

The Optimization of an Intelligent Traffic Congestion Level Classification from Motorists' Judgments on Vehicle's Moving Patterns

Thammasak Thianniwet, Satidchoke Phosaard, and Wasan Pattara-Atikom

Abstract—We proposed a technique to identify road traffic congestion levels from velocity of mobile sensors with high accuracy and consistent with motorists' judgments. The data collection utilized a GPS device, a webcam, and an opinion survey. Human perceptions were used to rate the traffic congestion levels into three levels: light, heavy, and jam. Then the ratings and velocity were fed into a decision tree learning model (J48). We successfully extracted vehicle movement patterns to feed into the learning model using a sliding windows technique. The parameters capturing the vehicle moving patterns and the windows size were heuristically optimized. The model achieved accuracy as high as 99.68%. By implementing the model on the existing traffic report systems, the reports will cover comprehensive areas. The proposed method can be applied to any parts of the world.

Keywords—intelligent transportation system (ITS), traffic congestion level, human judgment, decision tree (J48), geographic positioning system (GPS).

I. INTRODUCTION

TRAFFIC reports in real-time are essential for congested and overcrowded cities such as Bangkok or even in sparse and remote areas during a long holiday period. Without these, commuters might not choose the proper routes and could get stuck in traffic for hours. Intelligent Transportation System (ITS) with automated congestion estimation algorithms can help produce such reports. Several initiatives from both private and government entities have been proposed and implemented to gather traffic data to feed the ITS. According to our survey, most efforts focus on limited installation of fixed sensors such as loop-coils and intelligent video cameras with image processing capability. However, the costs of such implementations are very high due to the high cost of the devices, installation, and maintenance. Moreover, these fixed sensors are vulnerable to extreme weather typical in certain areas. Additionally, the installation of fixed sensors to cover all roads in major cities is neither practical nor economically

This work was supported by the National Electronics and Computer Technology Center (NECTEC), Thailand.

W. Pattara-Atikom is with the National Electronics and Computer Technology Center (NECTEC), under the National Science and Technology Development Agency (NSTDA), Ministry of Science and Technology, 112 Thailand Science Park, Phahon Yothin Rd., Klong 1, Klong Luang, Pathumthani 12120, Thailand (phone: +662-564-6900 ext. 2528, e-mail: wasan@nectec.or.th).

T. Thianniwet and S. Phosaard are with the School of Information Technology, Institute of Social Technology, Suranaree University of Technology, 111 University Ave., Muang Nakhon Ratchasima, Nakhon Ratchasima 30000, Thailand. (e-mail: thammasak@sut.ac.th and s@sut.ac.th).

feasible. An alternative way to collect traffic data at a lower cost with wider coverage is therefore needed.

Recently, mobile sensors or probe vehicles appeared as a complementary solution to fixed sensors for increasing coverage areas and accuracy without requiring expensive infrastructure investment. Two popular types of mobile sensors are GPS-based and cellular-based. GPS-based sensors are sensors with GPS capability and cellular-based sensors are sensors that use information from cellular networks as traffic sensors.

Cellular-based sensors are low in cost due to the large number of mobile phones and their associated infrastructures already in service. According to recent statistics, the mobile phone penetration rate in Thailand is expected to grow to 90% in 2009 [1]. However, GPS-based sensors are far more efficient to pinpoint vehicle locations; thus they can provide highly accurate vehicle movement information. Moreover, recent mobile phones have integrated GPS capability, such as the popular Apple iPhone and several other "smart" phones.

In this paper, we explored a model that can automatically classify traffic congestion levels for traffic reports. The model can be further implemented in the system that combines advantages of GPS-based sensors, in that they are highly accurate, and of cellular-based sensors, in that they are highly available. This model combined with mobile sensors can generate traffic reports that cover virtually all of the areas that vehicles and mobile networks can reach.

This paper is organized as follows: In Section II, we describe related works concerning traffic congestion reports. The methodology of the research is presented in Section III. The parameters optimization is demonstrated in Section IV. Section V provides results and evaluations, and Section VI offers a conclusion and the possibilities of future work.

