

**FEVZİ YASİN KABABULUT**, M.Sc.<sup>1</sup>  
(Corresponding author)  
E-Mail: fyasın.kababulut@udhb.gov.tr

**DAMLA KUNTALP**, Ph.D.<sup>2</sup>  
E-Mail: damla.kuntalp@deu.edu.tr

**OLCAY AKAY**, Ph.D.<sup>2</sup>  
E-Mail: olcay.akay@deu.edu.tr

**TİMUR DÜZENLİ**, Ph.D.<sup>3</sup>  
E-Mail: timur.duzenli@amasya.edu.tr

<sup>1</sup> Ministry of Transport and Infrastructure  
IV. Regional Directorate Çeşmebaşı Neighborhood  
Alpaslan Avenue No: 2, Ovaakça-Osmangazi, Bursa,  
Turkey

<sup>2</sup> Department of Electrical and Electronics Engineering  
Dokuz Eylul University  
Tınaztepe Campus, Buca, Izmir 35100, Turkey

<sup>3</sup> Department of Electrical and Electronics Engineering  
Amasya University, Technology Faculty  
A-Block, Room: A303, Amasya 05100, Turkey

Traffic on Motorways  
Preliminary Communication  
Submitted: 2 Oct. 2017  
Accepted: 9 July 2018

# SIMPLE AND EFFICIENT PREDICTION OF NEAR FUTURE STATE OF TRAFFIC USING ONLY PAST SPEED INFORMATION

## ABSTRACT

*Intelligent traffic systems attempt to solve the problem of traffic congestion, which is one of the most important environmental and economic issues of urban life. In this study, we approach this problem via prediction of traffic status using past average traveler speed (ATS). Five different algorithms are proposed for predicting the traffic status. They are applied to real data provided by the Traffic Control Center of Istanbul Metropolitan Municipality. Algorithm 1 predicts future ATS on a highway section based on the past speed information obtained from the same road section. The other proposed algorithms, Algorithms 2 through 5, predict the traffic status as fluent, moderately congested, or congested, again using past traffic state information for the same road segment. Here, traffic states are assigned according to predetermined intervals of ATS values. In the proposed algorithms, ATS values belonging to past five consecutive 10-minute time intervals are used as input data. Performances of the proposed algorithms are evaluated in terms of root mean square error (RMSE), sample accuracy, balanced accuracy, and processing time. Although the proposed algorithms are relatively simple and require only past speed values, they provide fairly reliable results with noticeably low prediction errors.*

## KEY WORDS

*ATS prediction; vehicle traffic; prediction of traffic status;*

## 1. INTRODUCTION

The foundations of traffic engineering were laid out by the traffic flow model of Lighthill-Whitham [1] and Richards [2] in the mid-1950s. Many studies on traffic parameter estimation are based on this model.

Yang [3] developed a traffic state estimation algorithm based on a specific stochastic traffic flow model which takes into consideration many random factors related to time and space, such as driver behavior, weather, and road conditions. These random events render predicting traffic parameters difficult. On the other hand, methods such as the Unscented Kalman Filter and Particle Filter, which were used in [3], make estimating traffic parameters possible. Yang proposed a stochastic macroscopic flow model that focused on such traffic parameters as density and volume. An application for dynamic pricing of toll lanes was suggested using this model [3].

In a study similar to Yang's work, Qui [4] also performed macroscopic traffic state estimations using traffic information from wireless networks. In that study, Qui was interested in speed and density as traffic parameters. While the standard Kalman Filter and Particle Filter methods were used for speed estimation, the Extended Kalman Filter method was reserved for density estimation. When it comes to speed estimation, there is no significant difference between standard Kalman Filter and Particle Filter. As for density estimation, the loop detector system was compared against Extended Kalman Filter based on real-time cell phone traffic data [4]. Results showed that the performance of the Extended Kalman Filter was not noticeably better than that of the loop detector system. However, the Extended Kalman Filter could still increase the accuracy of speed estimation at times [4].

The aim of another study [5], which estimated traffic parameters using the Extended Kalman Filter, was detecting traffic incidents. Congestion, accidents,

delays, roadworks, etc. can adversely affect the traffic flow and cause fatal accidents, material damages, and significant gas emissions. Real-time traffic incident management systems in that study aimed to reduce such incidents and to prevent occurrences of secondary events by informing the drivers [5].

In [6], traffic density was estimated by the Switching Mode Model (SMM), which is a macroscopic traffic flow model produced using the Cell Transmission Model. The Cell Transmission Model simulates a highway by dividing it into cells which are composed of equal distances traveled by vehicles at a given time interval [7, 8]. SMM is a hybrid method that switches between linear differential equations (or modes) by taking into account the congestion status of separated cells and other traffic parameters at the input-output boundaries in a highway [6].

Another study was focused on Dynamic Traffic Assignments (DTA) that evolved to the Cell Transmission Model in intelligent traffic systems [9]. DTA systems are traffic models that classify driver behaviors and guide them by giving information about dynamic equilibrium conditions including routes that are tracked for minimizing travel time and travel costs of drivers. Estimation of traffic parameters is highly important for those models. A new framework for those models was suggested in [9].

In [10], speed prediction was performed by using the Decision Tree Regressor (DTR) and K-Nearest Neighbor (KNN) algorithms employing traffic data measured by 122 sensors in a small part of the D100 state highway in Istanbul. The authors predicted the speed 30 minutes into the future on that highway section with the help of the current speed, along with the speed values 15 and 30 minutes into the past. They used measured speed values, day of the week, time interval, and weather conditions as features for

estimating the speed. Their results showed that information provided by the feature of weather condition did not improve the accuracy of their estimations [10].

As can be seen from the studies mentioned above, being the subject of many disciplines of traffic engineering, estimation of traffic parameters is a complex problem to be solved. Different from the methods used in those studies, in this work we propose simple yet efficient algorithms which use only the average speed data of the near past for predicting near future average traveler speed (ATS) (Algorithm 1) and traffic status (Algorithms 2 through 5).

The rest of the paper is organized as follows. In Section 2, we explain the employed traffic dataset in detail. In Section 3, we introduce five different algorithms that we propose. We evaluate the performances of the proposed methods employing several criteria and compare them in Section 4. In the Conclusions section, we discuss our results.

## 2. DATASET DESCRIPTION

The dataset used in this study was provided by the Traffic Control Center of Istanbul Metropolitan Municipality. It includes values of multiple traffic parameters belonging to a busy highway section (D100 Çağlayan) in Istanbul (Figure 1).

