Soft Comput (2018) 22:5707–5718
https://doi.org/10.1007/s00500-017-2610-y

FOCUS

Using machine learning and big data approaches to predict travel time based on historical and real-time data from Taiwan electronic toll collection

Shu-Kai S. Fan[1] · **Chuan-Jun Su**[2] · **Han-Tang Nien**[1] · **Pei-Fang Tsai**[1] · **Chen-Yang Cheng**[1]

Published online: 29 April 2017

Abstract As the technology in automation and computation advances, traffic data can be easily collected from multiple sources, such as sensors and surveillance cameras. To extract value from the huge volumes of available data requires the capability to process and extract patterns in large datasets. In this paper, a machine learning method embedded within a big data analytics platform is constructed by using random forests method and Apache Hadoop to predict highway travel time based on data collected from highway electronic toll collection in Taiwan. Various prediction models are then developed for highway travel time based on historical and real-time data to provide drivers with estimated and adjusted travel time information.

Keywords Big data · Random forests · Electronic toll collection (ETC) · Travel time prediction · Apache Hadoop

1 Introduction

To avoid congestion and to increase the utilization of the entire highway network relies on the ability to predict travel time in a timely manner (Chien and Kuchipudi 2003; Zhang and Rice 2003; van Lint 2006; Yildirimoglu and Geroliminis 2013; Vlahogianni et al. 2014). The predicted travel time provides the drivers as an aggregated traffic information that affects their travel plans. A reliable prediction in travel time needs to meet the following three objectives: accuracy, robustness, and adaptability (van Lint 2006). The traffic data on highway can usually be collected by surveillance devices on fixed locations, such as radio frequency sensors (Chien and Kuchipudi 2003), loop detectors (Zhang and Rice 2003; van Lint 2006; Yildirimoglu and Geroliminis 2013), and even cameras (Innamaa 2005). Various prediction methods have been used in processing these traffic data, such as time series methods (Fei et al. 2011), regression models (Wu et al. 2004; Qiao et al. 2016), and machine learning methods (Innamaa 2005; Khosravi et al. 2011). A more detailed description on researches related to travel time prediction can be found in Vlahogianni et al. (2014). For previous travel time prediction, interested readers can be referred to Li and Chen (2013, 2014), and Gal et al. (2017).

Modern sensor-enabled electronic products generate huge volumes of data in real time. For instance, a single aircraft turbine will generate 10 TB of data every 30 min, and Google processes more than 24 petabytes of data each day, while Facebook receives 10 million posts every hour. The emergence of big data has coincided with the development of social media, mobile communications technologies, cloud computing and new data analytics techniques to fundamentally change how we live, work, and interact. Mobile communications and social media are transforming individual engagement, and creating new expectations of security, trust, and value in return for personal information. Cloud computing is transforming IT and business processes. Big data analytics are producing new resources which are to transform business and industry in a paradigm shift. This exponential increase in data volume has overwhelmed the storage and processing capacities of mainframe computer systems along with existing technologies (Chen et al. 2014;

Communicated by Y. Ni.

✉ Shu-Kai S. Fan
morrisfan@mail.ntut.edu.tw

[1] Department of Industrial Engineering and Management, National Taipei University of Technology, Taipei City 10608, Taiwan, ROC

[2] Department of Industrial Engineering and Management, Yuan Ze University, Taoyuan City 32003, Taiwan, ROC

Kalambe et al. 2015). Big data is mostly unstructured and entails a wide range of formats and content, indicating that it can consists of many forms, such as traditional data, metadata, streaming, video, media, transaction, digital images, sensors and audios, social media data, among others.

This paper develops two models to predict freeway travel time through big data analysis of data collected from the Taiwan Highway Electronic Toll Collection Systems (ETC). The goal is for the system to provide drivers with accurate travel time predictions in response to real-time traffic conditions. Two travel time prediction models are established based on historical freeway data: one-destination travel time prediction (OTTP) and adaptive travel time prognosis (ATTP). The performance of the two models is evaluated under different scenarios. This paper intends to contribute to the highway travel time prediction in using big data techniques, and a big data analytics platform is built upon the framework of Apache Hadoop.

2 Random forests and Apache Hadoop

2.1 Random forests

Random forests are an ensemble of classifiers that construct multiple decision trees. A combination of tree predictors is created such that each tree depends on the values of a random vector or independently sampled features and assumes the same distribution for all trees in the forest (Breiman 2001a, b). To date, the random forests model has been widely applied to various research fields (Greenhalgh and Mirmehdi 2012; Chen and Howard 2016; Mistry et al. 2016; Xu et al. 2016; Joshi et al. 2017). For classification tasks, random forests typically give high accuracy and fast classification time. A random forests classifier requires training with large datasets, which in our study is readily available due to the nature and extent of the ETC data. Random forests can also handle the vector of thousands of features and produce a classifier with high classification accuracy if well trained. For this reason, random forests will be used in this paper to conduct highway travel time prediction.