II. RELATED WORKS

Congestion level estimation techniques for various types of collected data are our most related field. Traffic data could be gathered automatically from two major types of sensors: fixed sensor and mobile sensor. The study in [2] applied a neural network technique to the collected data using mobile phones. It used Cell Dwell Time (CDT), the time that a mobile phone attaches to a mobile phone service antenna, which provides rough journey speed. Our work employed another machine learning technique that better fit with the characteristics of the data. The use of GPS data would provide more precise traffic information than that roughly provided by CDT. The studies

in [3] and [4] estimated congestion levels using data from traffic cameras by applying fuzzy logic and hidden Markov model, respectively. Our work applied decision tree (J48) technique on mobile sensors. Using data collected from mobile sensors would cover far greater traffic ranges. The algorithm would learn over moving patterns of a vehicle. Sliding window technique with fixed window size was also used. The works of [5], [6] and [7] also investigated various alternative techniques related to our work.

In some countries, for example, as in the studies of [8] and [9], found out that the main parameters used to define traffic congestion levels are time, speed, volume, service level, and the cycles of traffic signals at which the motorists have to wait. Our work focused only on the interpretation of vehicle velocity since our work aimed to determine the congestion levels with a minimal set of parameters. Other physical road conditions, such as the number of lanes, obstacles on the road, etc., were not taken into account in this experiment. We assumed that the vehicle velocity was already a product of these factors and can be collected by almost all types of sensors. This made the model easier, broader and more versatile to be used. The congestion levels that we studied were limited to three levels: light, heavy and jam, which was sufficient and appropriate according to the study of [10]. After we successfully derived the congestion classification model, the GPS data were planned to be collected through mobile phones attached to or embedded with GPS device. The data would be sent through the existing data network, such as GPRS, EDGE, and so on. The next section described the methodology of the research.

III. METHODOLOGY

A. Collection of Empirical Data

The traffic data were collected from several highly congested roads in Bangkok, e.g., Sukhumvit, Silom, and Sathorn. A notebook connected to a USB GPS device was used to collect date, time, latitude, longitude, and vehicle velocity from GPS's GPRMC sentences. We captured images of road traffic conditions by a video camera mounted on a test vehicle's dash board. Our vehicle passed through overcrowded urban areas approximately 30 kilometers within 3 hours. In our experiment, we gathered the congestion levels from 11 subjects with as much as 10 years of driving experience each. They watched a 3-hour video clip of road survey and rated the congestion levels into three levels: light, heavy, and jam, which were represented by the scales 3, 2 and 1 respectively. The judged congestion levels were then synchronized with the velocity collected by the GPS device. The voted congestion ratings for a particular time were sometimes highly diverse. Therefore the concluded congestion levels from 11 subjects were calculated using a simple majority vote. It also makes sense that the congestion levels exhibit nominal/ordinal characteristics rather than interval/ratio scale characteristics. From the data gathered, we observed that the vehicle velocities fluctuated wildly and were also non-uniform. The fluctuations of the instantaneous vehicle velocities made it difficult to extract consistent features of the vehicle moving

patterns. To alleviate this oscillation, the traffic data was treated before feeding into a learning algorithm, i.e., decision tree model, which will be explained in detail in the next section.

B. Data Preparation

We minimized a set of attributes by concentrating only on the vehicle velocity and the moving pattern of a vehicle, from which we can infer different levels of congestion. Then, we applied three steps to prepare the data: 1) smoothening out instantaneous velocity, 2) extracting moving patterns of a vehicle using sliding windows technique, and 3) balancing the distribution of sampling data on each congestion level. Next, we will explain each procedure in details.

