Development of a Road Traffic Monitoring System

Olga Shpur
Department of Telecommunication
Lviv Polytehnic National University
Lviv, Ukraine
o.shpur@gmail.com

Yuliia Pyrih
Department of Telecommunication
Lviv Polytehnic National University
Lviv, Ukraine
yulia20122012@ukr.net

Taras Havryliv
Department of Telecommunication
Lviv Polytehnic National University
Lviv, Ukraine
taras.havryliv.mtr.2019@lpnu.ua

Nazar Peleh
Department of Telecommunication
Lviv Polytechnic National University
Lviv, Ukraine
van_plus_k@ukr.net

Oksana Urikova
Department of Finance
Lviv Polytehnic National University
Lviv, Ukraine
oksana.m.urikova@lpnu.ua

Andrii Branytskyi
Department of Telecommunication
Lviv Polytechnic National University
Lviv, Ukraine
andrii.v.branytskyi@lpnu.ua

***Abstract* – The problem of traffic forecasting in cities is considered. To improve the traffic situation in the city of Lviv, in particular to avoid traffic jams, a traffic traffic forecasting system is proposed, the architecture of which consists of a server using the Google API, a database, Amazon Forecast and a client. To create vehicle routes, the A * (A star) algorithm was chosen, which is more efficient for finding the best path in areas with a large number of obstacles, which is ideal for finding routes in cities. The obtained results demonstrate the effectiveness of the developed road traffic forecasting system, which indicates the feasibility of using the developed system to improve traffic in the city of Lviv.**

Keywords – traffic forecasting, route, Amazon Forecast, Heroku.

I. Introduction

In the city of Lviv in recent years there has been a significant increase in the number of road users, so the analysis of the distribution of road accidents, as well as visual display and prediction of road accidents to prevent and minimize tragic accidents and damage to health and property of road users movement is becoming increasingly important.

Today, the traffic safety system in Ukraine is based on means of accident prevention and prevention, although the modern development of information technology and scientific methods allows to act ahead - to predict [1], and accordingly, to prevent accidents on transport roads, purposefully exerting influence there , where the accident has not yet occurred, but there is a high probability. It is almost impossible to predict a specific accident due to the random nature of the accident, but researchers have determined that the aggregation of a large number of accidents over a sufficient area and time interval allows to establish a certain level of predictability, which can be described by mathematical and statistical relationships. (traffic intensity, weather conditions) [2, 3].

Therefore, the main problem of traffic safety is the lack of effective congestion forecasting systems for the optimal route, taking into account many criteria of road traffic. Thus, it is important to improve traffic management based on the use of modern infocommunication technologies.

II. The Problem of Forecasting Road Traffic in Urban Conditions

The problem of traffic forecasting is very extensive and includes some subtasks.

One of its most important features is the type of network where forecasting is performed, namely: urban networks and highways. The topology of such networks is radically different, as urban networks contain more short sections (sections of roads), while highways are much longer. As a rule, forecasting in cities is more difficult because the behavior of drivers is less predictable. Another key feature of this problem is the forecast horizon, which is divided into short-term (from 1 minute to approximately 30 minutes or 1 hour) and long-term (from 1 hour to 24 hours). As forecasting horizons (forecasting time) increase, the accuracy of forecasts increases. In addition to the type of networks and the forecasting horizon, the problem of traffic forecasting is determined by what variable and with what detail is forecast. Traffic forecasting can be performed for variables such as: traffic flow - the number of vehicles that pass through a certain area for a certain period of time (measured in vehicles / second or vehicles / hour); traffic density - the number of vehicles located in a certain area at the same time (measured in vehicles / meter or vehicles / kilometer); average speed - the average speed for vehicles on the site (measured in kilometers per hour or meter / second);travel time - the time for which the vehicle travels from the point of

2021 IEEE 16th International Conference on the Experience of Designing and Application of CAD Systems (CADSM) | 978-1-6654-3894-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/CADSM52681.2021.9385252

978-1-6654-3894-0/21/$31.00 ©2021 IEEE

departure to the destination (measured in seconds, minutes or hours).

To date, traffic data are generated in several ways:

• Surveys: traffic was measured using surveys conducted directly among the population. This data source has been replaced by other automated methods, which are cheaper.

