Detecting Urban Traffic Congestion with Single Vehicle

Chenqi Wang and Hsin-Mu Tsai
Intel-NTU Connected Context Computing Center and
Department of Computer Science and Information Engineering
National Taiwan University
Taipei 10617, Taiwan
Email: {r01922140,hsinmu}@csie.ntu.edu.tw

Abstract—Traffic congestion in urban areas is a severe problem in many cities around the world. Conventional infrastructure-based solutions to detect traffic congestion, such as surveillance cameras and road surface inductive loops, have the limitations of high deployment costs and limited coverage. In recent years, due to the popularity of mobile devices, solutions that do not require pre-deployed infrastructure start to emerge; in these solutions, sensor data is collected by mobile devices onboard the vehicles, sent to a central server via vehicle-to-infrastructure (V2I) or cellular communications, and used collectively to determine the traffic states of the roads. However, existing solutions require data from a considerably large number of vehicles on the same road to accurately detect traffic congestion of a particular road. In this paper, we propose a novel approach to detect the traffic states of the roads with only the data from a single vehicle. The biggest advantage of such an approach is that, unlike previously proposed solutions, the system can function properly even if there is only a smaller number of vehicles equipped with the system, which is usually the case at the early stage of the deployment of a vehicle-to-vehicle (V2V) network or a large-scale intelligent transportation system. In our solution, machine learning mechanisms are utilized to classify the traffic state by extracting the movement behaviors of a vehicle. Our model development and performance evaluation utilize highly accurate vehicle traces collected at several real-world intersections with lidar. In addition, to properly label the obtained data traces to either congested or free-flow and accurately reflect the reality, a previously proposed theoretical method is used in combination with human labeling. Evaluation shows that our approach can achieve a detection accuracy of 88.94%.

I. INTRODUCTION

Statistics show that there has been a huge increase in the percentage of global population living within cities since 1950, and this trend shows no sign of slowing down. Traffic congestion is one of the many problems created by the high population density in urban areas of a city. A recent investigation [1] shows that on average a U.S. commuter wastes 52 hours, 24 gallons of fuel, or 1,128 U.S. dollars per year due to traffic congestion. Traffic congestion has become one of the biggest economic problems that exist in every major city in the world, and urgently needs to be solved.

To solve the problem, one of the main tasks is to be able to accurately detect traffic congestion in real time. A recent study [2] shows that the state of a given road does not change drastically. That is, when a road is in congested state, even if the input of a road suddenly drops, the inertia of the existing queue causes that road segment to remain congested for an additional short period of time. Thus, if a traffic congestion can be detected and the information can be broadcasted to nearby vehicles that have not yet entered the congested road segment, avoiding that congestion is possible - by using a different route.

Figure 1 illustrates a common example. When a traffic congestion is detected by vehicle A, it can notify relevant nearby vehicles, either by broadcasting a message to vehicles within its communication range via V2V communications (e.g., vehicle C) or sending the information to the backend server via V2I communications (i.e., server D), which would then relay the information to vehicles that are travelling toward the congested road segment via V2I communications so that they can use an alternative route to avoid the congestion (e.g., vehicle B).

This is especially feasible in urban areas, as there exist more alternative routes between a pair of source and destination locations. It is also worth noting that, in urban areas, many factors influence the traffic state simultaneously and the traffic state changes frequently; as a result, it is not feasible to predict the traffic state based solely on the historic data and the current time of the day and/or day of the week. A more robust mechanism is required. Moreover, by directing the vehicles to avoid congested road segments, the traffic congestion could be alleviated and eliminated in less time.

