AUT Journal of Mathematics and Computing

AUT J. Math. Com., 3(2) (2022) 237-251
DOI: 10.22060/AJMC.2022.21399.1087

Original Article

A combined Apriori algorithm and fuzzy controller for simultaneous ramp metering and variable speed limit determination in a freeway

Mohammad Mahdi Zareian*[a], Mahmoud Mesbah[b], Sepehr Moradi[c], Mehdi Ghatee[c]

[a]*Department of Civil Engineering, University of Calgary, Canada*
[b]*Department of Civil and Environmental Engineering, Amirkabir University of Technology, Tehran, Iran*
[c]*Department of Mathematics and Computer Science, Amirkabir University of Technology, Tehran, Iran*

ABSTRACT: This paper proposes an integrated system to control ramps and adjust variable speed limits. It includes three essential modules to predict the starting time of congestion and a fuzzy controller to determine the parameters and a model predictive control. An Apriori algorithm that is a powerful tool for frequent pattern mining is used in the first module. The proposed system is neither sensitive to the traffic distribution nor computationally intensive. Two traffic simulators of Aimsun and CTMSIM are applied to validate the results. Compared with the most recent algorithms, including Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM), this system improves prediction accuracy up to 2.63%. The results of ramp metering and variable speed limit subsystems are also promising. The embedded controller shows 0.6% and 4% overall and rush hour improvement in the total travel time.

Review History:
Received:14 May 2022
Accepted:22 August 2022
Available Online:01 September 2022

Keywords:
Data mining
Frequent pattern mining
Apriori algorithm
Ramp metering
Variable speed limit

AMS Subject Classification (2010):
68T09; 90B20

(Dedicated to Professor S. Mehdi Tashakkori Hashemi)

1. Introduction

In the modern era, congestion has increased through the development of cities. Building new infrastructure such as freeways is sometimes impossible based on various constraints such as land use problems, economic burden, etc. So, governments must maximize the throughput of available infrastructure to solve or at least reduce the problem. To reach this end, considering the freeway as the main transportation vessel of modern cities plays a vital role because without spending huge money, delay, travel time, and safety of users can be increased. As a result, the utility of citizens and drivers will be increased too. It also has positive effects on other concerns in congested cities, such as air and noise pollution, sanitization, fuel consumption, global warming etc. This paper focuses on freeway control to improve these factors, which are essential in the modern era, especially in high-population cities such as Tehran, the capital of Iran. Our case study and data are related to Tehran too. There are different restrictions to control congestion ([44, 1]). Ramp metering (RM) and variable speed limit control (VSLC) are commonly used to control highway traffic, but they need real-time congestion prediction ([17, 27, 37]). Also, Some studies such as [21]

*Corresponding author.
E-mail addresses: mahdi.zareian@ucalgary.ca, mmesbah@aut.ac.ir, semolation@aut.ac.ir, ghatee@aut.ac.ir

Table 1: A summary comparison of controllers

Name	Mode	Advantages	Disadvantages	Features
ALINEA	Local	Ease to implement and implemented in real case studies	Lack of prediction and cannot react to dynamic changes in traffic flow	Most widely developed local ramp metering
SWARM	Competitive	Can improve performance of local ramp metering like ALINEA	Lack of cooperation between on-ramps.	It consists of two algorithms: the first one is for prediction and the second one is a local ramp metering.
HERO	Local & coordinated	Increase performance of ARIMA by a master-slave technique. Implemented in the real case study	Lack of prediction in the master-slave phase	Based on master-slave. A coordinated approach based on ARIMA
CTM	Coordinated	fewer parameters than METANET. Has a valid open-source simulator based on MATLAB.	More parameters than MFD	CTMSIM simulator is based on CTM. It helps many researchers to use it in their study.
Hierarchical	Coordinated	MFD has fewer parameters than CTM and METANET	Implementation of VSLC based on the MFD model is not solved yet.	Good choice for real-world application
METANET	Coordinated	The high precision model for ramp metering and VSLC on simulation and ideal situation	High number of parameters	Not practical in real case study because of complexity
GFLC	Coordinated	Supports VSLC and RM - fast and practical when an accurate system model or data is unavailable. Adaptive tuning of fuzzy controller parameters	Fuzzy system is simple (membership functions, inputs, and so on)	Fast genetic fuzzy approach. METANET model is used.

compare the effects of RM and VSLC on reducing travel time and increasing safety. Table 1 summarizes the most critical works on these subsystems.

In Table 1, A Local Feedback Control Law for On-Ramp Metering (ALINEA) ([30]), System-Wide Adaptive Ramp Metering (SWARM) ([29]), Heuristic ramp metering coordination (HERO) ([31]), cell transmission model based controller(CTM) ([9]), Macroscopic Fundamental Diagram based controller ((MFD) it is named Hierarchical in this paper) ([11]), METANET based controller ([6]), Genetic Fuzzy Logic Controller (GFLC) ([8]) are used. In the numeric result section ALINEA, SWARM, HERO, CTM, and Hierarchical are implemented.

The most important gaps in these systems are as follows:

- Some works are local and cannot predict the traffic during dynamic traffic changes.
- The accuracy of some works is not appropriate for controlling purposes.
- Some works are not usable because of their complexities and running times.

In this paper, the idea of [8] is used to implement a fuzzy controller. Although the presented fuzzy controller is similar to [8], the number of inputs and their values are different. Also, CTM has fewer parameters than METANET, so, it is considered as the basic model. Based on this policy, we can reduce complexity, so the proposed method improves the results in real-world applications. MFD model (Hierarchical ([11])) is not used because VSLC on the MFD is left to future works. For traffic flow prediction, short-term and long-term can be followed. The former predicts the next minutes or hours. The larger the forecasting horizon, the higher prediction's accuracy, see ([14]). Traffic prediction methods contain traffic assignment models [32, 7], and statistical and artificial intelligence methods, such as deep learning methods. [3] combined these two different approaches. Also, traffic prediction methods include parametric or model-based algorithms and non-parametric ones, see, e.g., [27]. Parametric methods are suitable in normal traffic flow conditions, but they are limited in the case of congestion or noisy data.

The parametric method based on the statistical time-series models, such as AutoregRessive Integrated Moving Average (ARIMA), uses a mathematical model for explaining the history of data as a time-series. It focuses on the mean of the data, and it may lose valuable information. Some non-parametric methods such as k-NN and SVR can be applied for traffic prediction ([27]). [43] utilized an improved k-NN for short-term traffic flow prediction.

