Efficient Bus Arrival Time Prediction Based on Spark Streaming Platform

Jing Liu
Inner Mongolia University
Hohhot, China
liujing@imu.edu.cn

Guanfeng Xiao
Inner Mongolia University
Hohhot, China
291746705@qq.com

***Abstract*—In smart city development, the prediction of bus arrival time is a popular research issue, which often uses GPS data and other related bus data to conduct collaborative data analysis. It is of great importance for improving the public transportation services. But the accuracy and the efficiency of bus arrival time prediction is still the major obstacles. In this paper, an optimized particle-filtering algorithm is used to establish a bus arrival time prediction model. To better solve the problem of prediction error and particle optimization in the process of using particle filter algorithms, the prediction model is improved by introducing the latest bus speed for collaborative data analysis, which improves the accuracy of the bus arrival time prediction based on the actual road conditions and can simultaneously predict the arrival time of multiple buses. Based on the above model and the Spark streaming platform, a real-time bus arrival time prediction software system is implemented. The experimental results show that our proposed model and system can accurately predict the bus arrival time and then well promote the bus travel experience for citizens.**

Keywords—Bus arrival time prediction, particle filter algorithm, Spark streaming, smart city

I. Introduction

Building a collaborative transportation platform in smart city is quite promising and important. Especially, the prediction of bus arrival time is a popular research issue, which often uses GPS data and other related bus big data to conduct collaborative data analysis. Predicting the arrival time of buses creates a better urban travel experience for citizens, and it is significant in promoting the development of smart cities [1]. However, for a city with dense population and complex public transportation conditions, it is really difficult to develop a universal, accurate and efficient model for predicting bus arrival time.

Currently, there have been many advances in related research on bus arrival time prediction, but major obstacles come from two aspects. First, the prediction model based on big data training and mining often results in low model prediction accuracy. Besides, models based on real-time data processing have emerged in recent years. Many application restrictions in these models also result in the problem of low prediction accuracy. Second, most studies now focus on theoretical modeling and experimental evaluations. However, in practice, it requires not only a suitable and accurate model but also an integrated big data analysis system solution for bus arrival time prediction applications. In particular, if the prediction system can be optimized using the mature big data technologies, the prediction accuracy will be greatly improved.

Based on a particle filter algorithm with characteristics of streaming computing, in this paper, we propose a novel bus arrival time prediction model. To better solve the problem of prediction error and particle optimization in the process of using the particle filter algorithm, our prediction model is optimized by introducing the latest bus speed for better collaborative data analysis, which improves the accuracy of the bus arrival time prediction and could simultaneously predict the arrival time of multiple bus stops. In addition, utilizing the advantages of the Hadoop Spark platform, we implement a bus arrival real-time prediction software system based on above model. The system can efficiently predict all unfinished bus stops. The arrival time of the buses at each subsequent stop that are predicted by our model will fully meet the daily needs of passengers.

II. Related Work

Bus arrival time prediction models are mainly based on machine learning algorithms and evolutionary algorithms. A neural network was used based on multilabel classification to predict bus arrival time [2]. Experimental results showed that neural networks perform better than decision trees, random forests, and naive Bayes methods. That paper also discussed an integrated neural network using the AdaBoost algorithm and RAKEL algorithm and found that the RAKEL integrated model performs better in multilabel accuracy. In [3], the studies combine the clustering and neural networks. The experimental results showed that the model is superior to the traditional BP neural network. The major problem is that it is difficult to immediately reflect the impact of factors such as urban public transport, road construction, and road conditions. In [4], the Kalman filter algorithm is applied. The experimental results show that using the Kalman filter model or using a hybrid model based on the Kalman filter has better prediction accuracy when predicting the bus arrival time. In [5], a hybrid model based on SVM and the Kalman filtering algorithm was proposed for the arrival time prediction of bus rapid transit in certain cities. First, the bus arrival time was forecasted using SVM, and then the Kalman filter algorithm was used to dynamically adjust the prediction results. The experimental results show that the prediction accuracy of this model was higher than that based on Kalman filtering. However, the Kalman filter has the problem of degrading prediction accuracy in non-Gaussian systems. In [6], a real-time bus arrival time prediction system was constructed based on particle filtering, and the particle filter algorithm was applied to the problem of bus arrival time prediction. The paper used the relative percent error (MAPE) and a method based on the Kalman filter to evaluate the better prediction results.