For each tree, the feature selection is conducted randomly, so the split node also differs from tree to tree, depending on the feature selection. The basic steps of the random forests algorithm are summarized below:

(1) Let the number of training cases be N, and the number of variables in the classifier be M (i.e., the number of features).
(2) Decide the number m of input variables (i.e., features) to be used to determine the decision at a node of the tree; m should be considerably smaller than M. In general, the default number m is the square root of M in many open-source software.
(3) To construct trees, choose a training set k times with replacement from all N available training cases. Each of these datasets is called a bootstrap dataset. The number k indicates the number of tress to be trained.
(4) For each tree node, randomly choose m variables on which to make the decision at that node. Calculate the best split based on these m variables in the training set.
(5) Each tree is fully grown and not pruned.

At each node of the individual decision tree, the best split is chosen based on a random variable. In this paper, the "Gini Index" is used to calculate the gini value to determine the best split point. The random forests algorithm uses the gini index taken from the classification and regression tree (CART) learning system to construct decision trees. The gini impurity represents a measure of how often a randomly chosen element from the set would be incorrectly labeled if it were randomly labeled according to the distribution of labels in the subset. If a dataset contains elements from two classes, the gini index is defined as follows (Harris and Grunsky 2015):

$$\text{Gini}(T) = 1 - \sum_{j=1}^{n} (p_j^2) \tag{1}$$

where p_j is the relative frequency of class j in a dataset T, and n is the number of classes in the dataset.

If a dataset T is split into two subsets T_1 and T_2 with respective sizes N_1 and N_2, then the gini index of the split data is defined by

$$\text{Gini}_{\text{split}}(T) = \frac{N_1}{N}\text{Gini}(T_1) + \frac{N_2}{N}\text{Gini}(T_2) \tag{2}$$

The flow chart of the random forests algorithm is presented as follows (Fig. 1):

2.2 Apache Hadoop

The Apache Hadoop was proposed by Doug Cutting and Mike Cafarella in 2005. It is an open-source software framework that supports data intensive distributed applications. Hadoop is designed for storing data and running applications on clusters of commodity hardware. It provides massive storage for any kind of data, enormous processing power, and the ability to handle virtually limitless concurrent tasks or jobs. It can work independently across multiple computers, processing very large volumes of data. There are two main cores in Hadoop: Hadoop Distributed File System (HDFS) and MapReduce. File distribution is handled through HDFS and MapReduce jobs. The Hadoop system features one master node and multiple slave nodes. HDFS is a storage layer in