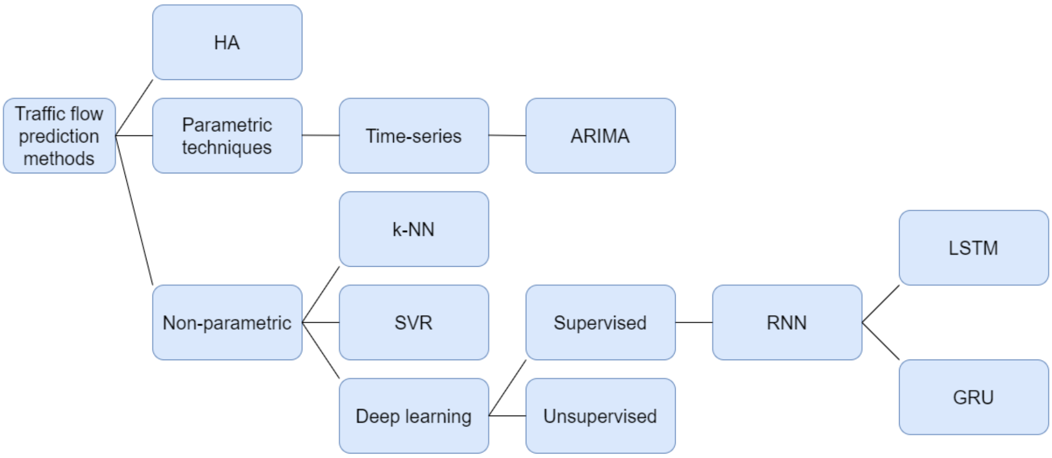

Figure 1: A roadmap of machine learning approaches for traffic flow prediction.

However, it cannot deal very successfully with stochastic and non-linear traffic features. Another non-parametric method is deep learning (DL) which can be developed in supervised and unsupervised techniques. Recurrent Neural Network (RNN) works well with time-series of traffic as it can extract relations from historical data by getting feedback from the output. When there is a more extended dependency between data, RNN fails. In such cases, Long Short-Term Memory (LSTM) can model time-series in short and long-term predictions ([27]). See [25], or [16]. Another effort is Gated Recurrent Unit (GRU) [4]. Unlike LSTM, GRU is more efficient and simpler in the training phase ([27]). It uses fewer training parameters and its accuracy is better than LSTM in some cases [20]. Figure 1 displays details of research branches and Table 2 summarizes the benefits as well as the limitations of the most effective prediction methods. For more research, one can refer [27].

Although these methods predict the traffic successfully, they cannot present an explainable plan for traffic prediction. To solve this gap, we follow a frequent pattern mining approach. The previous works also predict the congestion in the future, but we focus on specifying the congestion's starting time. Thirdly, the earlier works are sensitive to traffic distribution, but we extract useful information from the data patterns by applying the Apriori algorithm.

The rest of the paper is organized as follows. The frequent pattern mining based method for prediction is presented in Section 2. Section 3 uses the congestion prediction for an integrated ramp metering and a variable speed limit system. To this end, a fuzzy controller is applied. Numerical results are stated in Section 4. The final section presents a short conclusion with some proposals for future works.

2. Frequent pattern mining based prediction

Data mining is the base for various transportation studies. It is also necessary to examine the amount of data before and after implementing different intelligent transportation studies [24]. Considering these concepts, we wish to implement a method for estimating congestion's starting time near on-ramps. On-Ramp is a bottleneck in the freeway, where the formed congestion propagates upstream, so this prediction is essential for ramp controlling. The architecture of this method is presented in Figure 2. In what follows, the components of this architecture are explained.

2.1. Discretization

Use of classic frequent patterns in a freeway with continuous traffic flows is impossible. To solve this problem, we used the technique of [22] and [23]. For example, [23], defined a fuzzy clustering algorithm with k=3 clusters for each variable such as speed or density. Normally, in fuzzy clustering, the total number of clusters is an algorithm input. The main difference between fuzzy k-means clustering and classic k-means clustering is that in classic fuzzy k-means, each variable (in our case, the reported value of speed or density by a sensor) only belongs to one cluster, zero or one scenario. On the other hand, in fuzzy k-means clustering, one number or variable can be in more than one cluster based on probability. For example, suppose one sensor reported 60 km/h as speed. In the fuzzy k-means clustering, this value can be in cluster 5 with a probability of 60% and in cluster 6 with a probability of 30% simultaneously. This policy improves our clustering method's resulting clusters because we got better results when we changed our clustering method from classic k-means to fuzzy k-means based method. According to experimental results, the suitable value of k is less than 10, and there is no considerable difference between utilizing speed or density because of their correlation. By applying fuzzy k-means clustering for each on-ramp independently, the centers of the clusters can be declared. Fuzzy clustering needs to be independent for each on-ramp because of their distribution. Figure 3 reveals an example of this occurrence for two on-ramps. It means that if there are 7 on-ramps in our case study, 7 independent fuzzy k-means clustering are required.

2.2. Thresholding

Next, a method is required to identify the start of congestion on the freeway. A cluster number shows this congestion determined manually by the system engineer. For example, if the purpose of prediction is the speed detection when mainstream speed moves below 50 km/h and it is in cluster 4, so index 4 will be our threshold. This congestion can be detected in the database when the speed changes from index 5 to index 4 for the first time. The phrase ”for the first time” in the last sentence means that we want to find the first time the variable (speed) changes to index 4, so we will declare congestion when the speed moves to cluster 4 and the speed was higher than cluster 4 at least for m sampling time, where m is an input parameter of the system. This is because speed on the mainstream can fluctuate during rush hours, for example between clusters 3-4-5, and detecting this fluctuation is not fruitful for the system, so we need to disregard them. Also, the system engineer can determine this threshold according to the Level of the Service or LOS ([39]).

Subsequently, Traffic Detection Criterion Sensor (TDCS) needs to be introduced. AIMSUN and CTMSIM [19] simulators are used for creating the simulation data according to our real data. Figure 4 reveals one part of a freeway in Tehran implemented by AIMSUN.