978-1-7281-0350-1/19/$31.00 ©2019 IEEE

III. MODELING FOR BUS ARRIVAL TIME PREDICTION

A. Computing paradigm selection

In our city, the GPS devices in buses send their driving data, including location information, to the server at regular intervals (usually 15 seconds). These data have the characteristics of streaming data that arrive in real time. But for the server, the arrival time of each bus's location message is predictable and uniform in distribution, and the data traffic does not fluctuate violently. After the above analysis, it is determined that the application scene of the bus arrival time real-time prediction system has certain real-time computing requirements. However, in this scenario, the calculation does not need to occur at the time of arrival of each GPS record, i.e., the computing does not need to be performed at a specific time, and only a small batch of new GPS records need to be calculated together once for each period of time. The Spark streaming computing framework with the characteristic of small batch processing is more suitable for the above application scenarios, and the computing power is sufficient to meet the demands of prediction computing [7]. It is necessary to use real-time data and historical data in conjunction with each other for better collaborative calculation. The window mechanism in Spark supports the stream processing of data collections over a period of time (such as a day or a week). This meets the needs of dealing with historical data in the prediction.

B. Prediction algorithm selection

After Spark streaming framework is determined as the real-time computing paradigm, it is necessary to identify a prediction algorithm that meets the characteristics of the streaming computing and has a high prediction accuracy. Among current prediction algorithms, particle filters as nonlinear, non-training, highly flexible models [8] can adapt well to complex and rapidly changing traffic conditions, and the calculation process accords with streaming processing to some extent. For example, the implementation of the particle filter algorithm is based on the process of continuously acquiring the latest data and correcting and updating the existing prediction results through real-time calculations. In addition, the particle filter algorithm can be embedded in a streaming computation framework to promote the prediction accuracy. The particle filter algorithm approximates the probability density function by finding a set of random samples that propagate in the state space and replaces the integral operation with the sample mean to obtain the minimum variance estimation process of the system state. These samples are visually called particles.

The particle filter includes two stages of predicting and updating. The prediction process needs to collect real-time bus driving information to calculate the partial historical driving data of the weighted speed. The updating process continuously collects the latest bus driving records and re-predicting them. It is determined that the prediction system needs both real-time data and part of the historical data in conjunction with real-time data for real-time calculations.

Particle filter algorithms based on the sequential Monte Carlo simulation method have particle weight degradation problems in the iterative process. In the actual calculation process, after several iterative calculations, a large number of particles with very small weights appear in the particle swarm. These large numbers of particles exist but are not useful for describing the state distribution. They cause the particle set to no longer effectively represent the probability distribution of variables. With the increase in the number of invalid particles, many calculations are spent on useless particles, resulting in highly reduced computing efficiency. The basic idea of the resampling process in the particle filter algorithm is to filter out unwanted particles, i.e., particles with smaller weights and to duplicate particles with larger weights to supplement them so that particles continue to be used when the number of particles is not changed. Currently, the most commonly used resampling method is sequential importance sampling (SIS). In the Monte Carlo simulation method based on importance sampling, when new observation data arrive, it is necessary to recalculate the importance weights of all particles to re-evaluate the posterior probability density. Obviously, this process continues with the arrival of new observations, that is, the process of state tracking.

C. Bus GPS data preprocessing

The bus movement trajectory is formed by connecting segments of different lengths. The direct use of the distance traveled by vehicles within 15 seconds as the basic unit of calculation is not conducive to calculation and prediction. To regulate the minimum unit for the unified calculation and prediction, bus lines are discretized into key road segments of equal length as the minimum unit for calculation. That is, bus route discretization refers to dividing the segment trajectory formed by connecting GPS points into equal-length segments by inserting certain GPS points. The insertion point is called a key point, and the line segment between adjacent key points is called key road segment. A schematic diagram is shown in Figure 1.