• Sensors: devices that can detect some features of the road, such as the presence of the vehicle and its speed, etc.

• Cameras: real-time monitoring. This is a good way to reuse installed systems, but using camera recordings as traffic data requires software to convert images, such as traffic data;

• GPS-FCD: systems installed in modern cars allow you to find cars in real time with high accuracy. This type of data is the most desirable for traffic forecasting systems because it presents high quality data and no location restrictions.

The main problem is that sufficient transmission speed is required. Therefore, an important element of road safety is the presence of cameras and sensors, which are a source of information for the forecasting system about the current state of traffic.

III. Architecture of Road Traffic Monitoring System

To improve the road situation in the city of Lviv, in particular to avoid traffic jams, a traffic traffic forecasting system is proposed, the architecture of which is shown in Fig. 1. As you can see, its main elements are: a server using the Google API, a database, Amazon Forecast and a client.

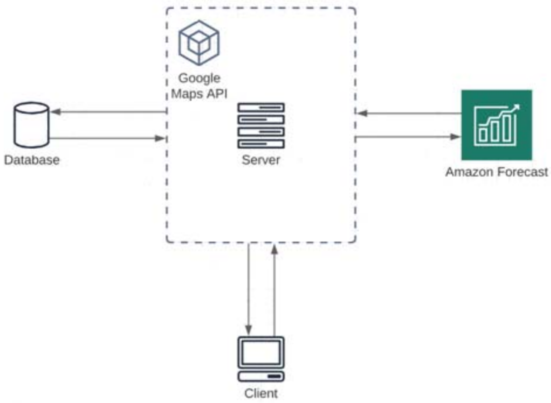

Fig.1. General architecture of traffic forecasting system

The server plays a key role for the entire system and interacts with all other elements of the system without exception [4]. For easy scaling of the system, it was decided to use the Google Maps API to obtain city maps with clearly defined road routes. The server receives requests to build routes, processes them and sends in response the best (with the lowest value of time) of the set of all possible. These queries are repeated at regular intervals to get the best route in real time. Also, information about current and possible places of congestion is sent to the client's side. Amazon's service, Amazon Forecast, processes data and forecasts. The Amazon Forecast API is used to communicate with the server. This service allows you to process a large set of data and create forecasts for the future, which will be taken into account by the server when building the next routes of customers. This process of communicating with the forecasting service will be continuous to improve the accuracy of future forecasts. All data received directly from this service will be sent to the database, where they will be stored and taken into account in subsequent requests.

Amazon Forecast is a fully managed service that uses machine learning to produce very accurate forecasts. This approach allows you to make accurate predictions for large data sets that have irregular trends. Amazon Forecast uses machine learning to combine time series data with additional variables to build forecasts, and does not require machine learning experience to get started. You only need to provide historical data, as well as any additional data that you think may affect your predictions. For example, the demand for a particular shirt color may vary depending on the season and the location of the store. This complex relationship is difficult to determine on its own, but machine learning is ideal for this case [5].

The Amazon Forecast service uses the Autoregressive Integrated Moving Average (ARIMA) machine learning algorithm, which is a commonly used statistical algorithm for forecasting time series. Once data is provided, Amazon Forecast automatically checks it and creates an appropriate forecasting model (Fig. 2).

The ARIMA algorithm is especially useful for simple data sets with time series less than 100. It is with these types of data that the designed system operates. Seasonal Autoregressive Integrated Moving Average (SARIMA) models aim to describe the current behavior of variables in terms of linear relationships with their characteristics.

This paper uses a paradigm to evaluate four non-seasonal and seasonal ARIMA algorithms: SWH2A, SWHSA, SWDP2A, SWDPSA. The SWH2A and SWHSA algorithms use hourly records in the learning window to calculate the parameters of the ARIMA and SARIMA hourly models. These models are used recursively to each hour to predict the hourly load. On the other hand, SWDP2A and SWDPSA use daily load records to calculate the parameters of the daily ARIMA and SARIMA models. In addition, hourly load records are used by SWDP2A and SWDPSA to calculate the 24-hour average daily hourly load profile. The daily ARIMA and SARIMA models are applied recursively to each day to forecast the total daily load, and then combined with the average 24-hour profile for forecasting for each hour.