As seen in Figure 4, the first sensor before each on-ramp is called Traffic Detection Criterion Sensor (TDCS). This selection is because the freeway’s congestion has a ”back-propagation” feature as congestion propagates from downstream to upstream. Some studies such as [12] suggest that the jam head has a constant propagation speed which is near 18 km/h for jam waves. Also, there are 7 on-ramps in our case study, so we have 7 TDCSs. Each TDCS is used to predict congestion through frequent pattern mining, which is explained in the following subsection. The function of TDCS is to detect congestion beginning with the described situation. Overall, through this method, we have some congestion starting times for each on-ramp (each TDCS) in our dataset.

Finally, a training dataset for frequent pattern mining must be created based on the data collected by each TDCS. To this end, samples between bt (beginning time) and bt-et (end time) from sn (number of sensors) sensors before each TDCS will be collected where et, bt and sn are the system’s inputs and they will be explained. For example, as seen in Figure 4, we recommend that 5 sensors before each TDCS are enough, so one useful value for

Table 2: Described methods with a summary of the most important features of them

Method	Advantages	Disadvantages
HA ([34, 35])	1) Ease of implementation 2) Speed of execution	1) response to sudden changes and events
ARIMA ([2])	1) Time-series based technique 2) Good for linear patterns	1) need a large database 2) rely on mean and miss information about far points.
ML ([43])	1) requires no assumption of an underlying relationship	1) time-consuming for large datasets 2) more error in non-linear problems
DL ([27])	1) requires no assumption of an underlying relationship 2) can handle stochastic and non-linear relationships	1) black box 2) complex training
Traditional RNN ([27])	1) requires no assumption of an underlying relationship 2) can handle stochastic and non-linear relationships 3) more stable on sequential data in short-time	1) black box 2) complex training 3) not perfect for long interdependency in data
LSTM ([25, 16])	1) requires no assumption of an underlying relationship 2) can handle stochastic and non-linear relationships 3) suitable for short and long-term prediction 4) better performance than LSTM when the database is large	1) black box 2) complex training 3) need a more extensive database than GRU for the best accuracy
GRU ([4])	1) requires no assumption of an underlying relationship 2) can handle stochastic and non-linear relationships 3) suitable for short and long-term prediction 4) use fewer training parameters, so need a smaller database for training	1) black box 2) complex training

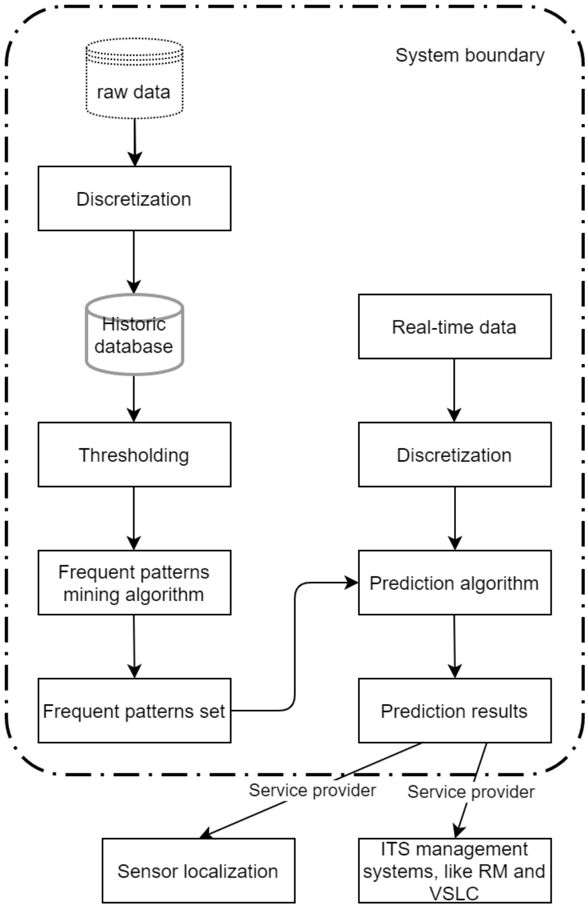

Figure 2: Architecture of congestion prediction by frequent patterns

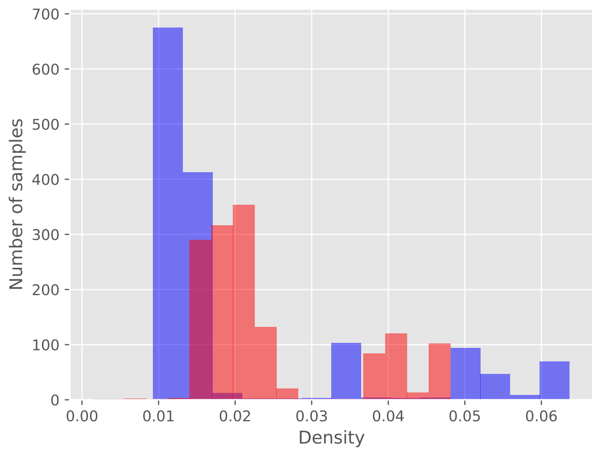

Figure 3: An example of the density distribution near two on-ramps after normalization in one day.

sn can be 5; meanwhile, these parameters can be determined by the tuning phase which will be described later. For understanding the 2 parameters st and bt, Figure 5 can be useful. In Figure 5, reported values by sn sensors before the TDCS will be extracted in time intervals between (bt, et). In order to clarify this process, suppose bt = 2 , et = 6 , sn = 5 and TDCS declare a jam in 18:10. In this situation, all reported data by these 5 sensors between 18:04 until 18:08 will be extracted. As one can see, interval length equals to 4 minutes and with time sampling (ts) = 30 seconds, 5 * 2 * 4 = 40 values (speed) will be collected for each detected jam by TDCS. This data is useful for frequent pattern mining on the jam condition. In the parameters tuning section (4.1.1), more detail is available.

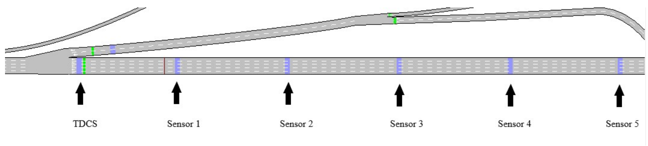

Figure 4: Screenshot of AIMSUN simulator from one on-ramp in the case study freeway in Tehran. The first sensor before the on-ramp is a Traffic Detection Criterion Sensor (TDCS).