D100 Çağlayan in the Çağlayan district of Istanbul is a short section of D100 state highway which starts from the Turkish border with Bulgaria and ends at the Iranian border having a total length of 1,788 km. D100 Çağlayan is one of the busiest highway sections in Istanbul, especially during rush hours. The traffic data for D100 Çağlayan includes the total number of vehicles in the lanes, average speed, and average occupancy labeled by time intervals of 10 minutes. Exemplary portions of the original 2013 and 2016 data are presented in Tables 1 and 2, respectively. Our

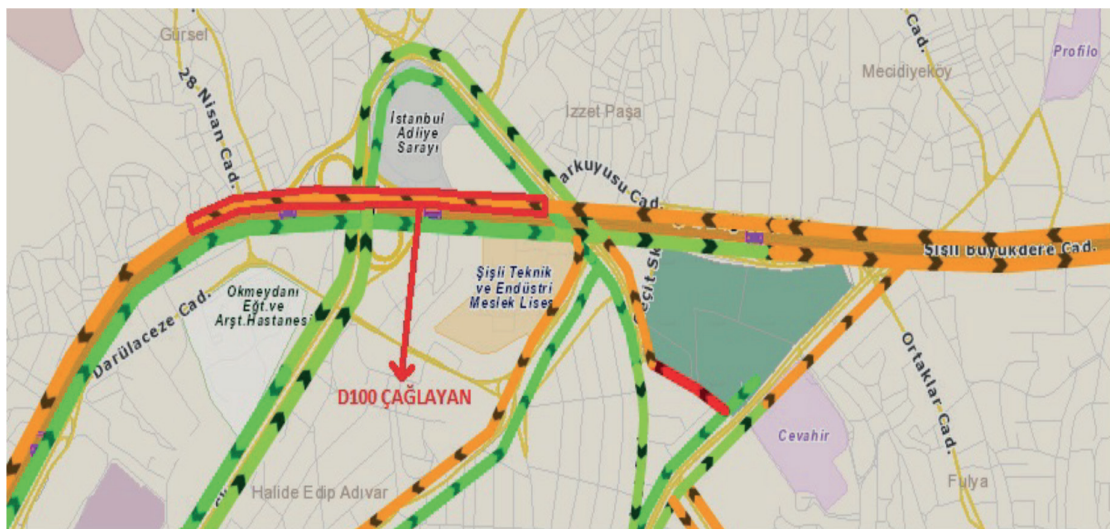


Figure 1 – Location of the highway section “D100 Çağlayan”

Table 1 – An exemplary portion of dataset for 2013

Sensor number	Total # of vehicles on lane 1	Total # of vehicles on lane 2	Total # of vehicles on lane 3	Average speed	Average occupancy	Time
363	5	116	100	95	1	01.10.2013 00:00
363	10	113	100	94	2	01.10.2013 00:10
363	10	120	90	93	1	01.10.2013 00:20
363	23	88	78	87	2	01.10.2013 00:30
363	23	100	85	86	1	01.10.2013 00:40
363	19	102	59	86	1	01.10.2013 00:50
363	17	85	39	87	1	01.10.2013 01:00
363	16	83	38	90	0	01.10.2013 01:10
363	7	71	21	88	0	01.10.2013 01:20
363	14	76	36	89	0	01.10.2013 01:30
363	14	68	30	87	0	01.10.2013 01:40
363	9	56	14	86	0	01.10.2013 01:50
363	13	53	25	86	0	01.10.2013 02:00
363	8	51	13	83	0	01.10.2013 02:10
363	9	61	16	88	0	01.10.2013 02:20
363	18	58	18	87	0	01.10.2013 02:30
363	11	49	9	89	0	01.10.2013 02:40
363	9	38	12	90	0	01.10.2013 02:50
363	7	19	7	85	0	01.10.2013 03:00
363	9	34	6	85	0	01.10.2013 03:10

Table 2 – An exemplary portion of dataset for 2016

Sensor number	Total # of vehicles on lane 1	Total # of vehicles on lane 2	Total # of vehicles on lane 3	Average speed	Average occupancy	Month	Day	Hour	Minute
363	264	315	308	73	7	1	4	12	0
363	319	391	377	72	7	1	4	12	10
363	332	363	390	72	7	1	4	12	20
363	326	338	380	"69"	8	1	4	12	30
363	340	352	387	71	7	1	4	12	40
363	263	280	309	70	8	1	4	12	50
363	195	215	230	70	10	1	4	13	0
363	312	350	385	"69"	9	1	4	13	10
363	318	359	388	70	9	1	4	13	20
363	315	372	385	71	8	1	4	13	30
363	317	380	391	72	7	1	4	13	40
363	320	390	386	72	7	1	4	13	50
363	316	366	387	71	7	1	4	14	0
363	297	340	379	67	10	1	4	14	10
363	291	315	"378"	"65"	11	1	4	14	20
363	290	308	379	64	9	1	4	14	30
363	302	326	392	67	9	1	4	14	40
363	315	328	381	67	9	1	4	14	50
363	312	328	381	67	10	1	4	15	0
363	245	272	300	68	9	1	4	15	10



traffic data was collected by radar-based Smart Microwave Sensors. These radar-based sensors are traffic measurement detectors specifically designed for traffic detection applications. They measure the distance to the objects entering the microwave radiation field. They obtain cross-sectional traffic flow information (occupancy, speed, etc.) at points where they are located. They are positioned along the road and mounted on poles planted at regular intervals of 1 km.

There are many travel-aid applications for drivers which are based on short-term prediction of traveling speed. Istanbul Metropolitan Municipality, like many other metropolitan municipalities in the world, offers a mobile phone application for drivers indicating the average speed value on roads and provides traffic status maps (Figure 2). Hence, fast, efficient, and reliable short-term prediction of average traveling speed or traffic status on metropolitan roads is crucial for such applications. It is also valuable and necessary for the development of adaptive traffic control and traffic management systems.

In this study, we used the average speed data on D100 Caglayan for the last 3 months of 2013 and the first 5 months of 2016. The reason for using traffic data of different years on the same highway section is to test the robustness of our methods. We expect that despite an increase in the number of vehicles, the traffic flow on a highway section should have similar characteristics over the years due to investment and improvements in highway infrastructure.

The average speed data is formed by taking the average of speeds of all the vehicles passing through that highway section. Vehicle speeds are automatically measured and the average is calculated every 10 minutes. However, sometimes erroneous measurements

Table 3 – Binary and ternary labeled speed data

Speed interval [km/h]	Binary states	Ternary states
51-121	0	0
30-50	1	1
0-29	1	2

are possible due to sensor, road, or communication failures or even no measurements at some instances. In those cases, the average speed data appears with a “null” notation indicating that it is faulty. After the faulty data samples were extracted, the remaining data was used in our study.