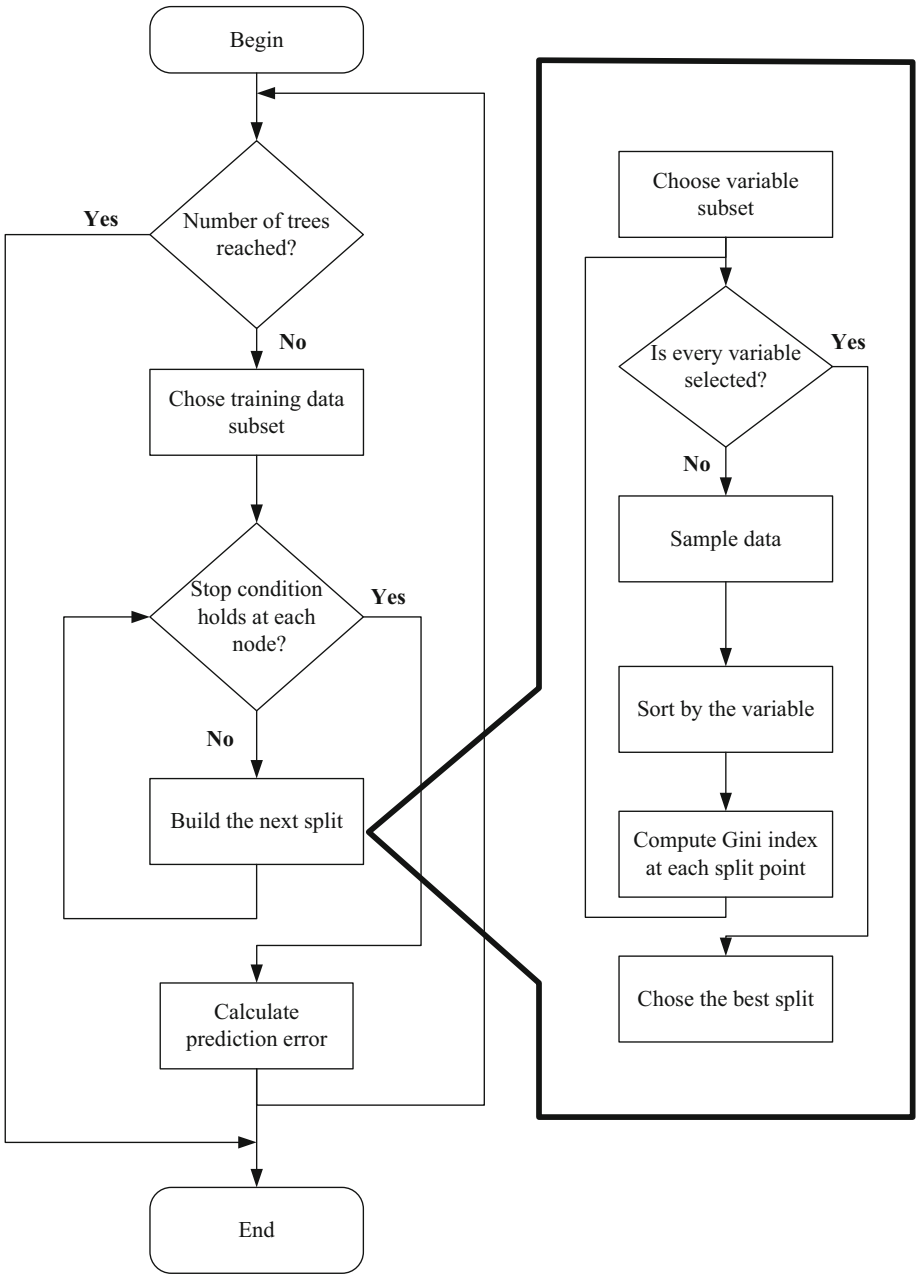


Fig. 1 Random forests algorithm procedure

which each data file is spread over many nodes. MapReduce works as a processing layer for the Hadoop programming model. The Hadoop architecture with one master and three slaves constructed in this study is illustrated in Fig. 2.

2.2.1 MapReduce

MapReduce is a programming model and a software framework that forms the core of Apache Hadoop. It allows for massive scalability across hundreds or thousands of servers in a Hadoop cluster. The Hadoop MapReduce library expresses computation in two phases: Map and Reduce (Dean and Ghemawat 2008). The map phase breaks down individual data elements into tuples (i.e., key/value pairs). The second processing step is reduce phase, which takes the output from a map as input and combines those data tuples into a smaller set of tuples. These are two phases of MapReduce shown in Fig. 3.

2.2.2 Apache Mahout

Apache Mahout is an open source project and scalable machine learning library (Jain and Jain 2014). Along with the Hadoop platform, Mahout is a promising technology for analyzing and solving data intensive problems, with built-in libraries for solving clustering, categorization, and classi-

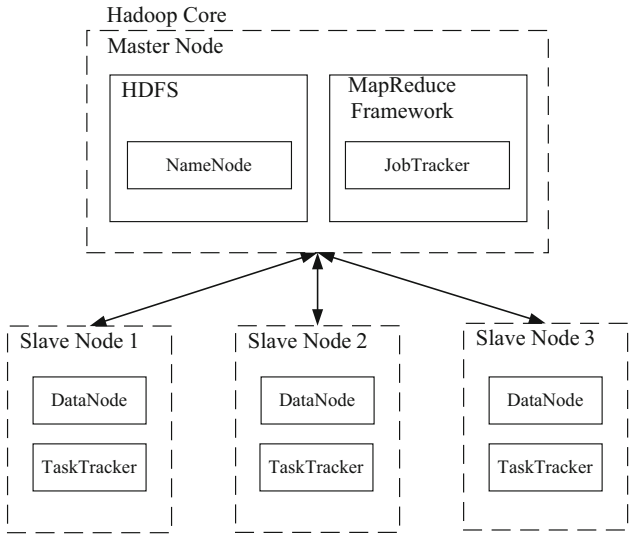


Fig. 2 Apache Hadoop architecture

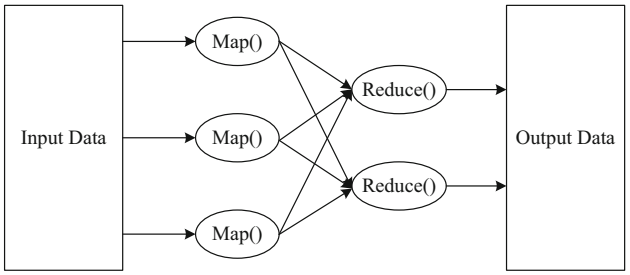


Fig. 3 Map and reduce phases

fication problems like K-mean, fuzzy K-mean, Dirichlet, Random Forests and others (Cunha et al. 2015). It can be used to apply multiple algorithms, run in parallel with Hadoop. Regarding the integration of Apache Hadoop and random forests, interested readers can be referred to Rio et al. (2014) and Singh et al. (2014).

3 Proposed approaches

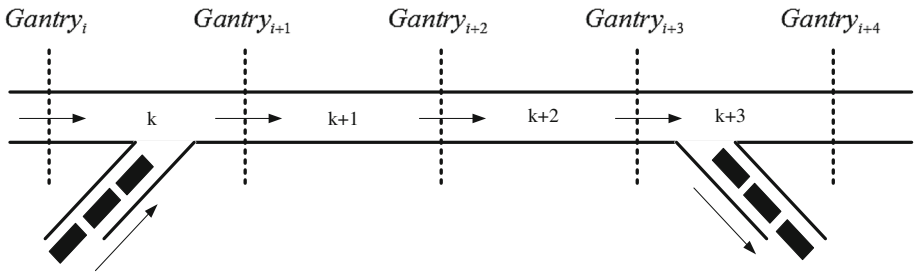
3.1 One-destination travel time prediction (OTTP)

In the one-destination travel time prediction (OTTP) module, the freeway under study is divided into k sections by gantry. Each section (from gantry to gantry) contains a range of useful traffic information including Time (by hour), Day, Vehicle Type, Traffic Flow, Travel Time, and Space Mean Speed. All the traffic information is recorded by sensors installed on the gantry. Figure 4 provides a schematic diagram of freeway sections in the OTTP module.