Three main categories of approaches to detect the traffic
state have emerged. The first one deploys infrastructure sensors such as magnetic inductive loop or cameras to detect the congestion [3], [4]. Unfortunately, due to the high deployment and maintenance costs of these devices, the coverage of these systems is usually limited to only the busiest intersections and road segments in a city. The second one [5]–[8] is more pervasive and utilizes the idea of gathering the sensor information from probe vehicles. A central server collects the information from probe vehicles and use the information to determine the traffic state. The estimated traffic state is then delivered to the users. The approach eliminates the drawbacks of the infrastructure-based solution, but requires a high penetration rate, i.e., a large number of probe vehicles, to guarantee real-time performance. This assumption is not practical for a system that is in its early stage, during which only a smaller number of vehicles are equipped with the capability to collect sensor information. Moreover, deployment and maintenance of the central servers could induce a huge overhead when the number of users becomes large. The last approach [9]–[12] uses a similar idea, while the only difference is to use a Vehicle-to-Vehicle (V2V) network instead of the Client/Server (C/S) paradigm - vehicles collect sensor information from neighboring vehicles and determine the traffic state by themselves. The advantages include the larger coverage due to the distribution and the high mobility of the vehicles and the elimination of the need to have a central server. However, it does not resolve the problem of requiring a high penetration rate, while it raises a few additional issues such as privacy problems.

Recently, a crowdsourcing mobile GPS navigation application, Waze [13], was developed to take advantage of real-time user-specified information. One of its features is to let the users actively report road segments with traffic congestion as they travel through them. In return, the application can obtain traffic congestion information at other locations, reported by other users, and hence route planning in this application can take the information into consideration. The crowdsourcing approach takes advantage of human better ability to judge whether there is a traffic congestion based on his/her own perception, and could provide higher detection accuracy. However, it also poses a few new challenges, such as how to prevent malicious persons from providing misleading information for their own advantages, in particular, to claim the roads that they would pass by are congested so that others would avoid using that road.

If the detection task can be delegated to the machine (a mobile device onboard the vehicle), maybe some of these problems can be mitigated. Based on this observation, we present a simple yet pragmatic approach by using a single vehicle to detect traffic congestion with only velocity information that can be obtained from GPS. Specifically, we extract the movement pattern features and use machine learning algorithms to classify the traffic state. The rationale of choosing the moving patterns as features is from the experience of traveling in cars - it becomes uncomfortable or even causes car sick in the back seat when the car is in terrible traffic - actually, even when you close your eyes, you can still realize the congested state outside by feeling the stop-and-go movement, which is a strong discriminating feature that could be extracted from the collected data. The biggest benefit of this feature is that it is independent of external factors such as road structure, the type of the vehicle, the duration of traffic light states, etc. The following summarizes the advantages of the proposed approach:

1) No infrastructure or central server is needed - the costs of deployment and maintenance are eliminated.
2) No communication of any kind is needed in the detection stage, since our model takes data from a single vehicle as input. As a result, there is no new security or privacy issues caused by our proposed approach. In practice, any device that uses our proposed classifier can predict the current traffic state by itself with only velocity information from GPS.
3) The proposed detection mechanism has no requirement of market penetration rate.

The contribution of this work includes the following:

1) The developed congestion detection mechanism is sufficiently general to be utilized in many devices, including all types of vehicles and road-side infrastructures to monitor the traffic state.
2) Our classifiers in this paper are trained and evaluated with real-world vehicle traces collected at multiple intersections by lidar, which is publicly available for the research community. The highly accurate data provides more fidelity to our evaluation results.
3) To collect the vehicle traces, we developed a novel approach to collect vehicle traces even from a congested road segment, where the velocity and the location of all vehicles in a particular lane can all be recorded.
4) The applications enabled by the proposed congestion detection mechanism can serve as a bootstrapping application to provide incentive for early adopters to purchase vehicles with V2V technologies, as the application can function even if there are only a smaller number of vehicles equipped with the system.

The rest of the paper is organized as follows. We present the design of our detection algorithm and experimental setup in Section II. Then our model is evaluated with experiments presented in Section III. Related works are discussed in Section IV. Finally, Section V concludes this work and illustrates directions for future work.

II. SYSTEM DESIGN AND EXPERIMENTAL SETUP

To enable vehicles to determine the current traffic state accurately similar to human judgement, we utilize machine learning algorithms to derive the model. The goal of the model is to be able to classify the road segments where the vehicle is currently traveling to be either a congested or a free-flow (non-congested) road segment. In this section, we will present the detailed design of our detection algorithm and how we derive the classification model.