In Algorithm 1 below, the average speed data is used in its raw form. For the other four proposed algorithms, the average speed data is first converted into binary or ternary states representing the traffic status before further processing.

In the binary form, the average speed data is labeled by “0” to represent “fluent” traffic status if it is higher than 50 km/h and by “1” to represent “congested” traffic status if it is lower than 50 km/h. In the ternary form, the average speed data is labeled by “0” (fluent) if it is higher than 50 km/h, by “1” (moderately congested) if it is between 30 km/h and 50 km/h, and by “2” (congested) if it is lower than 30 km/h [11]. These labels are indicated in Table 3. Here, speed intervals are assigned as determined by the Traffic Control Center of Istanbul Metropolitan Municipality.

### 3. PROPOSED ALGORITHMS

In all the algorithms, we construct a set of vectors through a history window containing five past data values toward predicting the future status of traffic for the

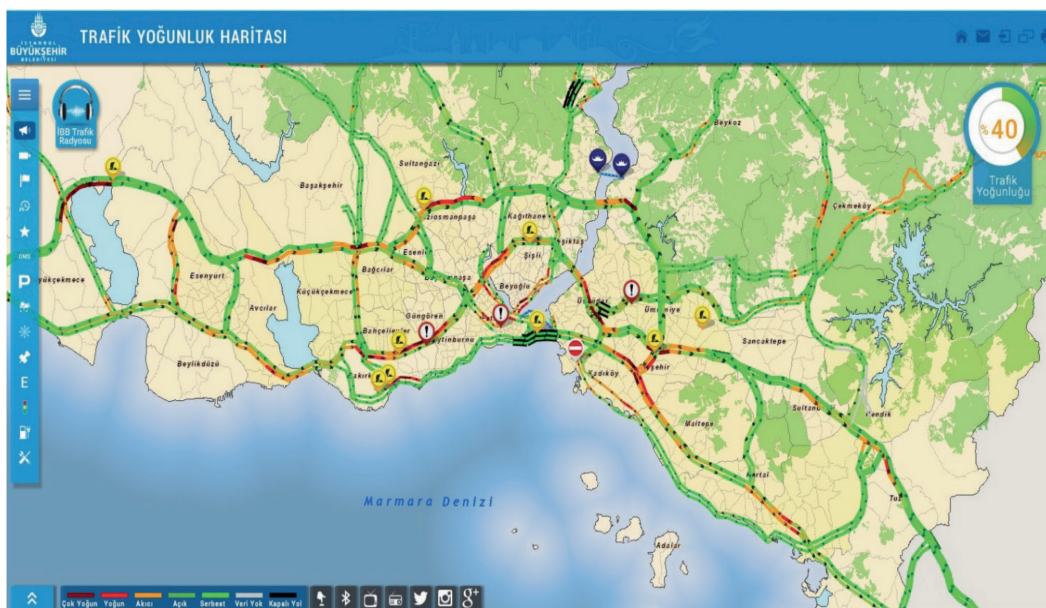


Figure 2 – Traffic Status Map of Istanbul

next 10-minute interval. In *Figure 3*, a prediction interval containing the history window and prediction slot is illustrated.

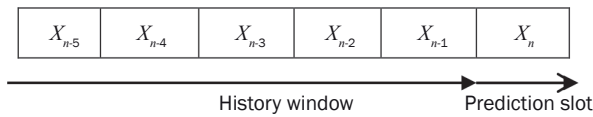


Figure 3 – Prediction interval

### 3.1 Algorithm 1

Algorithm 1 aims to find the tendency (increase or decrease) of the average speed data by looking at the differences between the successive speed values in the history window. By adding the mean value of those differences to the last speed data value, the average speed in the next 10-minute interval is predicted.

In *Figure 3*,  $X$  values in the history window slots represent the average speeds in the five past successive 10-minute intervals.  $X_n$  value in the prediction slot corresponds to the predicted average speed in the next 10-minute long interval. We calculate the differences of speed values as

$$D_i = X_{n+i-6} - X_{n+i-5}, \quad i = 1, 2, 3, 4 \quad (1)$$

By taking the average of the differences of speed values, the mean of the differences,  $M$ , can be calculated as

$$M = \frac{1}{4} \sum_{i=1}^4 D_i \quad (2)$$

Based on the value of  $M$ , the predicted speed,  $X_n$ , is determined as follows:

$$X_n = X_{n-1} + M \quad (3)$$

where  $X_{n-1}$  is the actual speed value in the last slot of the history window (see *Figure 3*).

For the next four proposed algorithms,  $X$  values in *Figure 3* correspond to binary or ternary traffic status labels (as opposed to actual average speed values) in the five past 10-minute intervals.

### 3.2 Algorithm 2 and Algorithm 3

In Algorithms 2 and 3 we use both binary and ternary labeled status data in the prediction interval. The nine parameters used in Algorithms 2 and 3 are introduced below.

Zero-one rate (ZO): Ratio of the number of 0-to-1 transitions in the history window to window length.

One-zero rate (OZ): Ratio of the number of 1-to-0 transitions in the history window to window length.

Zero-two rate (ZT): Ratio of the number of 0-to-2 transitions in the history window to window length.

Two-zero rate (TZ): Ratio of the number of 2-to-0 transitions in the history window to window length.

One-two rate (OT): Ratio of the number of 1-to-2 transitions in the history window to window length.

Two-one rate (TO): Ratio of the number of 2-to-1 transitions in the history window to window length.

Last 2 slot (L2S): Index of the last history window slot containing the value of 2.

Last 1 slot (L1S): Index of the last history window slot containing the value of 1.

Last 0 slot (LOS): Index of the last history window slot containing the value of 0.

*Algorithm 2:* In the binary version of Algorithm 2, if L1S is greater than LOS and the difference between them is less than the average duration of busy state  $1/ZO$  [12], the decision is given as State 1. Otherwise, the decision is given as State 0. On the contrary, if LOS is greater than L1S, and the difference between them is less than the average duration of idle state  $(1/OZ)$  [12], the decision is given as State 0. Otherwise, the decision is given as State 1. A similar logic applies for the ternary version. The flowchart of Algorithm 2 can be seen in *Figure 4* where PTS stands for “Predicted Traffic State”. In the flowchart, the additions for the ternary version are written in grey font, as opposed to blocks in black font belonging to the binary version.