To create vehicle routes, you must select a specific algorithm. Based on the analysis, the following most popular algorithms for route search are identified [6-9]: Dijkstri algorithm, Bellman-Ford algorithm, A * algorithm, Floyd-Warshall algorithm, Johnson algorithm, Lee algorithm (wave algorithm), Contraction hierarchies. After analyzing the features of their operation, the algorithm A* (A star) was chosen to search for routes in the developed system, because it is more effective for finding the best way in areas with many obstacles, which is ideal for finding routes in cities [10].

During the configuration of the developed system, the Spring Boot server was deployed to Heroku. The Spring Boot model for deploying standalone applications is perfect for Heroku. After successful registration, the Heroku CLI (Command Line Interface) for process control is installed on the Heroku service and authorization is performed. The next step is to create a server program based on the SpringBoot framework, for which the Spring Boot CLI was installed. By default, the program does not have its own logic - it is just an empty template that needs to be filled. Before you can deploy the application to Heroku, you need to create a Git repository for the program and add all the code to it [11].

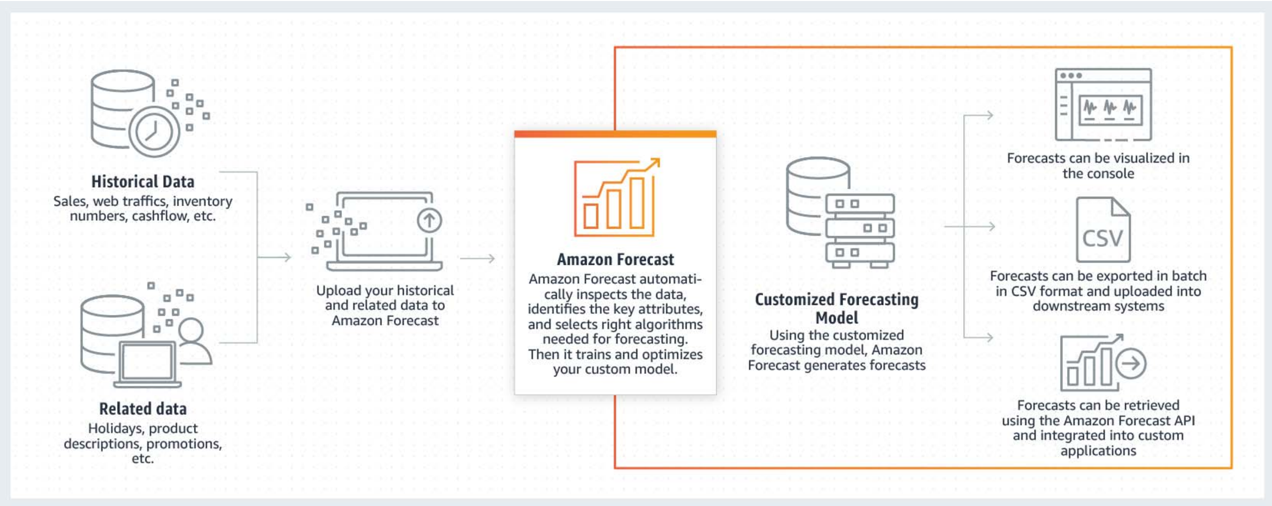

Fig.2. The general model of the Amazon Forecast service

After deploying the created server application, Heroku automatically identifies the program as a Maven / Java program due to the presence of the pom.xml file. The service has installed Java 8 by default, which can be easily configured using the system.properties file. It will launch the application using the default command. Thus, the server was successfully deployed. You can connect a PostgreSQL database to an existing server by running the following CLI command: *$ herokuaddons: createheroku-postgresql*.

You can now specify configuration variables for the server to display the URL required to connect to the database, DATABASE_URL.