1) Smoothening Out Instantaneous Velocity

Instantaneous vehicle velocity from the GPS data usually fluctuated widely, as shown in Fig. 1. The dotted and the thick line represent the instantaneous velocity and moving average velocity respectively. This fluctuation made it difficult for the learning algorithm to determine the pattern and classify the congestion level, as in [11], [13], and [14]. Therefore, we needed to smoothen out the fluctuation of instantaneous velocity. We applied a moving average algorithm by averaging the previous ζ instantaneous velocities which we will refer to as the resolution of moving average. The moving average equation is shown in Eq. 1. MV_t represents the moving average velocity at time t . In our experiment, ζ was set to 3. The optimization of ζ will be discussed in details in Section IV.

$$MV_t = \frac{\sum_{i=t-\zeta}^{t-1} V_i}{\zeta} \quad (1)$$

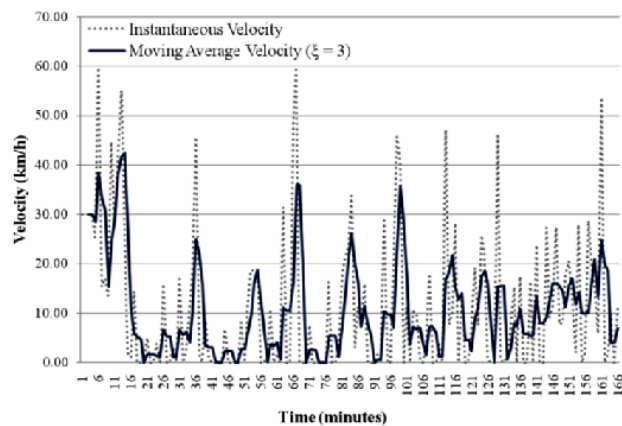


Fig. 1 Instantaneous velocity vs. moving average velocity ($\zeta=3$)

2) Extracting Vehicle's Moving Patterns

When the instantaneous velocity fluctuated less from the smoothening algorithm, it was easier to investigate vehicle's moving patterns. We successfully extracted moving patterns that were practical to be efficiently learned by the learning algorithm, which can be explained as follows. Our previous work suggested that we can use velocity to estimate congestion levels. However, we cannot say that only a single value of a vehicle's velocity at a moment can be used to

accurately determine the congestion level. In a real driving situation, a single value of an instantaneous velocity can be reported at varying congestion levels. For example, a vehicle needs to slow down for turning or stopping for a traffic light. In this condition, the traffic might be light but the velocity is relatively low, as per [13] and [14]. After carefully investigating the data, we successfully mimicked humans' judgements on congestion levels based on moving patterns of a vehicle which was derived from the historical data. Sliding windows, a technique that could satisfy such moving pattern extraction, was employed. We applied fixed sliding windows of size δ to capture moving patterns of a vehicle from the vehicle velocity. In our experiment, δ was set to 3, which means that we captured the moving patterns by a set of three consecutive moving average velocities. The moving pattern at time t with δ equals to 3 includes three consecutive samples of moving average velocity at time t (MV_t), and two priori moving average velocities at time $t-1$ (MV_{t-1}), and $t-2$ (MV_{t-2}). The optimization of δ will be discussed in Section IV. We also introduced a new attribute to represent the average velocity of each sliding window (each moving pattern), called AMV_t .

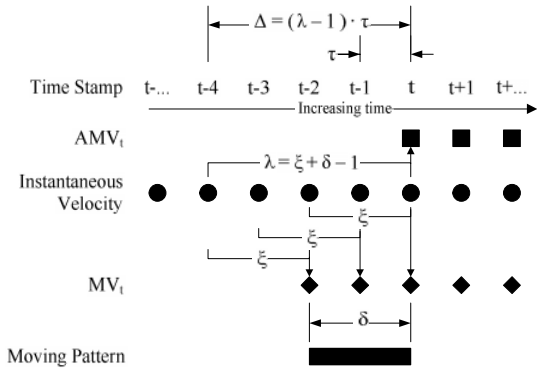


Fig. 2 The visualization of parameter relations

$$\lambda = \xi + \delta - 1 \quad (2)$$

$$\begin{aligned} \Delta &= (\lambda - 1) \cdot \tau \\ &= (\xi + \delta - 2) \cdot \tau \end{aligned} \quad (3)$$

Fig. 2 illustrates how to calculate the moving average at time t from instantaneous velocity, and how to extract the vehicle's moving patterns. Let λ represent the number of continuous instantaneous velocities needed to capture a moving pattern. Let Δ represent the time span of those set of instantaneous velocities. Thus, λ and Δ can be calculated by the value of δ and ξ as shown in Eq. 2 and Eq. 3 respectively. The value of τ is the time interval for capturing vehicle's moving patterns. In our experiment, τ was set to 1 minute. In Eq. 3, for $\xi=3$ and $\delta=3$, the number of data needed to form a sliding window (λ) was 5, representing the number of instantaneous velocities needed to calculate the AMV_t . Thus, the time span needed to extract moving patterns (Δ) was 4 minutes, representing the time span of each sliding window. The values of δ and ξ can be varied so that the moving patterns extracted that were fed to the learning algorithm were optimized. The optimization process is explained in Section IV.