*Algorithm 3:* The probability of predicting the congested state (State 2) correctly by Algorithm 2 is rather low for the case of ternary labeling. Hence, Algorithm 2 has been modified as Algorithm 3. In Algorithm 3, if there exist at least three identical data samples among the last five slots, the prediction is made based on the majority rule. For example, if the last five samples are 0, 1, 0, 1, 1, then the predicted state becomes 1. In the ternary version, if the number of occurrences (NO) of two states within the five past data values is equal, the predicted state is assigned as the state of the last slot (LS) of the history window. The flowchart of Algorithm 3 is given in *Figure 5*, where the black- and grey-colored blocks belong to the binary and ternary versions of the algorithm, respectively.

### 3.3 Algorithm 4 and Algorithm 5

Algorithms 4 and 5 are based on a Markov chain model as sketched in *Figure 6*. In Algorithms 4 and 5, prediction is performed by calculating the state transition probabilities in the history window and the probabilities of observing States 0 and 1. Those probabilities are obtained by analyzing the data values in the history window. After that, the value of the prediction slot is determined based on those probabilities.

The following probabilities are used in Algorithms 4 and 5:

$$P_0 = \frac{\# \text{ of } 0 \text{ in history window}}{\text{history window size}}$$

$$P_1 = \frac{\# \text{ of } 1 \text{ in history window}}{\text{history window size}}$$

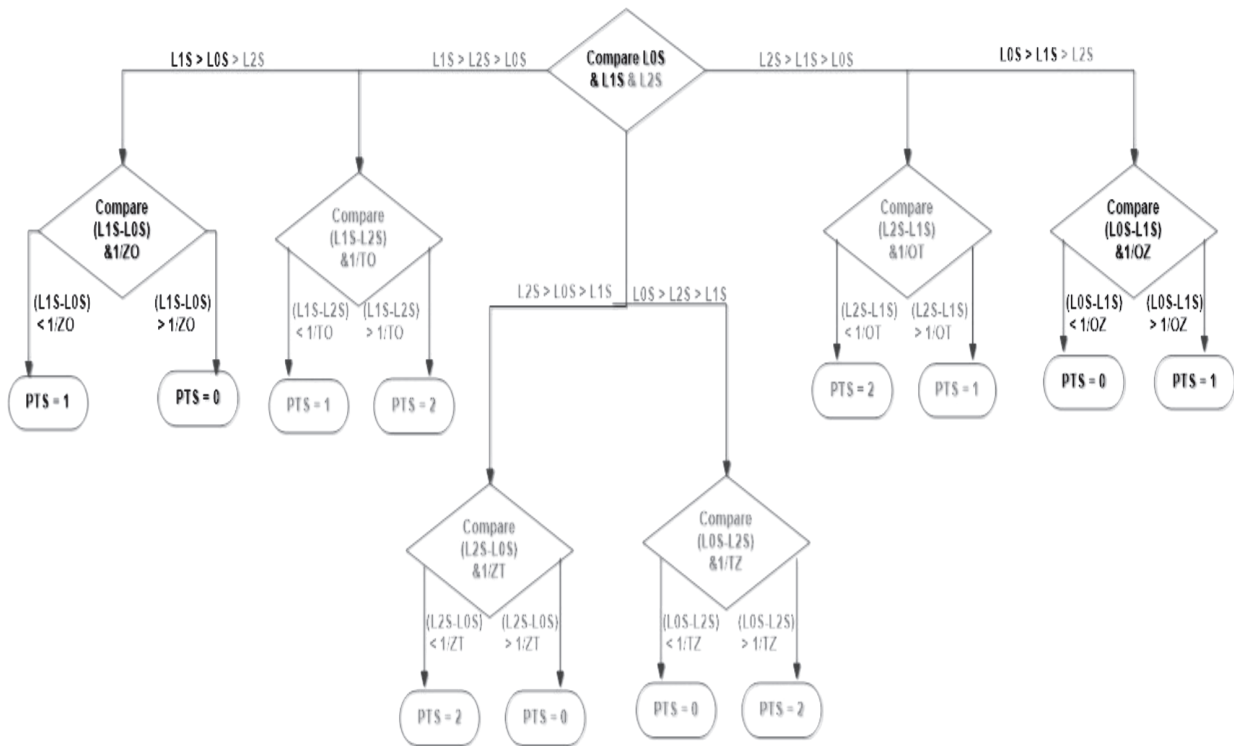


Figure 4 – Flowchart of Algorithm 2 (black font: binary; grey font: ternary)