A. Feature Selection

To use machine learning method, we need to extract features that contain information of the traffic state. Traditional features that have been used in existing literature include average velocity and traversal time. However, these features are often strongly influenced by factors that are not related to the traffic state of a road segment, such as the road capacity,
the speed limit, the type of vehicle, and the red light duration \cite{14}, etc. If these features are used, different vehicles in different roads could derive significantly different models, i.e., boundaries for classification of congestion and non-congestion in the feature space are different \cite{2} for different vehicles. In this case, generalization of the developed model to vehicles of different types or in different conditions is difficult.

Figure 2 shows the distribution of average velocity of a car traveling on a road, from both congested and free-flow road segments. One can observe that a significant number of vehicles with low average velocity do not correspond to a congested case; it is hard to derive a threshold to distinguish congested and free-flow road segments. Certainly, other feature such as traffic volume seems to be an ideal feature. However, the information cannot be obtained from a single vehicle, and is only suitable for infrastructure-based solution.

Figure 3 presents a velocity-distance graph of vehicle traces collected from the same road segment in two different traffic states. One can observe that, when traveling on a congested road segment, vehicles change the velocity frequently; when traveling on a free-flow road segment, vehicles tend to travel with a steady velocity and only stop at red lights. The steady velocity, however, varies with different vehicles. In our proposed algorithm, instead of using average velocity as the main feature, we attempt to capture the distinctive velocity change in the traces for traffic state classification. For example, there are vehicles travelling with steady velocities of 25 m/s and 10 m/s in the right figure; though there is a large difference in the actual velocity, yet the vehicles exhibit similar moving pattern. Note that this is independent of many aforementioned factors that is unrelated to the traffic state, and thus models derived with features that can capture the characteristics of the vehicle moving pattern can be generalized to detect congestion in different roads or with different vehicles.

Then, the problem becomes how to select a set of features that can best represent the moving pattern. In our algorithm, we utilize the simple concept of sampling in signal processing to convert the vehicle velocity trace to a sequence of samples, and each sample will be used as one feature of an instance in the machine learning algorithm.

Figure 4 shows an example of the process of feature extraction from the vehicle velocity trace. There are two main parameters, sampling length \( \lambda \) and sampling interval \( \delta \), that determine the dimension and attribute value of the feature vector. Specifically, we split the trace into small segments of length \( \delta \) and calculate the average velocity \( x_i \) of each segment. A set of consecutive \( \lambda \) samples constitute the final feature vector \( \mathbf{x} = [x_{i+1}, x_{i+2}, \ldots, x_{i+j}, \ldots, x_{i+\lambda}] \). As in signal processing, the smaller sampling interval \( \delta \) is, the higher fidelity the features can achieve to depict the original signal, and, in this case, the vehicle moving pattern.

There are, however, practical considerations when selecting the value of \( \delta \) in our study. Since our data is collected with lidar (please find the details in subsection II-B), the location of a car will fluctuate unpredictably because a 5-meter long car is supposed to be represented by a zero-length point in each timestamp and, in reality, not all the points of a car can reflect light back to lidar. Regardless of the point we choose to represent the location of the car (in our study, the most frontal point of the car is chosen to represent the location of the car since we found this part has a higher probability to reflect the light from lidar), it may vary as the vehicle moves forward,
which could lead to inaccurate measurement of velocity in each $\delta$. Certainly, this error is very small, but if the value of $\delta$ is chosen to be very small, the error in velocity could be influential. In this paper, the value of $\delta$ is chosen to be 5 meters, which is the average length of a car. We believe that this choice provides ideal balance between movement fidelity and quantization accuracy.