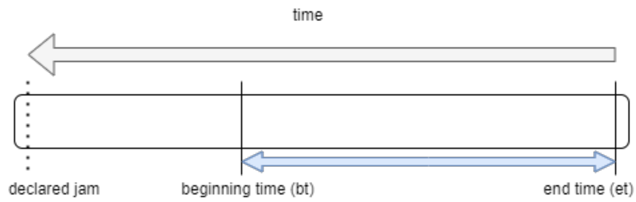

Figure 5: Overview of the congestion data extraction according to the time, the blue arrow shows the sampling time from the sensors.

2.3. Frequent patterns mining of traffic flows

As stated in the introduction section in detail, each of the statistical or ML or DL based approaches has flaws, so we want to eliminate these weaknesses through frequent pattern mining.

The next problem explained here is the frequent pattern mining process. In classic frequent pattern mining problems, item order in itemset is unimportant. Conversely, the order of reported values by each on-ramp in every sampling time plays a vital role. Suppose sensors of Figure 4 report speed in one sampling time such as in table 3. A point that should be repeated here is that these values are the index of each fuzzy k-means clustering.

Table 3: Two reported sampling times by five sensors. Both samplings i and j have the same values, but they show different traffic streams

Measured speed	Sensor 1	Sensor 2	Sensor 3	Sensor 4	Sensor 5
Sampling i	2	3	3	4	4
Sampling j	2	4	4	3	3

Both sampling i and j report the same values {2,3,3,4,4}, so a method is required to consider sensor number (consequently order of the sensors) in the itemsets. For this purpose, the solution described below is used. First, we suppose k in the fuzzy k-means clustering is less than 10; then we can sort the sensors based on the sensor numbers allocated to them by geographical positioning (flow stream), as seen in Figure 4. Next, , the sensors will create a two-segment number for each reported value. The right segment of this number is the reported speed (according to the fuzzy k-means algorithm with k ¡ 10) while the left segment is the sensor number. Through this policy, the values of Table 3 are transformed into Table 4.

In Table 4, the output of itemsets in the new format is:
Sampling i : { 12 , 23 , 33 , 44 , 54 }
Sampling j : { 12 , 24 , 34 , 43 , 53 }
Hence each 2-digit number is called a classic item in frequent pattern mining algorithms.

In the last process of this section, an Apriori algorithm such as [41] is implemented and utilized to find the frequent patterns of collected data. For more information reading [41] and [40] is recommended. Although the running time of the Apriori algorithm in frequent pattern mining is not a bottleneck in this paper, paper [33] can be used to reduce the algorithm's running time and storage space. Running time is not important here because the

Table 4: reported samplings of Table 3 are modified with the described method. The unit number is the reported speed cluster by the sensor and the digit number is the sensor number.

Measured speed	Sensor 1	Sensor 2	Sensor 3	Sensor 4	Sensor 5
Sampling i	12	23	33	44	54
Sampling j	12	24	34	43	53

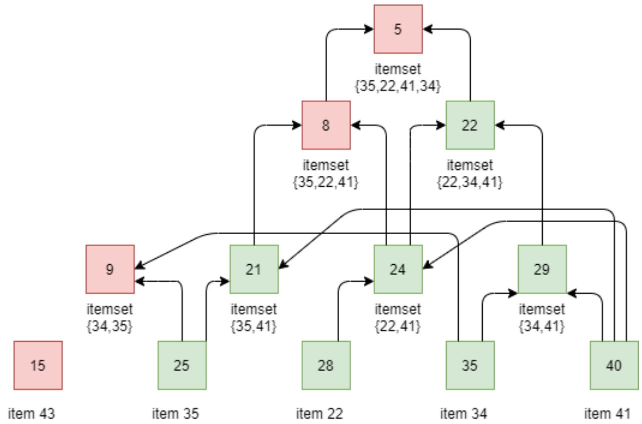

Figure 6: One part of the Apriori algorithm on 5 items with Minimum Support Threshold (MST) equals 20. Values inside the boxes are the support of each itemset and red boxes mean that their supports are less than MST, so they cannot be candidates for joining process in the Apriori algorithm. Also, green boxes mean that they are candidate. Further, in this example, itemset {22,34,41} is max pattern. One point about this figure is that all of the itemsets have not been displayed for better visualization.

process of finding the frequent patterns can be done at nightfall; through this policy, there is no sensation on the running time that is costly. Also, the frequent patterns can be updated according to time; for example, they can be updated every week or month according to the last data.

Through the described process, a set of frequent patterns is created for each on-ramp based on collected data. Also, our tests show that the prediction accuracy is dependent on data size, i.e. more data is associated with higher accuracy. However, larger data also causes the growth of the storage space and running time.

Thereafter, a probability can be determined for the incoming flow near each on-ramp in the test data with more detail described in the numerical results section. This probability is the likelihood of creating congestion for the next (bt,et) minutes. Then, based on a threshold called the decision threshold which is an input of the system, a decision can be made about the formation of congestion.

3. Ramp metering and variable speed limit control

3. This section develops an integrated cooperative ramp metering and variable speed limit control (VSLC) based on frequent patterns and model predictive control (MPC). This controller contains two main parts:

1. A passive fuzzy ramp metering controller based on the frequent patterns
2. An active controller (RM and VSLC) based on MPC.

Note that VSLC always works by MPC. Also, CTMSIM, a macroscopic traffic simulator is utilized. This simulator is based on Asymmetric Cell Transmission Model (ACTM), proposed by [9]. Figure 7 demonstrates our case study in CTMSIM. Other famous traffic flow models are:

- [18]: METANET,
- [6]: A newer and extension version of METANET,
- [11]: The Macroscopic Fundamental Diagram (MFD) model. Also, the MDF based controller ([11]) is implemented to compare with the proposed controller that is shown in Figure 8.

  A point here is that our controller is based on predictions. The more data, the more accuracy, so, we can have a better controller with more data. Some studies such as [24] suggest at least 2 months of data (60 days) is required.

3.1. Fuzzy controller based on frequent pattern mining

In our system, the fuzzy controller is a passive ramp metering controller. Some studies such as [8, 45, 38, 36] used fuzzy controllers. Fuzzy systems can run in real-time and they have tolerance against noisy data, so, a fuzzy controller is utilized based on the predictions of frequent pattern mining. The system has three inputs: density,

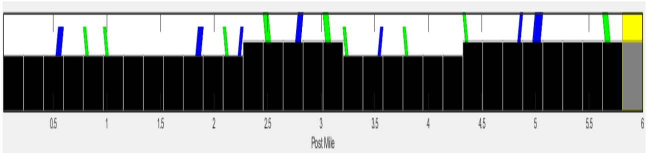

Figure 7: The architecture of the case study freeway in the CTMSIM. Blue ramps are on-ramp and green ones are off-ramp.