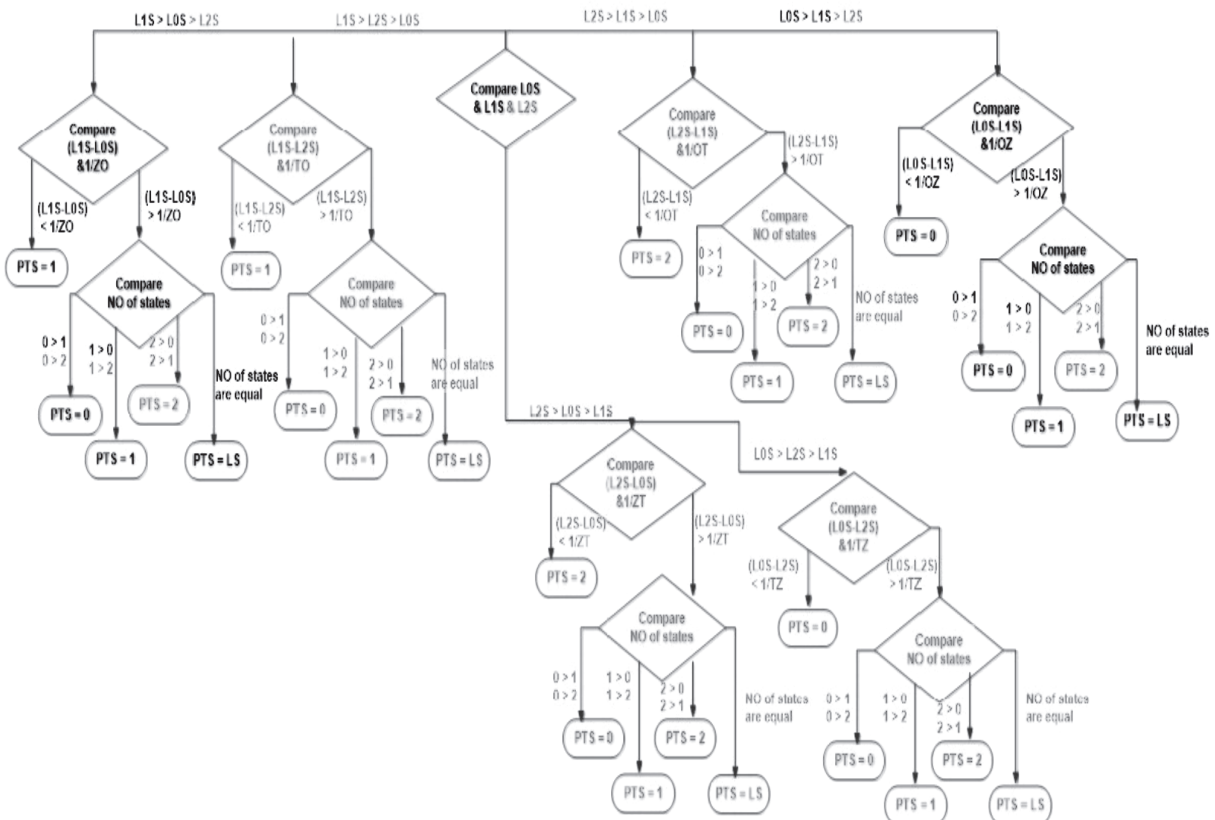


Figure 5 – Flowchart of Algorithm 3 (black font: binary; grey font: ternary)

$$P_2 = \frac{\# \text{ of } 2 \text{ in history window}}{\text{history window size}}$$

$$P_{ij} = \frac{\# \text{ of transitions from State } j \text{ to State } i \text{ in history window}}{\text{history window size} - 1}$$

$$i = 0, 1, 2 \text{ and } j = 0, 1, 2$$

$P(0)$  = Probability of observing State 0 in the prediction slot

$P(1)$  = Probability of observing State 1 in the prediction slot

$P(2)$  = Probability of observing State 2 in the prediction slot

$P(0)$  and  $P(1)$  for the binary version are calculated as

$$P(0) = (P_{00} \times P_0) + (P_{01} \times P_1) \quad (4)$$

$$P(1) = (P_{10} \times P_0) + (P_{11} \times P_1) \quad (5)$$

$P(0)$ ,  $P(1)$ , and  $P(2)$  for the ternary version are calculated as

$$P(0) = (P_{00} \times P_0) + (P_{01} \times P_1) + (P_{02} \times P_2) \quad (6)$$

$$P(1) = (P_{10} \times P_0) + (P_{11} \times P_1) + (P_{12} \times P_2) \quad (7)$$

$$P(2) = (P_{20} \times P_0) + (P_{21} \times P_1) + (P_{22} \times P_2) \quad (8)$$

**Algorithm 4:** In Algorithm 4, a decision is made by comparing  $P(0)$  and  $P(1)$ , which are defined above. Whichever is higher, the prediction slot is assigned as that state. If  $P(0)$  and  $P(1)$  are equal, we compare the number of occurrences of each state in the history window. The state with a higher number of occurrences (NO in Figures 7 and 8) is assigned as the predicted state. In the ternary version, if the number of occurrences of two states within five past data values is equal, the predicted state is assigned as the state occurring in the last slot (LS in Figures 7 and 8) of the history window. A flowchart of Algorithm 4 is given in Figure 7 where blocks belonging to binary and ternary versions are indicated by black- and grey-colored fonts, respectively.

**Algorithm 5:** In Algorithm 5, unlike Algorithm 4, the final decision is based on the transition probabilities only. The predicted state is assigned as the one with the highest transition probability, as explained below.

If the state of the last slot of the history window is 0, the probability of transition from State 0 to State 0 is compared against the probability of transition from State 0 to State 1 (additionally, the probability of transition from State 0 to State 2 in the ternary version). For example, if the probability of transition from State 0 to State 1 is higher, the prediction slot is assigned as 1. If the transition probabilities are equal, the same process as in Algorithm 4 is applied. The flowchart of Algorithm 5 can be seen in Figure 8, where the operations and blocks of binary and ternary versions are labeled by black- and grey-colored fonts, respectively.

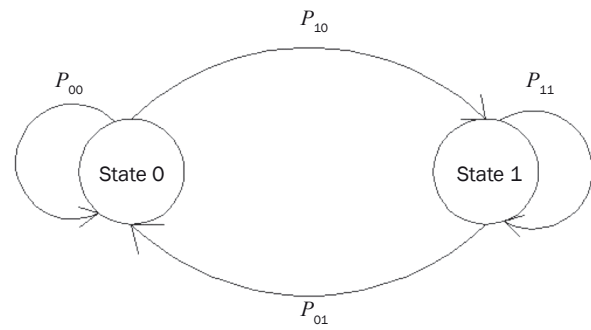


Figure 6 – A two-state Markov chain

#### 4. PERFORMANCE EVALUATIONS

We used sample accuracy ( $Acc$ ), balanced accuracy ( $Bac$ ), and root mean square error ( $RMSE$ ) for performance evaluation of our algorithms.

$Acc$  is one of the most commonly employed performance criteria for problems addressed in this paper. This criterion is calculated as the ratio of the total number of correct decisions to the total number of trials [13],

$$Acc = \frac{k_{cor}}{N} \quad (9)$$

where  $k_{cor}$  is the total number of correct decisions and  $N$  represents the total number of trials.

$Bac$  is another criterion that allows all classes to participate equally in the performance evaluation of classification problems. It is calculated as

$$Bac = \frac{1}{L} \sum_{i=1}^L \frac{k_i}{N_i} \quad (10)$$

where  $L$  denotes the total number of classes,  $k_i$  is the number of correct decisions for class  $i$ , and  $N_i$  represents the total number of trials for that class.

We used  $RMSE$  to measure the error in estimating speed by Algorithm 1.  $RMSE$  is calculated as

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (x_i - \hat{x}_i)^2}{N}} \quad (11)$$

In Equation 11,  $x_i$  represents the measured actual speed,  $\hat{x}_i$  denotes the predicted speed, and  $N$  represents the total number of trials.