3) Balancing Class Distributions

In our experiment, we captured vehicle's moving patterns every minute from 13:00 to 15:45. There were then a total of 166 instances in our universal data set. Since the calculations of MV_t and AMV_t depend on the previous cascading calculations, the first four instances ($\lambda-1$) were omitted. Therefore, there were 162 instances: 52 instances were in the class of jam traffic, 74 instances were in the class of heavy traffic, and only 36 instances were in the class of light traffic. Class imbalance may cause inferior accuracy in data mining learners, as [12]. Generally, classification models tend to predict the majority class if class imbalance exists. In this case, the class of heavy traffic was the majority class while the minority classes, the classes of light and jam traffic, were also highly important. Therefore, we needed to balance the class distributions to avoid the problem. By this step, we applied a simple technique to alleviate the problem of class imbalance by applying a technique that was similar to the technique of finding a least common multiple number. The result of class balancing yielded 448 instances with 156 instances on class jam, 148 instances on class heavy, and 144 instances on class light. Then, this data set was used to train the classification model, for which we explain the details in the next section.

C. Data Classification

The preprocessed data set was used to train and evaluate the classification model. Since we prior set δ and ξ to 3, our data set consisted of five attributes. The first three attributes were MV_{3t-2} , MV_{3t-1} , and MV_{3t} , which were three consecutive moving average velocities that represented the moving pattern. The fourth attribute was AMV_{3t} , which was the average velocity of the corresponding moving pattern. The last attribute was Level, which was the congestion level judged by human ratings. We chose the J48 algorithm, a well-known decision tree algorithm in the WEKA version 3.6.1 system, to generate a decision tree model to classify the Level. The goal attribute of the model was set to Level. The test option was set to 10-fold cross-validation. Fig. 3 shows steps of generation and evaluation of the classification model of our experiment. After training of the classification model, the derived decision tree is shown in Fig. 4. The decision tree's root node is AMV_{3t} attribute. This means that the average of the moving average velocity is the most important factor to determine the level of road traffic congestion. The result shows a promising technique of determining congestion with an overall accuracy of 91.29%.

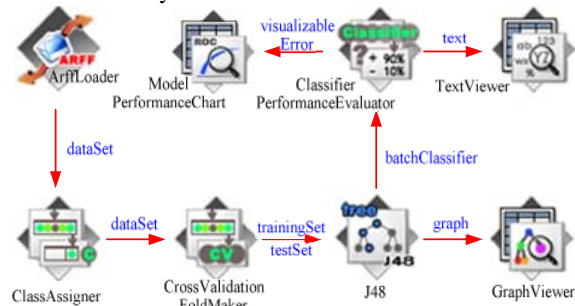


Fig. 3 Knowledge flow of our experiment

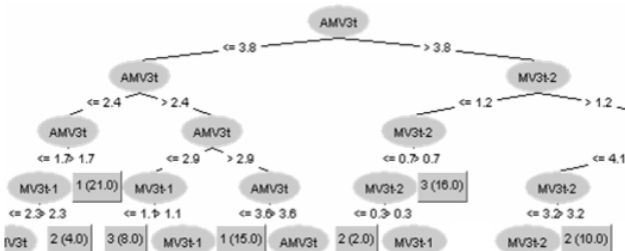


Fig. 4 The derived J48 decision tree ($\delta = 3, \zeta = 3$)

IV. PARAMETERS OPTIMIZATION FOR SLIDING WINDOWS

In the previous experiment, the ζ and δ were set to 3, which provided a promising model with an overall accuracy of 91.29%. The values of the ζ and δ can be varied to obtain a better model with higher accuracy. In this section, we conducted experiments to find the optimal values of ζ and δ . We divided the optimization process into two steps: 1) the optimization of the resolution of moving average (ζ), and 2) the optimization of the window size (δ). Each procedure is elucidated as follows.