As to $\lambda$, it is intuitive that the larger value of $\lambda$ we choose, the more representative information we can obtain, and in turn, the better performance can be achieved (this is verified by the experimental results in Section III). However, all road segments have limited length. If a road segment has a length of less than $\lambda\delta$, then the feature vector contains information contributed by other road segments and causes erroneous detection results. Therefore, a smaller value of $\lambda$ can be used in shorter road segments. Additionally, in the same road segment, the smaller value of $\lambda$ we select, the less time it will take to collect the required feature vector. Thus, the system can start predicting traffic state more quickly and it is more likely that the detection message can be sent in time. Moreover, from the machine learning’s point of view, to any given vehicle velocity trace of a fixed length (e.g., in time. Moreover, from the machine learning’s point of view, to any given vehicle velocity trace of a fixed length (e.g., in this paper, all collected traces are 70-75 meters long), we can derive more instances (feature vectors) in a cascading way (e.g., $[x_1, x_2, x_3], [x_2, x_3, x_4]$, etc.) for both training and testing with a smaller $\lambda$, which can make the results more convinced. And the sampled instances would have wider distribution of locations in the whole road segment, thus the trained model can be more generalized and robust. Experimental results with different $\lambda$ is elaborated on in Section III.

B. Data Collection

To obtain a robust and generalized model with machine learning algorithms, the data used to derive the model is extremely important. In our work, obtaining a data set with accurate vehicle velocity traces, both when the road segment is in congested state and in free-flow state is not trivial. Among all existing methods for data collection, the approach to use image processing techniques to process images obtained from cameras mounted at intersections seems to be the most effective method. However, when in congested state, with such an approach it is very difficult to distinguish the vehicles and to track the trajectories of vehicles correctly and accurately. Obstruction caused by neighboring vehicles in a congested state also create significant problems for tracking a vehicle. Inductive loops [15] is an alternative approach to collect the vehicle velocity traces. Though the deployment cost is high and the data is usually not available to the research community. Additionally, since the distance between two consecutive inductive loops is usually at least 30 meters, the obtained data does not have the required spatial resolution. Yet another alternative is to collect the trace from vehicles with onboard mobile devices equipped with GPS. However, with this approach it is very time consuming to obtain a large data set. The errors caused by weak GPS signal reception also causes non-negligible noises in the obtained data set.

Instead, we present a novel yet effective method to collect abundant, realistic, and accurate traces by using lidar - a laser ranging device with centimetre accuracy. The largest challenge of using lidar is that its light can be obstructed by nearby objects, similar to the case when using a camera. To avoid obstruction caused by nearby vehicles, we develop an innovative method by placing the lidar on the pedestrian overpass and scan the stream of passing vehicles in a single lane longitudinally, as shown in Figure 5. Due to limited space, the detailed description of the steps to processing the data obtained from lidar is not mentioned in this paper (videos of the captured traces processed with our algorithm can be found at http://mvnl.csie.ntu.edu.tw/DataCollection). It is worth mentioning that the range of the lidar used in this study is 80 meters, and thus the range that vehicles can be tracked is approximately 75 meters in our work, which cannot cover a road segment longer than that. Therefore, our study only focuses on the 80-meter road segment before the intersection, where we believe is the part of the road that is the most representative in the entire road, since the traffic signal has most influence to traffic on this part of the road. In real-world implementations, systems that use our classifier can generate the same 80-meter traces before the intersection to predict road state. On the other hand, since with our configuration the 2-dimensional lidar we used can only scan a vertical line, if a vehicle deflects from this line, it would disappear. Thus, not all the vehicles on that lane can be captured. However, we argue that the captured traces still have a high precision and authenticity, which can be verified by comparing the video taken by a camera mounted next to the lidar and the traces extracted by our codes. Particularly, the longer the captured trace is, the more accurate the trace will be - owing to the less probability of capturing other interferences such as scooters or pedestrians. When the trace is longer than 70 meters, the precision is close to 100%. To guarantee this accuracy, we only use labeled traces longer than 70 meters in a training process to obtain the models.

