future research should focus on additional platforms and also extend beyond road transportation networks.
REFERENCES


*Moses Adah Agana holds a Bachelor of Science Degree in Mathematics/Computer Science, a master’s degree in computer science and a Ph.D. in Computer Science (Cyber Security). He is an astute scholar who has been teaching, mentoring and supervising Computer Science students at various tertiary institutions since 1998, with several publications in both local and international journals. He has taught in the then Polytechnic Calabar (1998-1999), Federal College of Education Obudu (1999-2012), Federal University Lafia (2012), Federal University of Agriculture Makurdi (2012-2017), University of the Free State, South Africa, as a Postdoctoral Research Fellow (2017-2018), and in the University of Calabar (2017 till now). He has been a Head of Department at several instances and is the current Deputy Director, ICT in the University of Calabar. His research interest is predominantly in cyber security, Intelligent Systems and Internet of Things (IoT). Senior Lecturer and former Head of Department of Computer Science, University of Calabar Postgraduate student at the University of Calabar.*