1) The optimization of the resolution of moving average (ζ)

In this step, to find out the optimal resolution of moving average or the number of instantaneous velocities used to calculate an instance of the moving average, we repeatedly trained the models in the experiment by fixing the window size (δ) to 3 as the previous experiment and varying the resolutions of moving average. The value of ζ was adjusted from 2 to 8; and then the accuracy of the trained models was used as the measure to decide the best model.

We applied three steps to prepare the data. Firstly, the instantaneous velocities were smoothed to lessen fluctuation and make it easier to extract moving patterns. In this step, the MV_t with $\delta = 3$ and ζ ranging from 2 to 8 were calculated. Secondly, we extracted the vehicle's moving patterns. Three consecutive instances of the moving average of each one minute ($MV_{t-2}, MV_{t-1},$ and MV_t) were captured to make a sliding window of the size 3. AMV_t was computed to represent the average velocity of each sliding window. A moving pattern was extracted by a set of $MV_{t-2}, MV_{t-1}, MV_t,$ and AMV_t .

Thirdly, we balanced the class distribution. Since the calculations of MV_t and AMV_t depended on the previous cascading calculations, some instances ($\lambda-1$ instances) at the beginning of the data set were omitted. In this experiment, the maximum value of ζ was set to 8, requiring 7 instances to be computed, as per Eq. 2. Therefore, we began extracting the vehicle's moving patterns from the ninth minute and omitting the first nine instances. Thus, there were 157 instances, from a total of 166 instances; 52 instances were in the class of jam traffic, 69 instances were in the class of heavy traffic, and only 36 instances were in the class of light traffic. After we balanced the classes, the training data consisted of 631 instances; 208 instances were in the class jam, 207 instances were in the class heavy, and 216 instances were in the class light.

The preprocessed data set was fed into a J48 algorithm. The data set contained five data columns: $MV_{t-2}, MV_{t-1}, MV_t,$

$AMV_t,$ and Level (congestion level). The goal attribute of the model was set to Level. The test option was set to 10-fold cross-validation. The results of the classification models are show in Fig. 5.

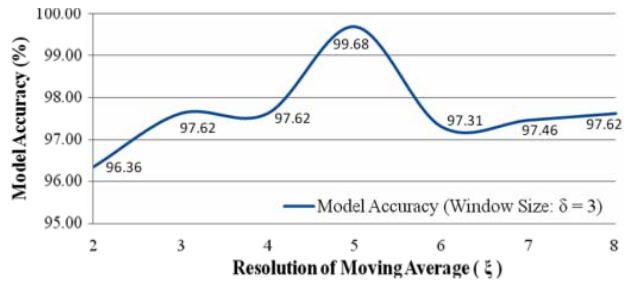


Fig. 5 The model accuracy ($\delta = 3, 2 \leq \zeta \leq 8$)

In Fig. 5, for the resolutions of moving average lower than 5, the models' accuracy tends to increase when the value of ζ increases. The model accuracy reaches the highest level when ζ is set to 5, and then falls off. This may occur when the data is over smoothing. Fig. 6 shows comparisons of the moving average velocity on various resolutions.