The OTTP module is developed for use in normal traffic conditions and does not account for unexpected traffic congestion or accidents; thus, the training data of Model One is divided into 24 hourly time intervals. The data during each time interval will be used to train one forest consisting of several decision trees. The representative training datasets in the OTTP module are displayed in Fig. 5. As to the user-specified departure time, the training dataset which is the closest to the departure time is used for travel time prediction via the OTTP module.

For OTTP feature selection, we considered the features of Time (by hour), Day, Gantry From, Gantry To, and Vehicle Type. The OTTP model includes two major steps: training and prediction. We constructed forests using randomly selected parameter combinations and different numbers of trees during the training step. For the prediction step, "travel

Fig. 4 Schematic diagram of freeway sections in the OTTP module



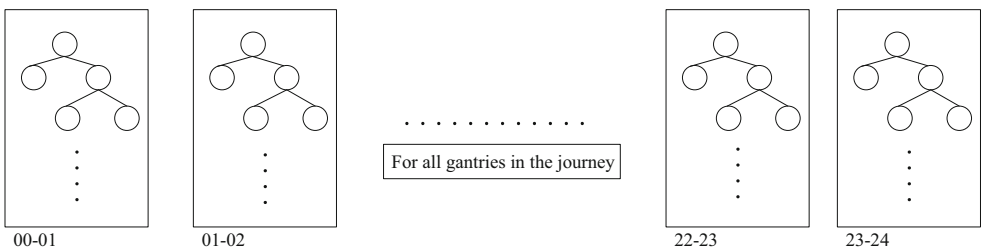


Fig. 5 Schematic diagram of time-based datasets in the OTTP module

time" as the target is predicted by voting via the forests trained in the first step. The detailed procedure of the two steps in the OTTP module for a single gantry-to-gantry section is illustrated in Fig. 6. Assuming that only two classes are considered in the figure, the result is assigned to class A by the voting output, where class A represents the travel time level in this segment.

Since the OTTP module is developed based on the one-time, gantry-to-gantry prediction model by means of historical data, it cannot accommodate dynamic, abrupt changes in freeway traffic conditions. To enhance prediction adaptability and accuracy, we developed Model two, an adaptive travel time prognosis, to remedy the shortcomings of the OTTP model.

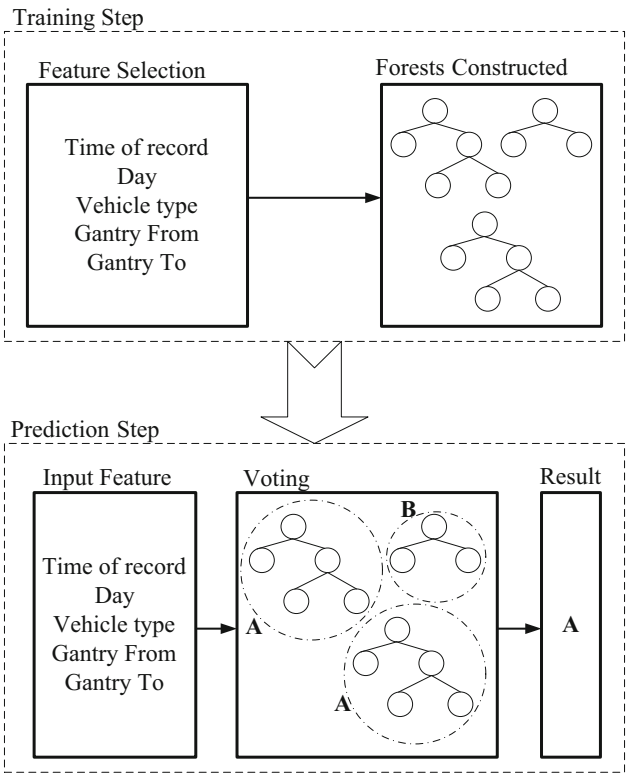


Fig. 6 Detailed two-step procedure in the OTTP module

3.2 Adaptive travel time prognosis (ATTP)

In the ATTP module, the freeway is, as before, divided into k sections by gantry. In Fig. 7, we intend to predict the travel time from interchange A to interchange C using the OTTP module and to adapt the travel time in terms of real-time data using the ATTP module. For illustration purposes, a large number of vehicles enter the freeway at interchange B, resulting in traffic congestion in section $k+1$. This unexpected activity (or event) would produce significant errors in travel time prediction by means of the OTTP module. The ATTP module is designed to reduce the severity of such prediction errors.

We collected real-time data from Traffic Data Collection System (TDCS), i.e., the Taiwan Highway Electronic Toll Collection Systems (ETC), and adapted travel time prediction to real-time traffic flows. We conducted travel time prognosis for two upcoming sections $k+1$ and $k+2$, based on the current section k. We constructed the forests involving imminent traffic conditions determined by gantry-based historical data, regardless of day or time of recording, as demonstrated in Fig. 8. We input the real-time data of these three sections to the forests for training according to individual gantries to update the travel time of each section. If the travel time prediction between OTTP and ATTP differs significantly, we update the OTTP travel time using the ATTP travel time to the freeway user.