![Figure 5. Our data collection method with lidar positioned on a pedestrian overpass. The lidar is mounted on a tripod and scan the vehicles in a single lane longitudinally.](image)
C. Classification Model

After collecting the trace data, we need to construct final feature vectors for learning stage. However, there exists an important problem - how to label the instances - which is often ignored. Most related work [14], [16], [17] labels the data manually. We argue that this could be too subjective since different people may have different ideas on what kind of road condition corresponds to a traffic congestion. In this paper, instead of relying solely on human judgement, we utilized the definition of congestion proposed by Marfia et al in [2]. The main theory of the definition is that a road can be regarded as congested only when the likelihood of finding it in the same congested state is high in the near future. In simple terms, if most vehicles (80% is used in [2]) suffer a long traversing time over a period of time in a road, the road is labeled as congested in that duration. [2] also carried out extensive real-world experiments to validate this definition. In our study, we use both this approach and human judgement based on the taken video to label the data - only an instance that is considered as congested/free-flow with both approaches are labeled as congested/free-flow and included in the final data set.

Additionally, as described in subsection II-A, a trace with \(l\) samples can be split into \(\frac{1}{2} - \lambda + 1\) instances of feature vectors. Thus, if we have \(N\) traces, we can obtain \(N(\frac{1}{2} - \lambda + 1)\) instances for learning. Then, we use 10-fold cross validation to evaluate the performance of our algorithm/model and avoid over-fitting. However, there exists a hidden machine learning problem here. That is, the instances extracted from the same trace can be expected to share some characteristics, resulted from the same vehicle and the same road segment. In particular, the adjacent two instances even have some identical features while only the orders in the feature vectors are different. In other words, these instances are more ‘familiar’ with each other. Therefore, if such instances appear in both the training and validation data sets, the obtained prediction accuracy numbers could be too optimistic. To avoid the problem, we utilized a simple yet effective mechanism by bundling the instances from the same traces in the cross validation process. Then, all such instances exist either in the training set or the validation set as a whole.

To train the classification models with the obtained data, we explore three state-of-art algorithms, Random Forests [18], Adaboost [19] and support vector machine (SVM). These algorithms can be implemented with WEKA [20] and LIBSVM [21]. Among the algorithms, Adaboost is a meta-algorithm, and we used decision stump as its base algorithm. One of its great advantages is its robustness to over-fitting. SVM is another commonly used algorithm, which attempts to find a hyperplane to classify the instances. Its performance is often dependent on the choice of the kernel and the parameters. Among the four kernels, RBF kernel can handle cases where the class labels and attributes are related in a non-linear manner, which seems to be most applicable in our work and thus chosen as the default kernel. In addition, to derive the optimal but not over-fitting parameters, we use a subset to run cross validation for parameter selections. The last candidate is Random Forests - a tree based learning method which constructs multiple decision trees using randomly sub-sampled features and output the final decision value by averaging the prediction of all the trees, which can reduce the variance of prediction. Thus Random Forests are robust and powerful and, as the result in Section III shows, it achieves the best performance out of the three. The other two algorithms also achieve sufficiently high accuracy, which can verify that our chosen features contain sufficient information to determine the traffic state.

III. EXPERIMENT AND EVALUATION

In this section, we show the detail of experiments carried out in two different roads and the performance evaluation of our work. Specifically, we collected two trace sets, Set A and Set B, in Keelung Road and Xinyi Road in Taipei, respectively. As shown in Figure 5, we collected the data at the intersections. And due to the limitation of the range of lidar and the height of the overpass, the actual range of road where the traces are collected is approximately the 75-meter segment before the intersection. To decrease the chance of capturing other interferences such as scooters or pedestrians, only vehicle traces that is longer than 70 meters is retained for analysis. In this way, we collected Set A with 84 positive (congested) and 141 negative (free-flow) traces and Set B with 67 positive and 243 negative traces - 535 traces in total. All the instances are labeled as subsection II-C depicts. Our data set is slightly class-imbalanced, but it does reflect the real-world situations - generally speaking, the number of cars in free-flow is larger than that in congestion during a day. Therefore, although we know unbalanced data may lead to inferior accuracy [22], we retain the whole data set to maintain the integrity and authenticity of our collected data set. In total, we carry out four experiments, including the following:

1) The first one is to evaluate the performance of our models. This experiment compares the results of the three learning algorithms with different choices of parameters and is used to select a reasonable algorithm as the default for the rest of our work.