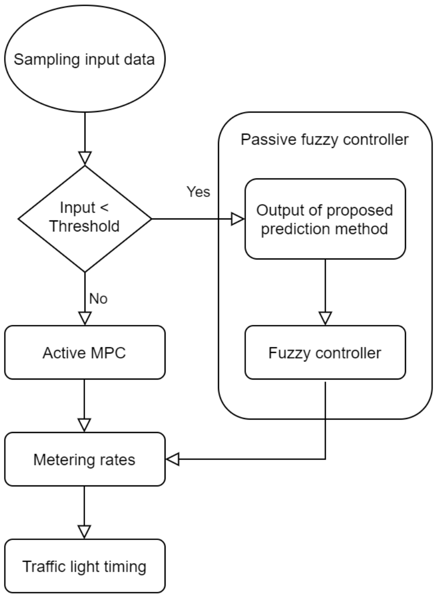

Figure 8: The general architecture of controller system

queue, and frequent pattern prediction, with each being scaled into [0,1]. Density and queue are the ratios of the current density (near each on-ramp) and on-ramp queue to maximum density plus queue capacity. The third input is the chance of the congestion's beginning in the next decision time. Further, the output is the metering rate for each ramp. Metering rate is a number within the range [0,1] and shows the portion of the cars the on-ramp must halt their entrance to the freeway in their queue for the next decision time. For each variable, the Gaussian membership function is used. Values of each input and output are mentioned below:
Density ={ low, medium, high, very high, extreme high }
Frequnet pattern = { very low, low, medium, high, very high }
Queue = { low, medium, high }
Metering rate = { very low, low, medium, high, very high }.

The paper uses the idea of [5] to automatically determine the fuzzy system's parameters based on the data distribution. For example, Critical Density (CD) is used to determine parameters in the density input as critical density has proved to be a good traffic flow measure which is dependent on the freeway. Our fuzzy controller and our rules are similar to [8]. For more information see that paper.

3.2. MPC

In this system, a fuzzy controller is a passive ramp metering controller. If the density is larger than a threshold, MPC will be activated as MPC optimizes the performance. In this situation, traffic flow fluctuates less than during the rush hour, so the algorithm running time is insignificanthas. There are two different optimization functions for ramp metering (1) and variable speed limit control (2) that should be coordinated. Since CTM is used to model the traffic flow by CTMSIM, the decision time is 5 minutes. Further, the freeway is divided into larger than normal cells in the CTM to implement a variable speed limit. Our normal cells have a length of 300 meters; in contrast, cells of VSLC have an average distance of 1 km. This policy cannot change the recommended speed every 300 meters as drivers need more time to adjust the new speed. Also, these rapid changes have some safety problems on the freeway. The maximum speed allowed in our case study is 90 km/h. Thus, to implement VSL, a discrete collection of speeds is created: {40, 50, 60 ,70, 80}. As explained in 2, —v(i)-v(i-1)— ¡= 1 is added to the VSLC optimization to enhance the safety and to smoothen the flow. The optimization problem is solved by stochastic

gradient descent. Finally, to ensure that a global optimum is achieved, 100 starting points are used.
Metering rate = MR,

Ramp metering:

$$\begin{aligned} \min \quad & \Big(\quad a * \sum Delay_{\text{on-ramps}} + b * \sum Traveltime_{\text{mainstream}} \quad \Big) \\ & \text{s.t.} \quad 0 \leq MR(i) < 1 \qquad \forall i \in on - ramps \\ & 0 < a, b \leq 1 \end{aligned} \tag{1}$$

VSLC:

$$\begin{aligned} \min \quad & \Big(\quad c * \sum Traveltime_{\text{mainstream}} + d * Variance(v(j)) \quad \Big) \\ & \text{s.t.} \quad v(j) \in \{40, 50, 60, 70, 80\} \quad \forall j \in cells \\ & |v(j) - v(j-1)| \leq 1 \\ & 0 < c, d \leq 1 \end{aligned} \tag{2}$$

a and b can change manually based on the field study or they can choose through tuning based on the performance of the system; however, some studies such as [11] prefer to set default value a = b = 1. In our research as with [10] we use a = 0.4 , b = 0.6. Also, c and d can be set through tuning or default value 1. We use tuning based on the system's performance.

3.3. Ramp signal timing

In the last part of the system, a module is required for transforming the metering rate to traffic signal cycle as we need to report it for the next decision signal time in the real-world application. In the on-ramp, only one or two cars (per lane) can enter the freeway at each green signal, so this module must work based on reality. Some researchers such as [15] demonstrate that the queue cars have a reaction time to the traffic signal based on their position in the queue. Generally, cars in front of the queue have a longer reaction time or departure headway. The first car in the line has about 4 seconds of departure time. Also, departure headway converges to (2,2.5) seconds. Thus, our module has a minimum and maximum green signal time phase to satisfy the two mentioned constraints. The cycle time (green + red) cannot be less than 4 seconds (based on departure time).

Based on the metering rate, we can find approximate cars that each on-ramp must let them enter the freeway in the next decision time, so for each on-ramp, signal timing can be calculated by:
Metering Rate = MR , Estimated Cars = EC , Cycle = C , Number of Cycles = NoC , Number of Lines = NoL , Decision Time = DT , Green Light = GL , Red Light = RL ,Allowed Cars = AC

$$(1 - MR) * EC = AC \quad , \quad NoC = C/DT$$

$$RL + GL = C \quad , \quad GL = 2 \quad \Rightarrow \quad RL >= 2$$

$$C * NoL * NoC = AC$$

$$\Rightarrow \quad (RL + 2) * NoL * ((RL + 2)/DT) = AC$$

$$\Rightarrow \quad (RL + 2)^2 NoL/DT = AC$$

$$\Rightarrow \quad RL = \sqrt{(AC * DT)/NoL} - 2$$

4. Numerical results

In the first subsection, frequent pattern prediction results will be discussed. Next, ramp metering and variable speed limit controller results will be mentioned.

4.1. Numerical results of prediction

In this subsection, first, parameter tuning will be discussed. Then, numeric results are explained.

4.1.1. Parameters tuning

To improve prediction accuracy, parameter tuning is helpful. To this end, a set of discrete values is considered, then the program runs with all of the values, and finally, the value leading to the best output according to the prediction score will be set to the variable.