The ATTP module feature selection considered the features of Vehicle Type, Traffic Flow, Gantry From, Gantry

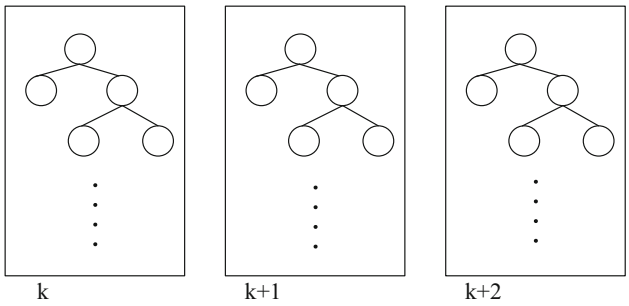
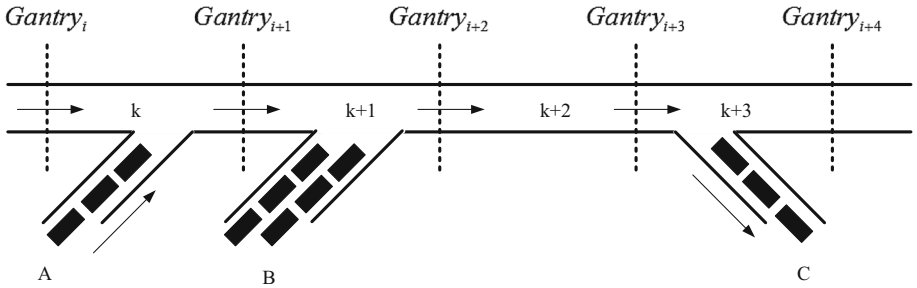


Fig. 8 Schematic diagram of gantry-based datasets in the ATTP module

Fig. 7 Schematic diagram of the ATTP module



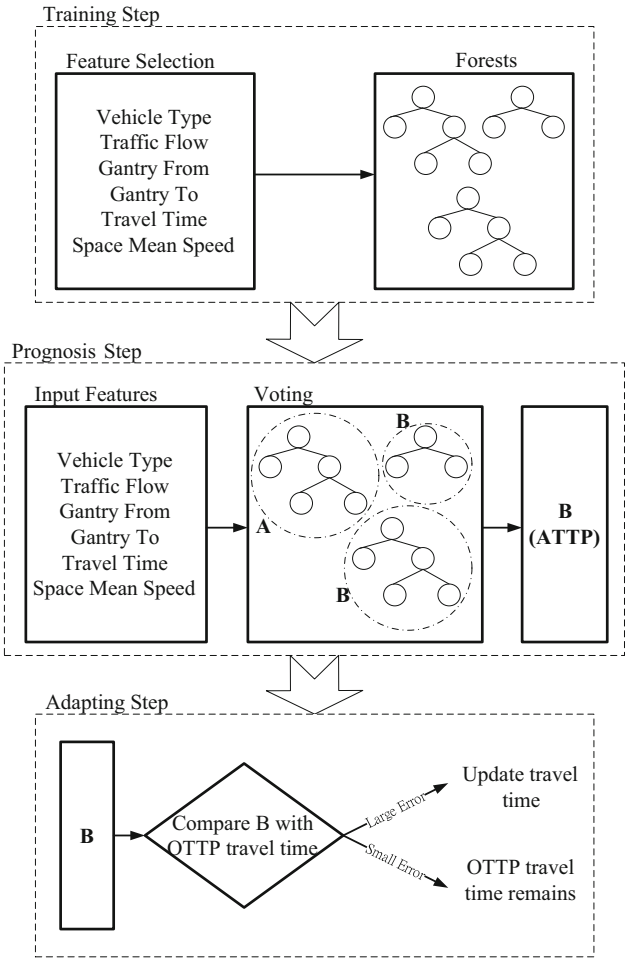


Fig. 9 Three major steps of the ATTP module

To, Travel Time, and Space Mean Speed. The ATTP module consists of three major steps: training, prognosis, and adaptation. As usual, we constructed forests with randomly selected parameter combinations and different numbers of trees in the current section and the following two sections during the training step. For the prognosis step, travel time prediction is obtained by voting with real-time data as inputs into the forests trained in the first step. We then compare the travel time prediction against the one returned by the OTTP module. If the OTTP module fails to reflect the current traffic situation in the upcoming sections, we update the travel time using the ATTP result. Figure 9 shows the three major steps of the ATTP module, where the result is assigned to class B by voting as the ATTP travel time prediction.

4 Experimental study

In this experimental study, data were taken from Traffic Data Collection System (TDCS), i.e., the Taiwan Highway Electronic Toll Collection Systems (ETC). TDCS records travel time, mean speed, and trip length for each vehicle, along with overall traffic flow by gantry. The content of TDCS is tabulated in Table 1.

In this section, all the data in 2015 are earmarked for training, and the January and February data in 2016 are used for testing. Two models were developed for application in different situations: one-destination travel time prediction (OTTP) and adaptive travel time prognosis (ATTP).

4.1 Prediction results of one-destination travel time prediction

Here, the experiments are classified by travel distance as short, medium, and long distance. For each travel distance, the number of features considered at each internal node of random forests is m, randomly chosen as suggested by Breiman (2001a, b) to be $m = \text{int}(\log_2 M + 1)$ where M is the total number of features. This is called the random subspace method. Due to the with-replacement sampling policy in random forests, every bootstrap dataset can have duplicate data records and some data records may be missing from the original dataset. This missing dataset is called out-of-bag examples. The out-of-bag (OOB) error is defined as the average prediction error of random forests models using the bootstrap aggregating to subsample data, in relation to the out-of-bag examples.