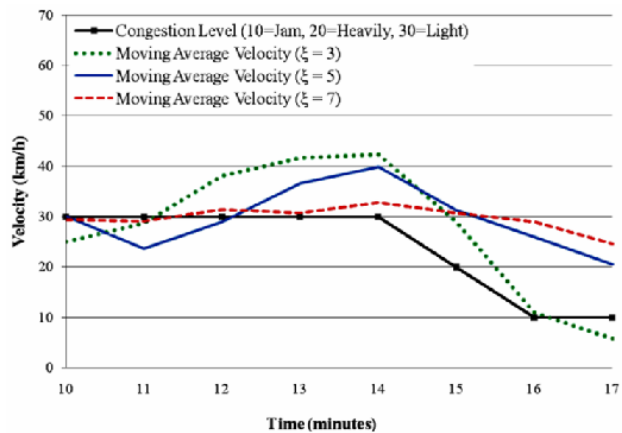


Fig. 6 Moving average velocity on various resolutions

Fig. 6 shows that the moving average velocity with $\zeta = 7,$ is over smoothing, while the moving average velocity with $\zeta = 5$ synchronizes to the users' opinions on congestion levels. From the experimental results above, we finally conclude that the optimal resolution of moving average is 5, which yields accuracy as high as 99.68%. It is to be noted that the accuracy of the model with $\zeta = 3$ and $\delta = 3$ was higher than the previous experiment result (99.68% vs. 91.29%). This is because the preprocessed data sets were different. On account of the cascading calculation, the first four instances in the previous experiment and the first nine instances in this experiment were removed. Five instances following the fourth instance were rated as heavy. As mentioned in the previous section, noise may occur in the class heavy; the model can achieve a higher accuracy by removing the noise.

2) The optimization of the windows size (δ)

In this step, to find out the optimal number of moving average instances to form the appropriate window size (δ), we repeatedly trained the models in the experiments by fixing the resolution of moving average (ζ) just derived from the previous step to 5 and varying the windows size (δ). The value of δ was adjusted ranging from 2 to 8; and then the accuracy of each trained model was used as the measure to compare for the best model.

To make it comparable with the results on previous step, the same data smoothening and class balancing technique were used. In the step of extracting vehicle's moving patterns, AMV_t , current MV_t , and a set of consecutive MV_t varying on the window sizes were captured for the model training. Table I shows the sets of attributes used in this experiment.

TABLE I
EXAMPLE SETS OF ATTRIBUTES OF EACH WINDOW SIZE

Window Size (δ)	Sets of Selected Attributes
2	$MV_{t-1}, MV_t, AMV_{2t}, \text{Level}$
3	$MV_{t-2}, MV_{t-1}, MV_t, AMV_{3t}, \text{Level}$
4	$MV_{t-3}, MV_{t-2}, MV_{t-1}, MV_t, AMV_{4t}, \text{Level}$
5	$MV_{t-4}, MV_{t-3}, MV_{t-2}, MV_{t-1}, MV_t, AMV_{5t}, \text{Level}$
6	$MV_{t-5}, MV_{t-4}, MV_{t-3}, MV_{t-2}, MV_{t-1}, MV_t, AMV_{6t}, \text{Level}$
7	$MV_{t-6}, MV_{t-5}, MV_{t-4}, MV_{t-3}, MV_{t-2}, MV_{t-1}, MV_t, AMV_{7t}, \text{Level}$
8	$MV_{t-7}, MV_{t-6}, MV_{t-5}, MV_{t-4}, MV_{t-3}, MV_{t-2}, MV_{t-1}, MV_t, AMV_{8t}, \text{Level}$

After successfully extracting the vehicle's moving patterns, those sets of attributes were fed into a J48 algorithm. Goal attribute of the model was set to Level. The test option was set to 10-fold cross-validation. The results of the classification models are shown in Fig. 7.

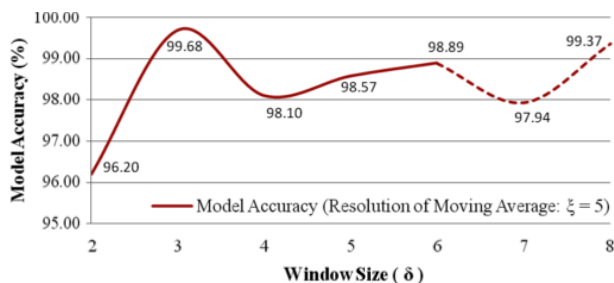


Fig. 7 The model accuracy ($\zeta = 5, 2 \leq \delta \leq 8$)

From the graph showing in Fig. 7, the model accuracy increases when the value of δ increases from 2 to 3. The model reaches the highest accuracy when δ was set to 3. The dotted line extended from the solid line represents the experimental results from which training data set contained missing values. These missing values occurred because of the incomputable of MV_t and AMV_t at the beginning of the data set, requiring at least ζ instances of vehicle velocity. In this experiment, we began extracting the vehicle's moving patterns from the ninth minute, as in the previous experiment, with $\zeta = 5$ and $\delta = 7$ where the AMV_t was unavailable. In this experiment, 11 instances of the instantaneous velocity were required to be computed, as Eq. 2, whereas there were only 10 points of the instantaneous velocities available on the ninth

minute. In this case, AMV_t with $\zeta = 5$ and $\delta = 7$ was treated as missing values and replaced by "?" in the data set. The occurrence of missing values in the training data set can create a bias on the decision tree model. Therefore, we finally concluded that the optimal window size is 3 which yielded accuracy as high as 99.68%.