In Algorithm 1, we first predict the ATS using only the average past speed data. In order to make comparisons with other algorithms we proposed, we convert those predicted ATS values into traffic status labels in accordance with the threshold values determined by the Traffic Control Center of Istanbul Metropolitan Municipality, as explained in Section 2. For the other four proposed algorithms, we first convert the speed data into traffic status labels (0, 1, and 2) via that conversion process. The predicted traffic status labels obtained by these algorithms are based on those converted labels. In the binary case, we classify the traffic status as fluent (0) or congested (1), while in the ternary case we classify it as fluent (0), moderately



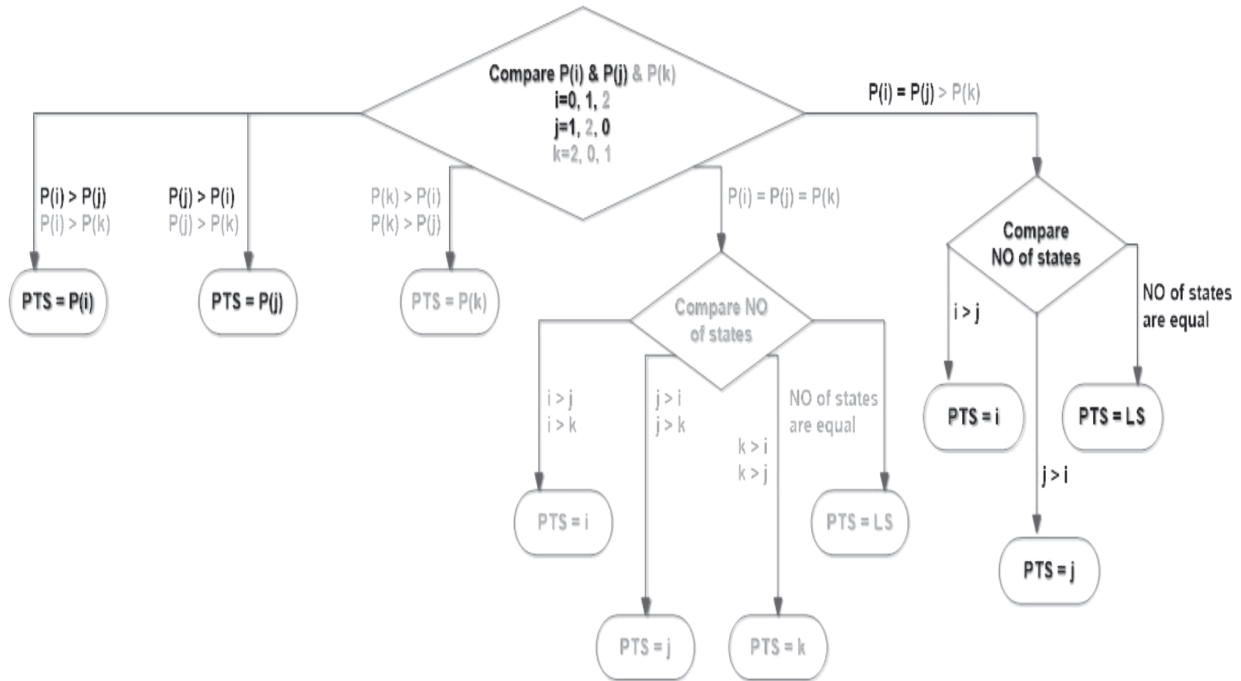


Figure 7 – Flowchart of Algorithm 4 (black font: binary; grey font: ternary)

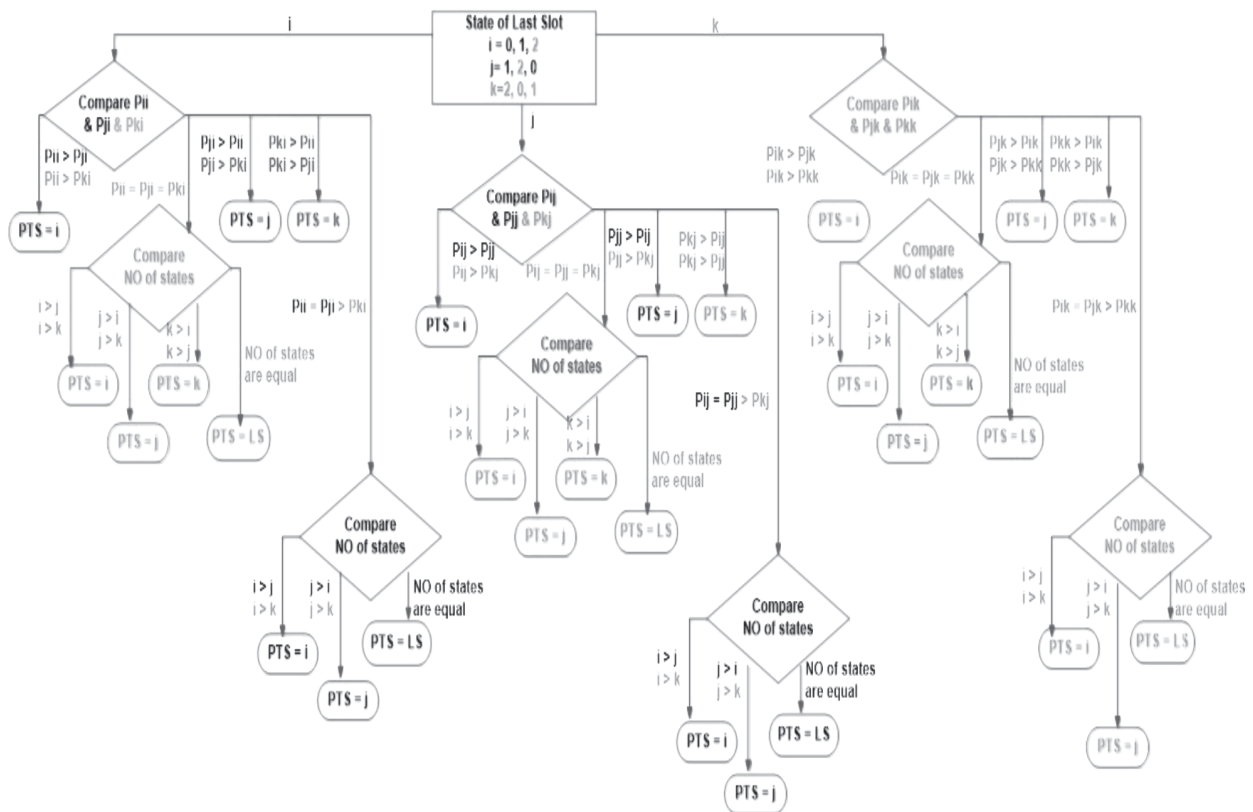


Figure 8 – Flowchart of Algorithm 5 (black font: binary; grey font: ternary)

congested (1), or congested (2). Comparisons of the algorithms presented for the binary and ternary cases in terms of sample accuracy (*Acc*) and balanced accuracy (*Bac*) are given in Tables 4-8 where values of “true positive”, “true negative”, “0 positive”, “1 positive”,

and “2 positive” are also displayed. For the binary case, “true positive” shows the probability of correct prediction for State 0 (fluent). Similarly, “true negative” indicates the probability of correct prediction for State 1 (congested). For the ternary case; “0 positive”, “1



positive”, and “2 positive” correspond to the probabilities of correct prediction of State 0 (fluent), State 1 (moderately congested), and State 2 (congested), respectively. In Table 6, we display the *RMSE* values of Algorithm 1 for the data obtained in 2013 and 2016 from the same highway section.