Based on the OOB error analysis, it has been discovered that the travel time OOB prediction error stabilizes as the number of trees reaches 50. To assess the effectiveness of travel time prediction for each travel distance, the number of trees in the forest is set to $k = 50$. In every experiment, we compute the prediction error for the evaluation purpose. The experimental results are displayed in Tables 2, 3, and 4. The performance measure, mean absolute percentage error (MAPE), is used to examine the prediction performance. The number in parenthesis is the standard deviation of MAPE. The MAPE statistic usually expresses accuracy as a percentage calculated as follows:

$$\text{MAPE} = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right| \tag{3}$$

where A_t is the actual travel time value and F_t is the forecast travel time value, and n is the number of the January and February data in 2016 for testing.

According to the simulation results shown above, the average MAPE results cluster around 5% for the three distance categories using the OTTP module. MAPE primarily dictates the averaged prediction error along the various roadway segments. The OTTP module may produce a noticeable prediction error, but the MAPE results are smoothed over a large amount of testing data. Basically, the OTTP module

Table 1 Features details of forms M03A–M08A from TDCS

Form no.	Content of features							
M03A	Time interval	Gantry ID	Direction	Vehicle Type	Traffic Flow	–	–	–
M04A	Time interval	Gantry From	Gantry To	Vehicle Type	Travel Time	Traffic Flow	–	–
M05A	Time interval	Gantry From	Gantry To	Vehicle Type	Space Mean Speed	Traffic Flow	–	–
M06A	Vehicle type	Detection time_O	Gantry ID_O	Detection time_D	Gantry ID_D	Trip length	Trip end	Trip information
M07A	Time interval	Gantry From	Vehicle Type	Averaged Travel Time	Traffic Flow	–	–	–
M08A	Time interval	Gantry From	Gantry To	Vehicle Type	Traffic Flow	–	–	–

is capable of providing freeway users a reasonably accurate travel time prediction for a specific day and time as no abrupt changes in freeway traffic conditions occur. In fact, the OTTP module seems unfazed with respect to three different distance categories, producing about 5% of MAPE. Since the OTTP module is constructed on a time basis, including the time by hour and the day by week, the prediction error should be stable in spite of travel distances.

4.2 Prediction results of adaptive travel time prognosis

In this section, the simulation study focuses on travel time adaptation in the short-distance category. As before, the number of features is set identical to the OTTP module and the number of trees is $k = 50$. We compute the prediction error of MAPE using the travel time prognostic. If the OTTP module produces a large error, the travel time will be updated to the ATTP result. The accuracy of the travel time prognostic is then evaluated and compared. In this study, if the prediction time between the OTTP and ATTP modules differs by more than 15%, then the prediction time by the ATTP module will be used to replace the one returned by using the OTTP module. The comparison results are shown in Tables 5, 6, and 7. Note that in an earlier simulation study three different percentages, 10, 15 and 20%, in prediction time difference between the OTTP and ATTP modules were tested. We found that if the level of 10% was used, then the update to the ATTP result will be overkill. If the level of 20% was used, then the traffic dynamic cannot be appropriately accommodated.

Comparison results show that, in some instances, the ATTP module significantly improves travel time prediction, possibly due to use of historical data to train the OTTP module leaving it unable to closely predict real-time traffic conditions in 2016. Under such circumstances, the ATTP module takes the dynamic traffic information into consideration and recalculates the travel time in a prognostic manner. Overall, the ATTP simulation results demonstrate that most travel times can be accurately predicted using real-time ETC data, with the ATTP module providing improved prediction accuracy over the OTTP model. Note that the ATTP module can be applied to longer distances. In this paper, we only presented the prediction results of the ATTP module on the short distances for the illustration purpose.

5 Conclusion

This paper implements a random forests algorithm on Apache Hadoop for travel time prediction and prognosis from Taiwan electronic toll collection (ETC) data. The forecast models use historical data and dynamic data. The current practice for predicting travel time assumes that vehicle speed remains constant along the various roadway segments. This approach produces large prediction errors, especially when segment speeds vary temporally. We developed models to predict travel time before departure and adapt to dynamic traffic patterns. Experimental results show that the one-destination travel time prediction (OTTP) model provides high accuracy for normal traffic conditions, while the adaptive travel time prognosis (ATTP) can effectively adapt to dynamic traffic patterns. Combining the two models provides highly accurate travel time prediction for freeway drivers. The OTTP module helps highway drivers select optimal departure times to avoid traffic congestion and thus minimize travel time. The ATTP module provides accurate time of arrival predictions based on analysis of current traffic conditions, allowing drivers to select alternate routes to further reduce travel times.