A list of the input parameters in this paper is described below:

1. k: number of the clusters in the fuzzy clustering algorithm. Values 7 and 9 are recommended. Larger values can lead to reduced support of the itemsets that are mined in section 2.3 where smaller values can decrease the ability to recognize speed changing as the size of clusters in the fuzzy clustering algorithm enlarges.
2. bt: This value describes the start time of the sampling for data extraction shown in Figure 5. For tuning this input, values 2 or 3 (minutes) are recommended.
3. et: This value describes the end time of the sampling for data extraction as depicted in Figure 5. For tuning this input, values 6 or 8 (minutes) are recommended. Larger values can lead to reduced prediction accuracy.
4. sn: This input is the number of the used sensors for congestion detection before each on-ramp (before the TDCS) and recommended values for this input can be 4 or 5.
5. MST: This input value is used for determining a described itemset in section 2.3 is frequent or not. In this paper, relative support is utilized. For this input {0.3,0.4,0.5,0.6} is recommended, however, in our case study, the value of 0.3 gives the best output.
6. decision threshold: This input value is used in the test process for determining whether the input from the sn sensors will create congestion or not. This threshold has a great impact on the output of the system. This is because with the large value of the decision threshold, the number of the predictions labeled as the congestion will diminish, so there is more chance to predict more congestion correctly though this can lead to increased number of the predictions labelled as congestion incorrectly. Further, with the smaller ones, the chance of the congestion prediction decreases, while the chance of the wrong predictions (times that traffic flow has no jam and we predict it as jam) also declines. For this reason, a set decision threshold ={0.4,0.5,0.6} is used, then the best value based on the F-measure will be selected.

4.1.2. System Evaluation

In this subsection, three famous methods ARIMA, LSTM, and GRU are utilized for comparison. ARIMA has had a lot of applications during the past three decades in time series problems (for example [42]). Also, this model has a good output in general. LSTM is very fruitful in time series problems and can find a better long interdependency between the data; however, this network cannot be used for some problems such as image processing. Also, GRU is similar to LSTM but can work a slightly better in some situations, as it has fewer parameters than LSTM, so, GRU needs less training data. Table 5 compares the proposed method with the described methods in Table 2.

ARIMA, LSTM, and GRU equal windows are created for the prediction time to establish equal conditions between the proposed method. For more details, if the frequent pattern-based prediction in section 2.3, predicts congestion in the next 6-10 minutes which is dependent on the input values of bt and et, this prediction time implemented with ARIMA, LSTM, and GRU. Then, the reported prediction accuracy is the prediction accuracy of the moments when congestion's beginning is declared with the method of section 2.3. Remember that in general prediction (not just congestion's beginnings prediction) Arima and LSTM and GRU have absolutely accurate output and in this situation using the proposed method is nonsensical, unless some changes and improvements or extensions are utilized in the method presented in this paper. Also, the reason for the better accuracy of GRU than LSTM is that LSTM has more parameters, so it needs more training data for better accuracy. Hence, according to the size of the data, the accuracy of the GRU and LSTM can change. So, to make the long story short, each method predicts the next windows (6-10 minutes) based on their own method under similar conditions. The proposed method is described in section 1.

For further comparison between the methods, two new experiments are designed. In real-world applications, data can collapse, for example, by communication or sensor technical violation, so, we face some missing data. To

Table 5: Comparison of the methods.

Method	Difficulty to implement	Preparation and training time	Running time	Storage space	Explainability	Overall Ranking
HA	very low	very low	very low	very high	very high	8
ARIMA	low	low	very low	low	high	4
ML	medium	medium	low	low	medium	7
DL	medium	medium	medium	medium	low	6
Traditional RNN	high	high	medium	high	low	5
LSTM	very high	high	high	high	very low	3
GRU	very high	high	medium	high	very low	2
Proposed Method	medium	very high	medium	medium	medium	1

simulate this situation, in two separate tests, 5% and 10% of the data are randomly selected and deleted, i.e. they will be missed, and two new databases are created and four methods tested on them. The testing method is exactly the same as before, and just a minor modification is required to fill the missing values. For filling the missing values of sampling i, the following relation is utilized:

$$Sample(i) = 0.3 * [Sample(i-2) + Sample(i+2)]$$
$$+0.7 * [Sample(i-1) + Sample(i+1)]$$

Table 6 shows aggregated numeric results. As one can see, famous GRU method has a better accuracy when we have no missing data. However, proposed prediction method works better when there are some loss data in the network based on various reasons.

Table 6: Jam beginning prediction accuracy of four methods with the same prediction time under three different scenarios. The first column of each measure is related to our main scenario (no missing values), and the two following columns are related to the latest scenario based on randomly deleted values in our dataset.

Method	Precision			Recall			F-measure		
	No Loss	5% Loss	10% Loss	No Loss	5% Loss	10% Loss	No Loss	5% Loss	10% Loss
ARIMA	71.4	67.6	56.5	73.6	69.8	58.4	72.48	68.68	57.43
LSTM	82.2	78.8	72.6	74.1	70.9	60.7	77.94	74.64	66.12
GRU	85.8	82.4	74.8	80.1	77.3	71	**82.85**	79.77	72.85
Proposed Method	84.1	82	76.8	80.4	78.1	74.2	81.47	80	**75.48**

4.2. Controller numerical results

This section compares the results of the proposed controller for ramp metering and variable speed limit jointly. The proposed method will be compared with other methods under different scenarios. To this end, we compare ours with:

- [30]: ALINEA is the most famous ramp metering algorithm.
- [11]: We named it hierarchical in Table 7. It is based on the MFD model and framework.
- [19]: we used a version of the competitive ramp metering algorithm SWARM implemented in CTMSIM. it contains two modules (the first one is local and the second one is a global predictive ramp metering.). We made no modification or change in the version of SWARM that was implemented in the CTMSIM, which used to have more numeric results.
- [31]: HERO, a slave-master approach is used. At the local level, it contains ALINEA and at a higher level, a module monitors the on-ramps and declares them to master (if it is useful).Also, it is implemented at Monash highway, Australia.
- [9]: CTM is our based method implemented in CTMSIM. It only includes the basic MPC module (without prediction based on frequent pattern mining).