V. RESULTS AND EVALUATIONS

A. Classification Model

By the two steps of the optimization, we successfully found that the optimal resolution of moving average (ζ) is 5 and the optimal window size (δ) is 3. The derived decision tree model achieved an overall accuracy as high as 99.68%. The optimized decision tree model is visualized in Fig. 8. Size of our decision tree is 107 nodes, 54 of which are leaf nodes. The time taken to build the model was approximately 0.09 seconds. Decision tree's root node is the MV_t attribute. It means that the *moving average velocity is the most important factor* to determine the level of road traffic congestion. AMV_t is also another highly important factor to determine the level of road traffic congestion as shown in the previous and the optimized decision tree.

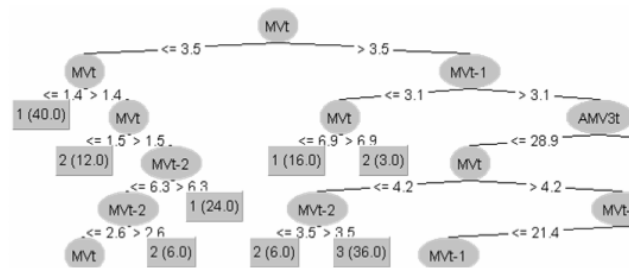


Fig. 8 The optimized J48 decision tree ($\delta = 3, \zeta = 5$)

B. Performance Evaluations

The optimized decision tree model achieved an overall accuracy of 99.68% with a root mean square error of 0.0368, and a precision and a true positive rate on each class ranging from 0.990 to 1.000, which was very high. False positive rate (FP Rate) ranged from 0.000 to 0.005, which was very low indicating a good model. Table II shows the optimized classifier's performance for each class in detail.

TABLE II
THE OPTIMIZED CLASSIFIER'S PERFORMANCE

Class	TP Rate	FP Rate	Precision
Jam (1)	1	0.005	0.99
Heavy (2)	0.99	0	1
Light (3)	1	0	1
Average	0.997	0.002	0.997

From Table II, the highest TP Rate is 1.000 on the class Jam and Light. It means that when the road traffic congestion level is light or jam, our optimized classifier correctly classify the traffic will 100%. The lowest TP Rate is 0.99 in the Heavy class. It can be interpreted that when the road traffic congestion level is heavy, our classifier will correctly classify the traffic 99%. The result of the optimized model evaluation is showed by a confusion matrix in Table III.

TABLE III
THE CONFUSION MATRIX OF THE OPTIMIZED MODEL

		Predicted Congestion Level		
		Jam	Heavy	Light
Instances Congestion Level	Jam	208	0	0
	Heavy	2	205	0
	Light	0	0	216

The number 2 in the confusion matrix, as per Table III, is the result of misclassification on the heavy traffic class, representing the only 2 instances of heavy class which the optimized model misclassified as jam traffic.

VI. CONCLUSION

In this study, we investigated an alternative technique to automatically classify the road traffic congestion levels that was highly consistent with road users' judgments. The technique minimally required data from GPS devices, which can be collected from participants through mobile data networks. Vehicle velocity can be used to determine the congestion level but the instantaneous velocity fluctuated widely. We smoothened out the oscillated instantaneous velocity by averaging it with historical velocities, which was called moving average velocity. We applied a sliding windows technique to capture the consecutive moving average velocities, which was called a moving pattern. We derived a new attribute, AMV3t, which represents the average velocity of the corresponding moving pattern. Parameters δ and ζ were set to 3. The moving patterns were captured every minute. Then road users' judgments and related information were learned utilizing a decision tree model (J48). The evaluations revealed that the decision tree model achieved an overall accuracy as high as 91.29% with a precision as high as 96.6%. The root mean square error was only 0.2171.