Table 2 MAPE prediction performance (%) for short distances in the OTTP module

Metric	Time							
	00–01	01–02	02–03	03–04	04–05	05–06	06–07	07–08
Avg. MAPE	7.0776 (2.4069)	4.3395 (1.3185)	5.5496 (1.7448)	4.1535 (2.8708)	5.4964 (2.7892)	4.6406 (1.1351)	4.7666 (1.2528)	3.3388 (2.5992)
Max. MAPE	67.3333	32.6	81.5	82.5	79.5	74.5	68.2	76.6
Min. MAPE	1.9418	0.0952	0.7116	2.4267	0.6308	2.8016	1.0785	2.5153
Metric	Time							
	08–09	09–10	10–11	11–12	12–13	13–14	14–15	15–16
Avg. MAPE	5.3964 (1.7398)	5.0833 (0.3796)	5.1639 (0.6952)	5.1556 (4.0314)	5.3919 (1.1216)	5.2998 (0.6035)	5.5080 (1.4612)	5.2982 (0.9632)
Max. MAPE	53	78.5	76.5	52.6667	79.5	48.6667	80.5	50.7
Min. MAPE	0.0924	0.0160	1.6709	1.4867	1.5563	0.3012	2.8303	1.5186
Metric	Time							
	16–17	17–18	18–19	19–20	20–21	21–22	22–23	23–24
Avg. MAPE	5.1755 (2.8435)	4.5826 (1.9911)	4.7019 (0.3991)	4.1382 (0.6948)	4.9516 (0.7410)	5.7420 (3.4756)	5.7831 (1.0260)	5.7741 (1.4338)
Max. MAPE	51.6667	52.6667	75.67	44.6667	52.6667	77.5	51.6667	49.68
Min. MAPE	3.7257	0.8319	1.2025	2.5962	1.2993	1.1889	1.2426	0.8598

Table 3 MAPE prediction performance (%) for medium distances in the OTTP module

Metric	Time							
	00–01	01–02	02–03	03–04	04–05	05–06	06–07	07–08
Avg. MAPE	6.9602 (2.8982)	4.4184 (1.6911)	5.5488 (1.9081)	4.1947 (2.4653)	5.5052 (1.3687)	4.5802 (4.2960)	4.6672 (2.4264)	3.2871 (3.3297)
Max. MAPE	67.3333	36	81.5	82.5	79.5	74.5	70.3	78.5
Min. MAPE	3.1030	0.4550	2.3277	0.3414	1.2734	1.0312	0.1319	1.5086
Metric	Time							
	08–09	09–10	10–11	11–12	12–13	13–14	14–15	15–16
Avg. MAPE	5.3782 (0.1609)	4.9993 (0.4771)	5.1239 (0.8327)	5.2416 (1.6359)	5.4441 (0.6776)	5.3046 (1.2129)	5.4311 (1.1228)	5.2694 (1.1687)
Max. MAPE	61	81.5	80.5	53.6667	79.5	52.7	80.5	52.6667
Min. MAPE	4.0951	0.4305	2.5124	2.9405	0.3599	2.7724	0.5316	1.6312
Metric	Time							
	16–17	17–18	18–19	19–20	20–21	21–22	22–23	23–24
Avg. MAPE	5.0853 (2.3210)	4.7469 (1.5360)	4.5253 (1.5746)	4.3522 (3.2804)	5.2835 (1.4159)	5.8546 (7.7296)	5.7924 (2.2621)	5.8974 (1.6338)
Max. MAPE	54.2	52.6667	80.5	47.52	80.5	88.64	53.82	53.5
Min. MAPE	4.0720	0.3548	2.0211	1.3840	1.9742	0.4339	1.9172	2.0063

Table 4 MAPE prediction performance (%) for long distances in the OTTP module

Metric	Time							
	00–01	01–02	02–03	03–04	04–05	05–06	06–07	07–08
Avg. MAPE	6.7809 (2.5665)	4.4196 (2.7301)	5.4775 (1.3911)	4.1432 (1.0811)	5.5230 (1.2593)	4.6526 (5.2853)	4.7959 (2.3983)	3.3329 (1.4965)
Max MAPE	101.5	36	81.5	82.5	79.5	76.5	74.5	80.5
Min MAPE	9.2283	0.2188	1.9738	0.5699	1.9492	1.6190	0.4629	2.8429
Metric	Time							
	08–09	09–10	10–11	11–12	12–13	13–14	14–15	15–16
Avg. MAPE	5.3896 (1.1801)	4.9326 (1.1073)	5.1647 (2.6024)	5.1985 (1.2725)	5.4402 (2.0346)	5.4056 (1.3825)	5.4020 (2.5311)	5.2488 (4.1662)
Max MAPE	68.6	82.5	80.5	55.47	80.5	80.5	80.5	52.6667
Min MAPE	5.3710	1.2216	0.4737	2.0310	2.8166	2.2903	2.0708	1.9680
Metric	Time							
	16–17	17–18	18–19	19–20	20–21	21–22	22–23	23–24
Avg. MAPE	4.8816 (2.8983)	4.6592 (4.8332)	4.4301 (1.1213)	4.5323 (1.4280)	5.3677 (4.7367)	5.7709 (1.1898)	5.7711 (1.3802)	5.7411 (2.9371)
Max MAPE	57.3	80.5	80.5	52.6667	80.5	88.64	53.82	62.5
Min MAPE	15.6855	2.3000	0.8438	2.9633	2.2048	0.3491	2.6034	2.7565

Table 5 ATTP versus OTTP short-distance results (1)

Time	Gantry From	Gantry To	MAPE of OTTP	MAPE of ATTP	Improvement (%)
01/18	Zhongli	Airport	21.6250	10.8750	10.7500
8:45	Airport	Taoyuan	27.8750	12.4770	15.3980
	Taoyuan	Linkou	10.4736	8.0000	2.4736