The processing time values (in seconds) in the last columns of Tables 4-7 are measured on a PC having an Intel Core i7 CPU with 6 GB of RAM memory and 2.40 GHz processor speed using the Matlab R2011a software. In each column, the algorithm with the best performance is emphasized by employing boldface characters.

When we compare the prediction performances in Tables 4-7 for 2013 and 2016 data of the same highway section, we observe that the performances are close as expected. However, as can be seen from the *Acc* and *Bac* values, the prediction performance for

2013 was slightly better than for 2016. If we examine the ATS data more closely, we can explain the better performance for 2013 as follows. Firstly, we observed in Tables 4-7 that if the probability of correct prediction of State 0 (for binary case it is “true positive”, and for ternary case it is “0 positive”) is high, then *Acc* increases. Hence, we compared the frequencies of occurrence of State 0 (fluent) in the entire data sets of 2013 and 2016 and noticed that they were 75% and 71% for 2013 and 2016, respectively. This indicates that, as we expected, State 0 appeared more frequently in the 2013 dataset. Secondly, we wanted to detect the unexpected traffic events (accidents, rain, snow, etc.) in those two years. For that purpose, we evaluated the addition of probabilities of abrupt transitions from the fluent state (State 0) directly to the congested state (State 2) and vice versa. Additions of probabilities of these sudden transitions were 0.0179 and 0.0226

Table 4 – Performance evaluations of proposed algorithms for ternary case using data of 2013

	0 Positive	1 Positive	2 Positive	<i>Acc</i>	<i>Bac</i>	Time [s]
Algorithm 1	0.9780	0.1741	0.9410	0.9544	0.6977	0.2427
Algorithm 2	0.9858	0.17	0.2099	0.7921	0.4552	0.3158
Algorithm 3	0.9777	0.2105	0.9304	0.9524	0.7062	0.4139
Algorithm 4	0.9647	0.1134	0.8816	0.9296	0.6532	0.1853
Algorithm 5	0.9857	0.1053	0.8777	0.9443	0.6562	0.2023

Table 5 – Performance evaluations of proposed algorithms for binary case using data of 2013

	True positive	True negative	<i>Acc</i>	<i>Bac</i>	Time [s]
Algorithm 1	0.9780	0.9353	0.9674	0.9567	0.1335
Algorithm 2	0.9777	0.9326	0.9665	0.9552	0.5160
Algorithm 3	0.9777	0.9326	0.9665	0.9552	0.5867
Algorithm 4	0.9634	0.8802	0.9427	0.9218	0.2379
Algorithm 5	0.9634	0.8802	0.9427	0.9218	0.1226

Table 6 – Performance evaluations of proposed algorithms for ternary case using data of 2016

	0 Positive	1 Positive	2 Positive	<i>Acc</i>	<i>Bac</i>	Time [s]
Algorithm 1	0.9717	0.1506	0.9281	0.9400	0.6835	0.1648
Algorithm 2	0.9777	0.1365	0.2299	0.7608	0.4480	0.4172
Algorithm 3	0.9692	0.1553	0.9179	0.9357	0.6808	0.5423
Algorithm 4	0.9529	0.0871	0.8657	0.9087	0.6352	0.2441
Algorithm 5	0.9801	0.0659	0.8579	0.9255	0.6346	0.2654

Table 7 – Performance evaluations of proposed algorithms for binary case using data of 2016

	True positive	True negative	<i>Acc</i>	<i>Bac</i>	Time [s]
Algorithm 1	0.9717	0.9260	0.9586	0.9489	0.1507
Algorithm 2	0.9692	0.9233	0.9561	0.9463	0.6752
Algorithm 3	0.9692	0.9233	0.9561	0.9463	0.7891
Algorithm 4	0.9500	0.8685	0.9266	0.9093	0.1604
Algorithm 5	0.9500	0.8685	0.9266	0.9093	0.1617

Table 8 – RMSE values for Algorithm 1

Dataset	RMSE
2013	10.5606
2016	10.5200

for the datasets of 2013 and 2016, respectively. This slight increase can be the result of the fact that the 2016 dataset covers a longer time span of challenging winter weather conditions, and we can infer that a greater number of unexpected events occurred in 2016. Thus, the above two factors may explain the minor performance degradation in simulations for the 2016 dataset.

When we compare the results of binary and ternary cases, we observe that binary prediction is more successful than ternary prediction, especially when *Bac* is taken into consideration. Binary prediction is more general, with only one threshold; hence its error rate is lower, as expected.

When we look at the results in terms of *Acc* and *Bac*, we see that Algorithm 1 is the most successful method for both binary and ternary prediction and for both datasets of 2013 and 2016. Finally, in Table 8, when we compare the *RMSE* values (the error between the actual speed data and the predicted speed data) of Algorithm 1 for 2013 and 2016, it can be seen that there is a negligible difference between the two years.

## 5. CONCLUSIONS

In this study, the future state of traffic (as ATS or traffic status) on a busy highway section is predicted by five different simple and efficient algorithms. In Algorithm 1, the average speed values were predicted by looking at past data for two different datasets belonging to the years of 2013 and 2016. In the other four algorithms, the same datasets were used after being converted into the binary or ternary labeled states in order to predict the future traffic status. Looking at the traffic data of the same highway section in 2013 and 2016, it was observed that the average number of vehicles increased from 319 in 2013 to 756 in 2016. However, the average speed values did not quite change. From this result, it can be deduced that although the traffic load increased, ATS stayed nearly the same due to possible infrastructure improvements.

When the algorithms are evaluated using performance criteria, Algorithm 1 seems the most successful one in terms of *Acc* and *Bac*. However, for all the algorithms, the probability of correctly predicting State 1 for ternary case is extremely low, and because of this *Bac* values are also low. It has been suspected that the lack of success for predicting State 1 in the ternary version is caused by the adverse effects of unexpected traffic events involving rush hour, accidents, and seasonal weather conditions such as rain or snow.

Notwithstanding the failure for this specific case, one of the greatest advantages of the proposed methods against other existing methods is in terms of the processing time, as can be seen in the last columns of the tables. All the proposed algorithms predict ATS or traffic status in a very short time. When the algorithms are evaluated in terms of *Acc*, their success rates surpass 90% (96.74% at maximum), which is another important advantage of the proposed methods. In addition, in this study, predictions were performed using only the past speed data without the requirement of multiple parameters such as total number of vehicles, average occupancy, day of the week, time of the year, weather conditions, etc., which were used in other relevant studies. Therefore, the reliable predictions obtained in a very fast manner by our proposed algorithms can be considered as a modest contribution to the related field.