Heroku also provides the pg command, which shows detailed information about the connected database:

$ heroku pg
=== DATABASE_URL
Plan: Hobby-dev
Status: Available
Connections: 0/20
PG Version: 12.2
Created: 2020-11-20 09:18 UTC
Data Size: 7.9 MB
Tables: 0
Rows: 0/10000 (In compliance)
Fork/Follow: Unsupported
Rollback: Unsupported
Continuous Protection: Off
Add-on: postgresql-animated-55555

This indicates that the database is running on Postgres 9.3.3 with a single row of data. When you create a database add-in, Heroku automatically populates the environment variables SPRING_DATASOURCE_URL, SPRING_DATASOURCE_USERNAME, and SPRING_DATASOURCE_PASSWORD. These environment variables allow SpringBoot to connect to the database without any other configuration.

IV. RESEARCH OF EFFICIENCY DEVELOPED SYSTEM

To demonstrate the operation of the designed system, the client side was presented in the form of a web page, which can then be replaced by more comfortable for end users applications, such as mobile application, web browser extensions, applications on various platforms (GPS navigators, etc.). The client has access to the geolocation of the end user who uses the system and provides the ability to form a route by specifying the starting point of the route (or use the current geolocation) and destination. Based on the actual traffic situation at the time of the route request and taking into account the forecasts of possible congestion - a route will be created that will correspond to the shortest travel time. To demonstrate the operation of the forecasting system, the initial set of historical data was set manually.

This route may change in real time during the driver's train, taking into account new circumstances. All created predictions will be checked to see if they were true or not. If the forecast is confirmed or erroneous, the forecasting system will get the result and based on it will learn to create more accurate forecasts in the future. The following condition was used to determine the congestion $7 \leq n \leq 10$ for $d = 30$, where n - the number of vehicles, d - the length of the study area, m. That is, it is assumed that with the number of vehicles from 7 to 10 on the road section of 30 meters a traffic jam is formed.

In fig. 3 shows the division of the road surface into sections for calculating the number of vehicles.

These virtual sections are formed in such a way that there are no problems when calculating congestion in oncoming lanes. Virtual rings are superimposed on each section, which are intended to delineate the area within which the number of vehicles will be calculated. Google Maps API provides the ability to calculate the number of markers in a particular neighborhood, in this case, the neighborhood is a ring, the marker - the number of cars.

To check the correct operation of the system, a traffic jam marker was artificially added on the section of the route road that was considered the most optimal. The system periodically checks whether the current route is still the most optimal, and in case of getting the best - offers it.

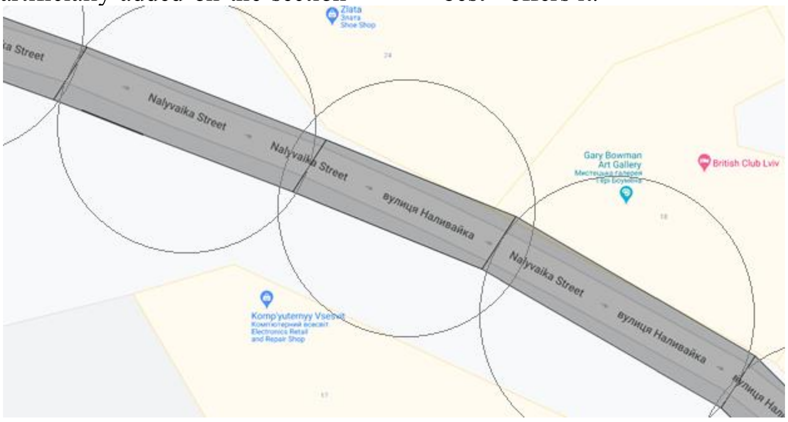

Fig. 3 Division of the road surface into sections for calculating the number of vehicles

In fig. 4 shows the change of the optimal route taking into account the new traffic data.

The red pointer in the figure indicates the presence of congestion at the time of laying the route, the blue line indicates the proposed route. It should be noted that the optimal route will be the one on which the vehicle will move for the shortest period of time.

Existing and predicted congestion are equally important for the system when selecting a route, ie the existing congestion at a certain point in time will be equated to a congestion that could not be noticed by the system, which is possible with the following requirements: the road section is not equipped with congestion sensors; vehicles do not have GPS trackers; vehicles cannot be recognized by navigation systems for any other reason.