The model was optimized. The experiments revealed that the optimal resolution of moving average (ζ) was 5 and the optimal window size (δ) was 3. The evaluations revealed that the optimized decision tree model achieved an overall accuracy of 99.68% with a precision as high as 100%. The root mean square error was only 0.0368.

There are several opportunities for future research. For example, evaluations of the model against other data sets from various road types or vehicles could minimize the chance of the over fitting problem. Moreover, the model can be implemented in the real world. We plan to integrate such a model into the existing ITS system in Bangkok. The technique will also be extended so as to cover the whole country if possible.

REFERENCES

[1] S. Phoosuphanusorn, "New mobile-phone users up 30%," Bangkok Post, May 2007.
 [2] W. Pattara-atikom and R. Peachavanish, "Estimating Road Traffic Congestion from Cell Dwell Time using Neural Network", the 7th International Conference on ITS Telecommunications (ITST 2007), Sophia Antipolis, France, June 2007.

[3] P. Pongpaibool, P. Tangamchit, K. Noodwong, "Evaluation of Road Traffic Congestion Using Fuzzy Techniques," Proceeding of IEEE TENCON 2007, Taipei, Taiwan, October 2007.
 [4] F. Porikli and X. Li, "Traffic congestion estimation using hmm models without vehicle tracking" in IEEE Intelligent Vehicles Symposium, June 2004, pp. 188-193.
 [5] J. Lu and L. Cao, "Congestion evaluation from traffic flow information based on fuzzy logic" in IEEE Intelligent Transportation Systems, Vol. 1, 2003, pp. 50-33.
 [6] B. Krause and C. von Altrock, "Intelligent highway by fuzzy logic: Congestion detection and traffic control on multi-lane roads with variable road signs" in 5th International Conference on Fuzzy Systems, vol. 3, September 1996, pp. 1832-1837.
 [7] R. B. A. Alessandri and M. Repetto, "Estimating of freeway traffic variables using information from mobile phones," in IEEE American Control Conference, 2003.
 [8] J. T. Lomax, S. M. Tuner, G. Shunk, H.S. Levinson, R. H. Pratt, P. N. Bay and B. B. Douglas, "Quantifying Congestion: Final Report" National Cooperative Highway Research Program Report 398, TRB, Washington D.C., 1997.
 [9] R. L. Bertini, 2004. Congestion and Its Extent. "Access to Destinations: Rethinking the Transportation Future of our Region", Minnesota, U.S.A.
 [10] K. Choocharukul, "Congestion Measures in Thailand: State of the Practice." Proceedings of the 10th National Convention on Civil Engineering, May 2005, pp. TRP111-TRP118.
 [11] W. Pattara-atikom, P. Pongpaibool, and S. Thajchayapong, "Estimating Road Traffic Congestion using Vehicle Velocity", Proceedings of the 6th International Conference on ITS Telecommunications, Chengdu, CHINA, June 2006, pp. 1001-1004.
 [12] D. J. Drown, T. M. Khoshgoftaar, and R. Narayanan, "Using Evolutionary Sampling to Mine Imbalanced Data", Proceedings of the 6th International Conference on Machine Learning and Applications (ICMLA 2007), OH, USA, December 2007, pp. 363-368.
 [13] T. Thianniwet, S. Phosaard, and W. Pattara-Atikom, "Classification of Road Traffic Congestion Levels from GPS Data using a Decision Tree Algorithm and Sliding Windows" in Proc. of the World Congress on Engineering (WCE 2009), vol. I, London, U.K., 2009, pp. 105-109.
 [14] T. Thianniwet, S. Phosaard, and W. Pattara-Atikom, "Classification of Road Traffic Congestion Levels from Vehicle's Moving Patterns: A Comparison between Artificial Neural Network and Decision Tree Algorithm" in Electronic Engineering and Computing Technology: Lecture Notes in Electrical Engineering, vol. 60, S.-L. AO and L. Gelman, Eds. Netherlands: Springer, 2010, pp. 261-271.