### FEVZİ YASIN KABABULUT<sup>1</sup>

(Corresponding author)

E-Mail: fyasin.kababulut@udhb.gov.tr

Doç. Dr. DAMLA KUNTALP<sup>2</sup>

E-Mail: damla.kuntalp@deu.edu.tr

Doç. Dr. OLCAY AKAY<sup>2</sup>

E-Mail: olcay.akay@deu.edu.tr

Dr. TİMUR DÜZENLİ<sup>3</sup>

E-Mail: timur.duzenli@amasya.edu.tr

<sup>1</sup> Ulaştırma ve Altyapı Bakanlığı

IV. Bölge Müdürlüğü Çeşmebaşı Mah.

Alpaslan Cad. No:2 Ovaakça-Osmangazi/Bursa, Türkiye

<sup>2</sup> Elektrik ve Elektronik Mühendisliği Bölümü

Dokuz Eylül Üniversitesi, Tınaztepe Kampüsü,

Buca, İzmir 35100, Türkiye

<sup>3</sup> Elektrik - Elektronik Mühendisliği Bölümü

Teknoloji Fakültesi, Amasya Üniversitesi

A-Blok, Oda No: A303, Amasya 05100, Türkiye

## SADECE GEÇMİŞ HIZ BİLGİSİNİ KULLANARAK YAKIN GELECEKTEKİ TRAFİK DURUMUNUN BASİT VE VERİMLİ TAHMİNİ

### ÖZET

Akıllı trafik sistemleri, şehir hayatının en önemli sorunlarından biri olan trafik sıkışıklığı problemini çözmeye çalışmaktadır. Bu çalışmada; probleme, geçmiş ortalama yolcu hızını (OYH) kullanan trafik durum tahmini yoluyla yaklaşmaktayız. Trafik durumunun tahmini için beş farklı algoritma önerilmektedir. Bu algoritmalar, İstanbul Büyükşehir Belediyesi Trafik Kontrol Merkezi tarafından sağlanan gerçek verilere uygulanmıştır. 1. Algoritma, bir otoyol kısmı üzerinde, aynı kısım üzerinden elde edilen geçmiş hız bilgilerini kullanarak gelecek OYH değerini tahmin etmektedir. Önerilen diğer algoritmalar, algoritma 2'den 5'e doğru, yine aynı yol segmenti üzerindeki geçmiş hız bilgilerini kullanarak yolun gelecekteki durumunu akıcı, orta seviyede sıkışık veya sıkışık olarak tahmin etmektedir. Burada trafik durumları, daha önceden belirlenmiş OYH aralıklarına göre atanmaktadır. Önerilen algoritmalarda, giriş verisi olarak, ardışık şekilde 10 dakika aralıklarla ölçülmüş geçmiş 5 OYH değeri kullanılmıştır.

Önerilen algoritmaların başarımı, ortalama karekök hatası (OKH), örnek doğruluğu, dengelenmiş doğruluk ve işlem süresi cinsinden değerlendirilmiştir. Önerilen algoritmalar, görece basit olmalarına ve sadece geçmiş hız değerlerine ihtiyaç duymalarına rağmen, tahmin hatası fark edilir seviyede düşük olan güvenilir sonuçlar sağlamaktadırlar.

## ANAHTAR KELİMELELER

OYH tahmini; araç trafiği; trafik durumunun tahmini;

## REFERENCES

- [1] Lighthill MJ, Whitham GB. On Kinematic Waves II: A Theory of Traffic Flow on Long Crowded Roads. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*. 1955;229(1178): 317-345.
- [2] Richards PI. Shock Waves on The Highway. *Operations Research*. 1956;4(1): 42-51.
- [3] Yang L. *Stochastic traffic flow modeling and optimal congestion pricing*. PhD thesis. University of Michigan; 2017. Available from: <https://search.proquest.com/docview/1151383009?pq-origsite=gscholar>
- [4] Qui Z. *Macroscopic traffic state estimation for large scale freeway network using wireless network data*. PhD thesis. University of Wisconsin-Madison; 2007.
- [5] Škorput P, Mandžuka S, Jelušić N. Real-time Detection of Road Traffic Incidents. *Promet – Traffic & Transportation*. 2010;22(4): 273-283. Available from: <http://www.fpz.unizg.hr/traffic/index.php/PROMTT/article/view/192/97>
- [6] Munoz L, Sun X, Horowitz R, Alvarez L. Traffic Density Estimation with The Cell Transmission Model. *Proceedings of American Control Conference, 2003 June 4-6, Denver, CO, USA*. Available from: <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1240418>
- [7] Daganzo CF. The Cell Transmission Model: A Dynamic Representation of Highway Traffic Consistent with The Hydrodynamic Theory. *Transportation Research Part B: Methodological*. 1994;28(4): 269-287.
- [8] Daganzo CF. The Cell Transmission Model, Part II: Network Traffic. *Transportation Research Part B: Methodological*. 1995;29(2): 79-93.
- [9] Bosnjak I, Jusufurancic I, Visnjic V. Modelling Framework for Dynamic Multiclass Traffic Assignment in ITS Environment. *Promet – Traffic & Transportation*. 2004;16(2): 71-76. Available from: <http://www.fpz.unizg.hr/traffic/index.php/PROMTT/article/view/576/430>
- [10] Gülaçar H, Yaslan YH, Oktuğ SF. Short Term Traffic Speed Prediction Using Different Feature Sets and Sensor Clusters. *IEEE/IFIP NOMS 2016 Workshop: International Workshop on Platforms and Applications for Smart Cities (PASC), 2016 Apr 25-29, Istanbul, Turkey*. Available from: <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7503000>
- [11] Kababulut FY, Kuntalp D, Düzenli T. New Methods of Density Estimation for Vehicle Traffic. *9<sup>th</sup> International Conference on Electrical and Electronics Engineering (ELECO), 2015 Nov 26-28, Bursa, Turkey*. Available from: <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7394525>
- [12] Kay S. *Intuitive probability and random processes using MATLAB*. New York: Springer US; 2006.
- [13] Carrillo H, Brodersen KH, Castellanos JH. Probabilistic Performance Evaluation for Multiclass Classification Using The Posterior Balanced Accuracy. *ROBOT 2013: First Iberian Robotics Conference, 2013, Madrid, Spain*; 2013. p. 347-361. Available from: [https://link.springer.com/content/pdf/10.1007%2F978-3-319-03413-3\\_25.pdf](https://link.springer.com/content/pdf/10.1007%2F978-3-319-03413-3_25.pdf)