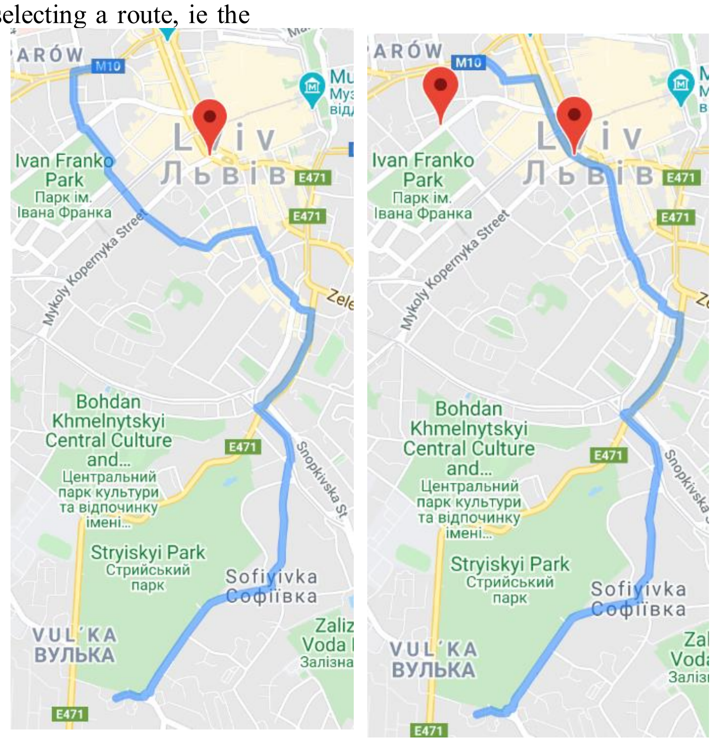

Fig. 4. Changing the optimal route taking into account new traffic data

In fig. 5 presents the results that were taken into account when choosing a route (in descending order of priority of route selection).

The optimal route that was built can be considered as follows: the first value is the number of traffic jams on the route (0), the second - the amount of time to overcome the route (14 min), and the last - the length of the route (4.4 km).

```
0: "0 • 14 min • 4.4 km"
1: "0 • 14 min • 4.8 km"
2: "1 • 15 min • 3.9 km"
0: "1 • 15 min • 3.9 km"
1: "1 • 17 min • 4.4 km"
2: "1 • 17 min • 4.8 km"
```

Fig. 5. The results of changing the optimal route taking into account the new traffic data (for two cases from Fig. 4)

From the figure 5 you can see that in the event of a traffic jam, all routes were given a value of 1 in the position of the number of traffic jams on the road. The route was rebuilt. In order to numerically present the benefits of using the designed traffic forecasting system, the system was tested on about 100 different routes and in different road situations (different number of traffic jams) and selected values with minimum and maximum time gain: t_1 - time to overcome the route without using the forecasting system; t_2 - time to overcome the route using the forecasting system; Δt - time difference for overcoming the route without and using the forecasting system in percent.

Let's calculate Δt at the minimum fixed gain in time: t_1 = 15 min; t_2 = 14 min:

$$\Delta t_{\min} = 100 - \frac{14 \cdot 100}{15} = 6{,}7\,\% \cdot \quad (1)$$

Let's calculate Δt at the maximum fixed gain in time: t_1 = 20 min; t_2 = 17 min:

$$\Delta t_{\max} = 100 - \frac{17 \cdot 100}{20} = 15\,\% \,. \quad (2)$$

The above calculations demonstrate the efficiency of using the developed system of road traffic forecasting, namely, the following gains in time - from 7 to 15% (average values).

In the histogram shown in Fig. 6 presents the time gain for overcoming a route laid without a forecasting system and with a forecasting system (10 routes were considered.

The obtained results show the expediency of using the designed system for vehicles in urban conditions.