2) To evaluate the feasibility of generalizing the obtained model to be utilized for data obtained from other roads, we evaluate three different modes, namely internal, combined, and external modes.

3) Since a trace can be split into several sub feature vectors, the smaller \(\lambda\) we set, the more opportunities we can have in a road (assume the road is longer than 80 meters) to predict the result. This experiment is implemented to see if more opportunities lead to better performance.

4) All data is collected by lidar, which has a high accuracy. However, in reality, we usually use GPS to obtain the information, and its precision may have an error of 5-10 meters. We carried out this experiment to investigate the impact of the GPS error to the detection accuracy.

A. Algorithm Selection

As discussed in subsection II-C, there are three learning candidates in our work, thus we need to choose an optimal one as our final classifier. Additionally, the values of the two main parameters in our model, \(\lambda\) and \(\delta\), needs to be determined. As illustrated in subsection II-A, we choose \(\delta\) to be 5 meters. Then we compare detection performance when different choices of
\(\lambda\) are used with the three learning algorithms. To obtain a fair and accurate evaluation, we performed 10-fold cross validation by using the combination of Set A and Set B, called set C, as the evaluation data set and used the same parameters for the three algorithms - that is, RandomForests with number of trees set as 500, AdaBoost with Decision Stump as base algorithm and 500 iterations, SVM with RBF kernel and \((C, \gamma)\) set as \((1, 4)\). Table I shows the average accuracy of the three learning algorithms with different values of \(\lambda\).

<table>
<thead>
<tr>
<th>Value of (\lambda)</th>
<th># total data</th>
<th># folds in CV</th>
<th>RandomForests</th>
<th>AdaBoost</th>
<th>SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7862</td>
<td>10</td>
<td>74.71%</td>
<td>79.21%</td>
<td>67.68%</td>
</tr>
<tr>
<td>3</td>
<td>6792</td>
<td>10</td>
<td>84.16%</td>
<td>82.49%</td>
<td>79.35%</td>
</tr>
<tr>
<td>5</td>
<td>5722</td>
<td>10</td>
<td>86.35%</td>
<td>84.75%</td>
<td>82.07%</td>
</tr>
<tr>
<td>7</td>
<td>4652</td>
<td>10</td>
<td>87.39%</td>
<td>85.84%</td>
<td>83.59%</td>
</tr>
<tr>
<td>9</td>
<td>3582</td>
<td>10</td>
<td>87.75%</td>
<td>86.84%</td>
<td>85.34%</td>
</tr>
<tr>
<td>11</td>
<td>2572</td>
<td>10</td>
<td>88.00%</td>
<td>87.09%</td>
<td>85.79%</td>
</tr>
<tr>
<td>13</td>
<td>1942</td>
<td>10</td>
<td>89.05%</td>
<td>87.72%</td>
<td>86.87%</td>
</tr>
<tr>
<td>15</td>
<td>382</td>
<td>10</td>
<td>91.59%</td>
<td>89.43%</td>
<td>87.86%</td>
</tr>
</tbody>
</table>