Table 6 ATTP versus OTTP short-distance results (2)

Time	Gantry From	Gantry To	MAPE of OTTP	MAPE of ATTP	Improvement (%)
01/17	Sanchong	Taipei	4.7222	0.1944	4.5278
15:20	Taipei	Yuanshan	31.2000	14.6200	16.5800
	Yuanshan	Neihu	5.0344	0.1379	4.8965

Table 7 ATTP versus OTTP short-distance results (3)

Time	Gantry From	Gantry To	MAPE of OTTP	MAPE of ATTP	Improvement (%)
02/01	Wudu	Dahua	0.1875	0.1875	0.0000
20:55	Dahua	Badu	0.0769	0.0769	0.0000
	Badu	Keelung	31.0000	0.8000	30.2000

Acknowledgements This study was partially funded by the Ministry of Science and Technology (Taiwan) Grant: MOST 105-2221-E-027-052 -MY3.

Compliance with ethical standards

Conflict of interest All the authors of this paper declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

Breiman L (2001a) Bagging predictors. Manuf Neth Mach Learn 24:123–140

Breiman L (2001b) Random forests. Manuf Neth Mach Learn 45:5–32

Chen M, Mao S, Liu Y (2014) Big data: a survey. Mob Netw Appl 19:171–209

Chen FH, Howard H (2016) An alternative model for the analysis of detecting electronic industries earnings management using stepwise regression, random forest, and decision tree. Soft Comput 20:1945–1960

Chien SI-J, Kuchipudi CM (2003) Dynamic travel time prediction with real-time and historic data. J Transp Eng 129(6):608–616

Cunha J, Silva C, Antunes M (2015) Health Twitter Big Bata Management with Hadoop Framework. Proc Comput Sci 64:425–431

Dean J, Ghemawat S (2008) MapReduce: simplified data processing on large clusters. Commun ACM 51(1):107–113

Fei X, Lu C-C, Lui K (2011) A bayesian dynamic linear model approach for real-time short-term freeway travel time prediction. Transp Res Part C 19:1306–1318

Gal G, Mandelbaum A, Schnitzler F, Senderovich A, Weidlich M (2017) Traveling time prediction in scheduled transportation with journey segments. Inf Syst 64:266–280

Greenhalgh J, Mirmehdi M (2012) Traffic sign recognition using MSER and random forests. In: Proceedings of the 20th European signal processing conference

Harris JR, Grunsky EC (2015) Predictive lithological mapping of Canada's north using random forest classification applied to geophysical and geochemical data. Comput Geosci 80:9–25

Innamaa S (2005) Short-term prediction of travel time using neural networks on an interurban highway. Transportation 32:649–669

Jain E, Jain S (2014) Categorizing Twitter Users on the basis of their interests using Hadoop/Mahout Platform. In: Proceedings of the 9th international conference on industrial and information system

Joshi A, Monnier C, Betke M, Sclaroff S (2017) Comparing random forest approaches to segmenting and classifying gestures. Image Vision Comput 58:86–95

Kalambe YS, Pratiba D, Shah P (2015) Big data mining tools for unstructured data: a review. Int J Innov Technol Res 3(2):2012–2017

Khosravi A, Mazloumi E, Nahavandi S, Creighton D, van Lint JWC (2011) A genetic algorithm-based method for improving quality of travel time prediction intervals. Transp Res Part C 19:1364–1376

Li CS, Chen MC (2013) Identifying important variables for predicting travel time of freeway with non-recurrent congestion with neural networks. Neural Comput Appl 23:1611–1629

Li CS, Chen MC (2014) A data mining based approach for travel time prediction in freeway with non-recurrent congestion. Neurocomputing 133:74–83

Mistry P, Neagu D, Trundle PR, Vessey JD (2016) Using random forest and decision tree models for a new vehicle prediction approach in computational toxicology. Soft Comput 20:2967–2979

Qiao W, Haghani A, Shao C-F, Lui J (2016) Freeway path travel time prediction based on heterogeneous traffic data through nonparametric model. J Intell Transp Syst 20(5):438–448

Rio SD, Lopez V, Benitez JM, Herrera F (2014) On the use of MapReduce for imbalanced big data using Random Forest. Inf Sci 285:112–137

Singh K, Guntuku SC, Thakur K, Hota C (2014) Big data analytics framework for peer-to-peer botnet detection using random forests. Inf Sci 278:488–497

van Lint JWC (2006) Reliable real-time framework for short-term freeway travel time prediction. J Transp Eng 132(12):921–932

Vlahogianni EI, Karlaftis MG, Golias JC (2014) Short-term traffic forecasting: where we are and where we're going. Transp Res Part C 43:3–19

Wu C-H, Ho J-M, Lee DT (2004) Travel-time prediction with support vector regression. IEEE Trans Intell Transp Syst 5(4):276–281

Xu Y, Zhang Q, Wang L (2016) Metric forests based on Gaussian mixture model for visual image classification. Soft Comput. doi:10.1007/s00500-016-2350-4

Yildirimoglu M, Geroliminis N (2013) Experienced travel time prediction for congested highways. Transp Res Part B 53:45–63

Zhang X, Rice JA (2003) Short-term travel time prediction. Transp Res Part C 11:187–210