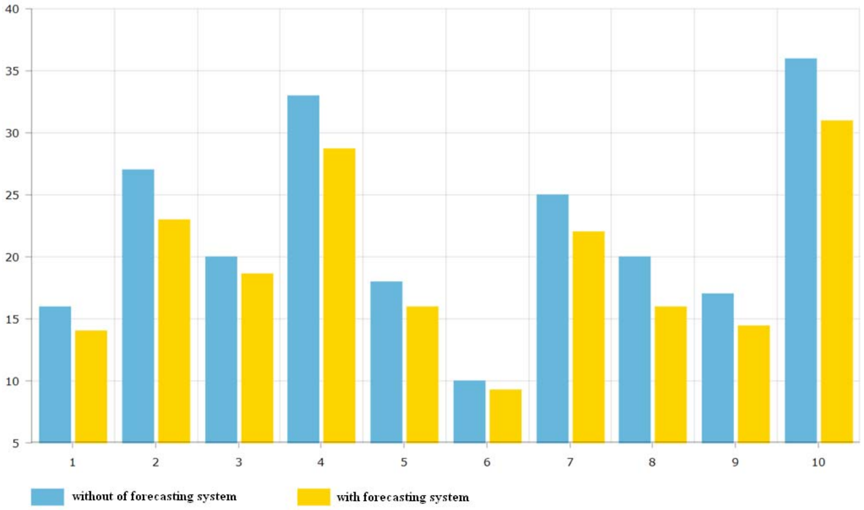

Fig. 6. Histogram of the time of overcoming route

V. Conclusion

The paper presents ways to obtain traffic information. To improve the traffic situation in the city of Lviv, in particular to avoid traffic jams, a traffic traffic forecasting system is proposed, the architecture of which consists of a server using the Google API, a database, Amazon Forecast and a client. To create vehicle routes, the A * (A star) algorithm was chosen, which is more efficient for finding the best path in areas with a large number of obstacles, which is ideal for finding routes in cities. Autoregressive Integrated Moving Average, a self-learning machine learning algorithm, was used for forecasting. The process of setting up the developed system is described and the efficiency of its paper is studied. The obtained results demonstrate the efficiency of using the developed road traffic forecasting system, namely, the gain during the trip - from 7 to 15%, which indicates the feasibility of using the developed system to improve traffic in the city.

References

[1] I. Lana et al., "Road traffic forecasting: Recent advances and new challenges", *IEEE Intelligent Transportation Systems Magazine,* 2018, 10(2), pp. 93-109.

[2] V.Yu. Pogrebnoy et al., "Primenenie geograficheskih informatsionnyih sistem v zadachah optimizatsii vyibora marshruta", *Nauchnyie issledovaniya i ih prakticheskoe primenenie,* 2014, s. 40–52.

[3] N.G. Markov, "Kombinirovannyiy algoritm prognozirovaniya dorozhnoy obstanovki na osnove metodov nechetkogo poiska", *Dokladyi Tomskogo universiteta,* 2013, s. 182–187.

[4] M. Klymash, et al., "A brief survey architecture of feedback systems for interactive E-Government ICT platforms", Proceedings *15th Int. Conf. TCSET*, 2020, pp. 458-461.

[5] Y. Lu et al., "Traffic flow prediction with big data: a deep learning approach", *IEEE Transactions on Intelligent Transportation Systems,* 2015, 16(2), pp. 865-873.

[6] A.I. Pavlenko, "Rozv'yazannya zalezhnih vid chasu zadach poshuku naykorotshogo shlyahu", *Zhurnal obchislyuvalnoyi ta prikladnoyi matematiki*, 2015, s. 24-37.

[7] M. Klymash et al.,"A modified genetic algorithm for increasing the efficiency of routing in self-organized networks", *2019 IEEE Int. Conf. PIC S and T,* 2019, pp. 447-450.

[8] Y. Pyrih et al.,"A Modified Simulated Annealing Algorithm Based on Principle of the Greedy Algorithm for Networks with Mobile Nodes", *Proceedings 15th Int. Conf. TCSET 2020,* 2020, pp. 482-485.

[9] B. Strykhalyuk et al., "Synthesis of distributed service-oriented structures cloud networks is based on algorithm for determining hyperbolic virtual coordinates", *in Proc. of XIIIth Int. Conf. CADSM 2015,* 2015, pp. 231-235.

[10] A. Goldberg "Computing the shortest path:A* search meets graph theory",*Technical Report, Microsoft Research*, 2004, 25 p.

[11] B.L. Smith, **"**Comparison of parametric and nonparametric models for traffic flow forecasting**"**, *Transportation Research Part C*, 2002, pp. 303-321.