Note that the number of total data is determined by the choice of \(\lambda\) since a fixed-length trace can be split into different number of feature vectors with different \(\lambda\) values. One can observe that, as \(\lambda\) increases, the size of data set decreases, whereas the accuracy shows a steady rising trend. One reasonable interpretation is that with the increase of \(\lambda\) the features vectors can be more representative of the moving pattern thus more discriminative to classify the traffic state, which also prove the reasonableness of our selected features. Additionally, RandomForests generally outperforms the other two methods, and thus is chosen as our default learning algorithm. To illustrate the improvement when using a machine learning based approach compared to a naive thresholding approach, we compare the obtained results with that for an optimal average velocity thresholding approach as a lower bound benchmark. In this alternative approach, we calculate the average velocity of all cars in the same data set and select an optimal velocity threshold that is used to classify the data instance so that the overall error rate is minimized. With this approach the highest obtained accuracy is 75.51%. While the number seems sufficiently high, it should be emphasized that the classes are imbalanced in our data set - even if we classify all the instances to the free-flow class, we can still achieve a 71.78% accuracy. Therefore, we need a more discriminable metric - the receiver operating characteristic (ROC) curve, which compares the performance of the classifier across the entire range of class distributions and error costs. Figure 6 shows the ROC curve of our models with four different values of \(\lambda\) compared with the optimal thresholding method, where each point on the curve represents the average over 10 folds of false positive rate and true positive rate for a fixed threshold. The line marked with triangles denotes the benchmark one. From the figure we can see the area under curve (AUC) of our models is significantly larger than that of the optimal thresholding method. Specifically, when the tolerance of false positive rate surpasses 20%, we can achieve up to 95% true positive rate while the optimal thresholding method has less than 40%. Therefore, our method demonstrates an evident accuracy superiority over the traditional method of using a velocity thresholding approach to determine traffic state.

### B. Evaluation of Generalization

We hope our model can be applicable in different situations with different cars. In this subsection we will evaluate the feasibility for generalization. Since Set A and Set B are collected from different roads, we can derive three evaluation modes, which will be called internal, combined, and external modes, respectively. With internal mode, the training and the testing data are from the same roads - cross validation is performed in Set A or Set B; With combined mode, the training and the testing data are sampled from a combined data set, which includes traces from two different roads - cross validation is performed in set C; external mode completely separates the training data and the testing data, set A is used to train a model to predict data instances in set B, and vice versa. Figure 7 shows the result of the three modes with different choices of \(\lambda\) values, where the results of internal and external modes are the average of the two cases. One can observe that the internal mode has best performance - up to 92.20% on average, while the external one has the lowest performance. This is intuitive since the traces from the same road may share some characteristics and thus can be more discriminable. However, when \(\lambda\) is set to 15, external mode can still achieve an accuracy of 88.94%. Moreover, as long as the \(\lambda\) is more than 7, i.e., a road segment is longer than 35 meters, we can achieve an accuracy higher than 85%. In short, the results show that generalization of our models is possible; it can achieve a sufficiently high accuracy even when the model is used to detect congestion in other roads.

### C. Enhancement of Result

As illustrated above, an \(l\) long trace can be split into \(\frac{l}{3} - \lambda + 1\) sub feature vectors. In other words, we have \(\frac{l}{3} - \lambda + 1\) opportunities to predict the traffic state and thus we can use a voting mechanism to determine the final output. To see if it can help to enhance the performance, we carried out another experiment, where the detection result is generated for each trace instead of for each data instance. In other words, we only evaluate the accuracy of the traces instead of the \(\lambda\)-dimension feature vectors in this experiment. It should be noted that, all trained classifiers remain unchanged; the training process does not need to be performed again. The only difference lies in the detection stage. To determine the class for each trace, we
simply use a voting mechanism - we use the derived classifier to first classify each of the $\lambda$-dimensional sub feature vectors so that we can get $\frac{d}{2} - \lambda + 1$ labels for each trace. Then, the class label with more votes is chosen to be the final selected class. Figure 8 shows the result of this method. Surprisingly, the accuracy is boosted generally, especially when $\lambda$ is low. Although it does not exceed the best performance - the result of $\lambda = 15$, it improves the performance when using different $\lambda$ values to a higher level, which can reduce the variance of detection performance due to different choices of $\lambda$ values. In summary, we believe that the voting mechanism can serve as an add-on to provide robustness to our model and further improve the detection performance.