Table 7 summarizes results that improve the total travel time based on the no-controller situation. So, we considered the difference between travel time of no controller scenario and travel time when each controller method is used. All of them are implemented under the same situation with a decision time 5 minutes. Some papers call it decision windows or controller windows. The proposed system shows 0.6% overall improvement and 4% improvement in the rush hour in total travel time, so, one point can be concluded: in a freeway with high traffic flow during rush hour, using the proposed system is more rational. In our case study, the freeway has a high volume of cars during rush hours. The reason for this result is related to the proposed frequent pattern mining approach.

Table 7: Controller results based on the decision time equal to 5 minutes

	Overall improvement (%)	Rush hour improvement (%)	Non-rush hour improvement (%)
Proposed method	27.4	29.7	25.8
Proposed method without VSL	25.1	28.5	23.9
CTM + VSL	24.8	24.4	24.3
Hierarchical	26.8	25.7	**2**8.8
ALINEA	14.6	12.3	16.5
HERO	21.9	23.5	20.2
SWARM	15.1	15.4	14.9

5. Conclusion

This paper has proposed a new and fast method for predicting a specific traffic situation. According to the numerical results, this paper's novel approach can compete with traditional methods such as ARIMA, LSTM, and GRU in a defined field study. Based on the perspective of the paper's writers, this framework and the general solution can be fruitful in various fields especially those mentioned in section 3. According to Table 6, a significant conclusion is that although in the test under simulation conditions where there is no missing data or error in the data, the performance of the GRU is better, in the real-world implementation use of the proposed method is more reasonable because based on multiple factors some data can be declared as missing data. The paper's authors believe that the proposed method's better performance is using the fuzzy k-means algorithm in the discretization section. This new method opens the doors for researchers to use a data mining and frequent pattern-based approach for predicting events or to use them in some applications such as traffic flow control and sensors localization in the freeway, instead of methods that rely on some black boxes including the deep neural network, such as LSTM and GRU.

For future research, the proposed method can be enhanced and implemented in other fields and applications. Also, we believe that this paper is not the final point of this research; it is just a starting point. One improvement in the controller section is that some improvements to the fuzzy system can be fruitful, such as adding a new input to the system. Also, origin-destination estimation based on big data can help us to have a better traffic flow prediction; studies such as [28] are helpful. Besides, in some case studies, some on-ramps are more critical than others. In this situation, the idea of [26] can be used to set different priorities for the on-ramp, however, expert knowledge is needed.

Using and comparing other ramp metering methods in the controller section can be fruitful. For example, the game theory concept is practical in ramp metering. Based on Nash equilibrium, players or on-ramps compete to reach the equilibrium point. One of the best examples of related studies is [13]. The implementation of this approach at the macroscopic level can be helpful in the proposed method. It can be an option for our passive controller to react faster to dynamic changes in flow during rush hours.

Acknowledgement

The authors thank Dr. Alexander A. Kurzhanskiy from UC Berkley for sharing the CTMSIM simulator.

References

[1] S. Abpeykar and M. Ghatee, *A real-time decision support system for bridge management based on the rules generalized by cart decision tree and smo algorithms*, AUT Journal of Mathematics and Computing, 1 (2020), pp. 95–100.

[2] M. S. Ahmed and A. R. Cook, *Analysis of freeway traffic time-series data by using Box-Jenkins techniques*, no. 722, 1979.

[3] B. Alsolami, R. Mehmood, and A. Albeshri, *Hybrid statistical and machine learning methods for road traffic prediction: a review and tutorial*, in Smart Infrastructure and Applications, Springer, 2020, pp. 115–133.

[4] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, *Empirical evaluation of gated recurrent neural networks on sequence modeling*, arXiv preprint arXiv:1412.3555, (2014).

[5] H. R. Eftekhari and M. Ghatee, *A similarity-based neuro-fuzzy modeling for driving behavior recognition applying fusion of smartphone sensors*, Journal of Intelligent Transportation Systems, 23 (2019), pp. 72–83.

[6] J. R. D. Frejo, I. Papamichail, M. Papageorgiou, and B. De Schutter, *Macroscopic modeling of variable speed limits on freeways*, Transportation research part C: emerging technologies, 100 (2019), pp. 15–33.

[7] M. Ghatee and S. M. Hashemi, *Traffic assignment model with fuzzy level of travel demand: An efficient algorithm based on quasi-logit formulas*, European Journal of Operational Research, 194 (2009), pp. 432–451.

[8] A. H. Ghods, A. R. Kian, and M. Tabibi, *Adaptive freeway ramp metering and variable speed limit control: a genetic-fuzzy approach*, IEEE Intelligent Transportation Systems Magazine, 1 (2009), pp. 27–36.

[9] G. Gomes and R. Horowitz, *Optimal freeway ramp metering using the asymmetric cell transmission model*, Transportation Research Part C: Emerging Technologies, 14 (2006), pp. 244–262.

[10] M. Gregurić, E. Ivanjko, and S. Mandžuka, *The use of cooperative approach in ramp metering*, Promet-Traffic&Transportation, 28 (2016), pp. 11–22.

[11] Y. Han, M. Ramezani, A. Hegyi, Y. Yuan, and S. Hoogendoorn, *Hierarchical ramp metering in freeways: an aggregated modeling and control approach*, Transportation research part C: emerging technologies, 110 (2020), pp. 1–19.

[12] A. Hegyi, S. Shladover, X. Lu, and D. Chen, *A cooperative speed control algorithm to resolve jams*, Delft University of Technology: Delft, The Netherlands, (2012).

[13] S. Heshami and L. Kattan, *Ramp metering control under stochastic capacity in a connected environment: A dynamic bargaining game theory approach*, Transportation Research Part C: Emerging Technologies, 130 (2021), p. 103282.

[14] S. Ishak and H. Al-Deek, *Performance evaluation of short-term time-series traffic prediction model*, Journal of transportation engineering, 128 (2002), pp. 490–498.

[15] X. Jin, Y. Zhang, F. Wang, L. Li, D. Yao, Y. Su, and Z. Wei, *Departure headways at signalized intersections: A log-normal distribution model approach*, Transportation research part C: emerging technologies, 17 (2009), pp. 318–327.

[16] D. Kang, Y. Lv, and Y.-y. Chen, *Short-term traffic flow prediction with lstm recurrent neural network*, in 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), IEEE, 2017, pp. 1–6.

[17] M. Kontorinaki, I. Karafyllis, and M. Papageorgiou, *Local and coordinated ramp metering within the unifying framework of an adaptive control scheme*, Transportation Research Part A: Policy and Practice, 128 (2019), pp. 89–113.