D. Performance in Reality

Until now, all evaluation results are based on the accurate data set collected by lidar. However, in most real-world scenarios, GPS is usually utilized to obtain the data and GPS has a positioning error of up to 10 meters. Thus, to evaluate the performance in reality, we designed an experiment to introduce artificial errors manually in the testing data set (not in the training set). Specifically, we varied the related error rate from 5% to 100%, and average the results of 10 experiments for each value. Note that we use the original method instead of aforementioned voting mechanism in this experiment since the result of the latter have a larger variance, which may lead to more uncertainty for evaluation. Figure 9 shows the final result of the three modes when $\lambda$ is set to 9 and 15, respectively. One can observe that, even when the related error rate reaches 10% and 20%, our optimal model can still achieve more than 87% and 85% accuracy, respectively. This encouraging result verifies the feasibility of our work in reality, when GPS is used to provide the sensor data.

IV. Related Work

A large amount of works have been proposed for the congestion detection problem. Some are infrastructure-based [3], [4], which are costly. Thus, other probe-based [5]–[8] and V2V-based [9]–[12] methods are proposed. However, most of these works required a large penetration rate, that is, they assumed a considerable proportion of vehicles equipped with their systems, which is not practical currently. Therefore, Peachavanish et al. [16] and Pattara-Atikom et al. [17], as well as Park et al. [23] proposed different methods by using individual vehicle to detect the congestion by itself, which are more related to our work. Among them, [16] used Cell Dwell Time (CDT) - the hand off time between two cell base stations, as the main features and used neural networks for classification. This method can only give an approximate suggestion of where the congested area is since the range of base station is often extensive. [17] has a similar idea to ours, whereas their features of moving pattern are samples of a time-indexed velocity function instead of that of a distance-indexed one in our work. We argue that the velocity-time graph is not as suitable as the velocity-distance graph, since the stop-and-go patterns are based on the behavior in different locations. Moreover, if we use the information of the past 5 minutes to detect congestion, it is almost impossible to determine the exact location of the congestion, unless the entire trajectory is recorded, which brings extra overhead. In addition, their method of collecting data is also less than desirable. The data is collected from the same vehicle, and all data instances are extracted from the same trace. Therefore, the obtained performance numbers are too optimistic (see discussions in subsection II-C). [23] also utilized a machine learning mechanism to classify the traffic state. However, since all their experiments are performed in simulations, including during the feature selection stage, it
is hard to determine whether the mechanism is feasible in reality. Yoon et al. [14] also thought of the importance of trace movements in determining the traffic condition. Specially, they emphasized the effect of red light duration in detecting congestion and used spatial mean speed to remove red light influence. This idea is similar to directly averaging the features in our work to determine the traffic state, which can also prove the reasonableness of our features. However, additional works needs to be done to collect a series of traversing information from the probe vehicles in order to determine the duration of the red light. This is not the case with our system, where detection is fulfilled only by individual cars.

V. CONCLUSION AND FUTURE WORK

In this paper, we present an effective and practical approach to determine the traffic state in urban scenarios. The biggest benefit of our work is that it does not require external assistance for traffic congestion detection - the task can be fulfilled by a single vehicle. Because of this distinctive feature, our work presents itself as a practical solution in reality, especially as part of a navigation application in a vehicular network with V2V communications. Our system works even in early stage of building such a network, when only a small number of vehicles on the road are equipped with the technology. In addition, we proposed a novel yet effective method to collect highly accurate real-world vehicle trace by using lidar. For this study, the obtained data is labeled not only by human judgement but also based on an existing theory, which brings higher fidelity to our evaluation results. Our classifier achieves high accuracy with internal and external modes of 92.20% and 88.94%, respectively. In the future, we plan to collect more data at a number of different locations and investigate the use of other features from the vehicle trace to further improve the classification performance. Additionally, GPS data will be collected from mobile devices onboard vehicles on the road, in order to validate that with GPS data our proposed method and developed model can still yield high accuracy.

ACKNOWLEDGMENT

This work was supported in part by National Science Council, National Taiwan University, and Intel Corporation under Grants NSC-101-2911-I-0 02-001 and NTU-102R7501, and a MediaTek Fellowship.

REFERENCES