[18] A. Kotsialos, M. Papageorgiou, C. Diakaki, Y. Pavlis, and F. Middelham, *Traffic flow modeling of large-scale motorway networks using the macroscopic modeling tool metanet*, IEEE Transactions on intelligent transportation systems, 3 (2002), pp. 282–292.

[19] A. Kurzhanskiy and P. Varaiya, *Ctmsim—an interactive macroscopic freeway traffic simulator*, Department of Electrical Engineering and Computer Sciences, Berkeley, CA., USA, (2008).

[20] J. Li, L. Gao, W. Song, L. Wei, and Y. Shi, *Short term traffic flow prediction based on lstm*, in 2018 Ninth International Conference on Intelligent Control and Information Processing (ICICIP), IEEE, 2018, pp. 251–255.

[21] Z. Li, C. Xu, D. Li, P. Liu, and W. Wang, *Comparing the effects of ramp metering and variable speed limit on reducing travel time and crash risk at bottlenecks*, IET Intelligent Transport Systems, 12 (2017), pp. 120–126.

[22] L. Lin, Q. Wang, and A. W. Sadek, *A novel variable selection method based on frequent pattern tree for real-time traffic accident risk prediction*, Transportation Research Part C: Emerging Technologies, 55 (2015), pp. 444–459.

[23] L. Lin, K. Yuan, and S. Ren, *Analysis of urban freeway traffic flow characteristics based on frequent pattern tree*, in 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), IEEE, 2014, pp. 1719–1725.

[24] X. Ma, A. Karimpour, and Y.-J. Wu, *Statistical evaluation of data requirement for ramp metering performance assessment*, Transportation Research Part A: Policy and Practice, 141 (2020), pp. 248–261.

[25] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, *Long short-term memory neural network for traffic speed prediction using remote microwave sensor data*, Transportation Research Part C: Emerging Technologies, 54 (2015), pp. 187–197.

[26] A. Meshkat, M. Zhi, J. L. Vrancken, A. Verbraeck, Y. Yuan, and Y. Wang, *Coordinated ramp metering with priorities*, IET Intelligent Transport Systems, 9 (2015), pp. 639–645.

[27] A. Miglani and N. Kumar, *Deep learning models for traffic flow prediction in autonomous vehicles: A review, solutions, and challenges*, Vehicular Communications, 20 (2019), p. 100184.

[28] M. Moghaddam, M. Mesbah, and M. Hickman, *Using big data sources for origin-destination matrix adjustment*, in 39th Australasian Transport Research Forum (ATRF), Auckland, New Zealand, 2017.

[29] G. F. Paesani, *System wide adaptive ramp metering in southern california*, in ITS America 7th Annual Meeting and Exposition: Merging the Transportation and Communications RevolutionsIntelligent Transportation Society of America (ITS America), 1997.

[30] M. Papageorgiou, H. Hadj-Salem, J.-M. Blosseville, et al., *Alinea: A local feedback control law for on-ramp metering*, Transportation Research Record, 1320 (1991), pp. 58–67.

[31] I. Papamichail, M. Papageorgiou, V. Vong, and J. Gaffney, *Heuristic ramp-metering coordination strategy implemented at monash freeway, australia*, Transportation Research Record, 2178 (2010), pp. 10–20.

[32] S. Peeta and A. K. Ziliaskopoulos, *Foundations of dynamic traffic assignment: The past, the present and the future*, Networks and spatial economics, 1 (2001), pp. 233–265.

[33] L. Shuwen and X. Jiyi, *An improved apriori algorithm based on matrix*, in 2020 12th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), IEEE, 2020, pp. 488–491.

[34] B. L. Smith and M. J. Demetsky, *Traffic flow forecasting: comparison of modeling approaches*, Journal of transportation engineering, 123 (1997), pp. 261–266.

[35] Y. J. Stephanedes, *Performance indicators and policy evaluation in rural transit.*, (1981).

[36] C. Taylor, D. Meldrum, and L. Jacobson, *Fuzzy ramp metering: Design overview and simulation results*, Transportation Research Record, 1634 (1998), pp. 10–18.

[37] S. Tscharaktschiew, *Why are highway speed limits really justified? an equilibrium speed choice analysis*, Transportation research part B: methodological, 138 (2020), pp. 317–351.

[38] J. Xu, X. Zhao, and D. Srinivasan, *On optimal freeway local ramp metering using fuzzy logic control with particle swarm optimisation*, IET Intelligent Transport Systems, 7 (2013), pp. 95–104.

[39] H. Yang, M. G. Bell, and Q. Meng, *Modeling the capacity and level of service of urban transportation networks*, Transportation Research Part B: Methodological, 34 (2000), pp. 255–275.

[40] Y. Ye and C.-C. Chiang, *A parallel apriori algorithm for frequent itemsets mining*, in Fourth International Conference on Software Engineering Research, Management and Applications (SERA'06), IEEE, 2006, pp. 87–94.

[41] X. Yuan, *An improved apriori algorithm for mining association rules*, in AIP conference proceedings, vol. 1820, AIP Publishing LLC, 2017, p. 080005.

[42] G. P. Zhang, *Time series forecasting using a hybrid arima and neural network model*, Neurocomputing, 50 (2003), pp. 159–175.

[43] L. Zhang, Q. Liu, W. Yang, N. Wei, and D. Dong, *An improved k-nearest neighbor model for short-term traffic flow prediction*, Procedia-Social and Behavioral Sciences, 96 (2013), pp. 653–662.

[44] Y. Zhang and P. A. Ioannou, *Stability analysis and variable speed limit control of a traffic flow model*, Transportation Research Part B: Methodological, 118 (2018), pp. 31–65.

[45] L. Zhao, Z. Li, Z. Ke, and M. Li, *Fuzzy self-adaptive proportional–integral–derivative control strategy for ramp metering at distance downstream bottlenecks*, IET Intelligent Transport Systems, 14 (2020), pp. 250–256.

Please cite this article using:

Mohammad Mahdi Zareian, Mahmoud Mesbah, Sepehr Moradi, Mehdi Ghatee, A combined Apriori algorithm and fuzzy controller for simultaneous ramp metering and variable speed limit determination in a freeway, AUT J. Math. Com., 3(2) (2022) 237-251
DOI: 10.22060/AJMC.2022.21399.1087