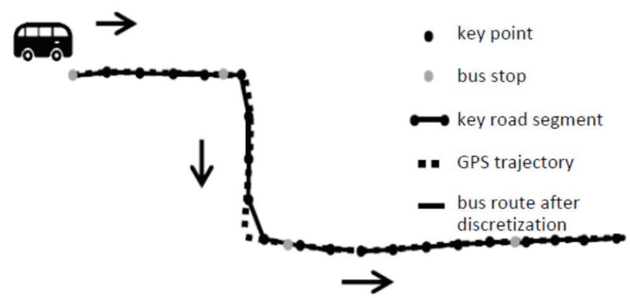

Fig. 1. Schematic diagram of bus route discretization

Combined with the actual situation, the length of the key road segment in this paper is taken as 150 meters. Assuming that the bus runs at a constant speed in the key road segment between two adjacent GPS points, the trajectory is divided into several key road segments by inserting key points. In the discretized bus route, the bus stop is no longer the basic unit of prediction, and it is only part of the key road segment information. In determining the specific location of a bus stop in a key road segment, a relative time approach is used, that is, the time required to travel to the bus stop since the most recent key point. In the bus driving data, the position, time and instantaneous speed of each GPS are known, and the time interval of every two GPS points is also known. For each key point and key road segment, the key point position, the average speed in the key road segment, and the travel time of the key road segment are also needed. In addition to the relevant key road segment information needed in the calculation process, the location of the bus stop needs to be mapped to its own key road segment because the user needs to see the result that the bus will take a certain time to reach a certain bus stop. For preprocessing results, if there is a bus stop in a key road segment, the label of that bus stop is the output, and it is necessary to know the detailed location of that bus stop in this key road segment.

D. Improvements on the multistep prediction accuracy

Using the particle filter model in bus arrival time prediction has the following problems. (1) *Prediction error*. If the average speed on the previous road segment is used to predict the driving time on all subsequent road segments, the error will be continuously amplified, and the prediction result of the following road segment will be unavailable. (2) *Acquisition of observations*. The updating process of the particle filter needs to obtain new observations to calculate the weights of the particles to filter the better particles. However, at the same time when all the arrival times of the non-arrival stops are predicted, the observations of some key road segments are not available for each calculation until the bus arrives at the last bus stop.

The key aspects to solving these two problems lie in the calculation of weighted speeds and the construction and optimization of observations. This paper uses the following method to address these two problems. (1) *Introducing the average speed of the latest previous bus in the same road segment*. The average speed in all key road segments of the latest previous bus in the same direction can be calculated as part of the weighted speed. The proportion of instantaneous speed can be reduced within a reasonable range. To a certain extent, the proportion of the historical average speed is reduced. In this way, the accuracy and adaptability of the model will be greatly improved. The weighting coefficient of each weighted speed needs to be determined through the experiments. (2) *Construction of optimized observations*. For some key road segments of the updating process where observations are not available, the arrival time of the latest previous bus at each key point better reflects characteristics of the road conditions in the most recent time period, so we select this metric as the observation value. To make this "observation" as close as possible to the actual observation, two optimizations are considered. First, we eliminate the time difference caused by the bus departure exceptions or passengers in the first road segment. Second, we eliminate the time difference between the arrival times of all the buses that have passed through the road segment.

E. Prediction model establishment

The core parts and algorithm steps of our bus arrival time prediction model are described in detail as follows.

- *model specification in detail*

First, the equation of the state and observation equation are described as follows.

$$x_k = Ax_{k-1} + B\frac{S_{k-1}}{v_{k-1}} + w_{k-1} \tag{1}$$

$$y_k = Hx_k + n_k \tag{2}$$

Then, weighted speed calculation formula is specified.

$$v_{k-1} = av_{c,k-2} + bv_{l,k-1} + cv_{h,k-1} \tag{3}$$

$$(0<a<1,\ 0<b<1,\ 0<c<1,\ a+b+c=1)$$

The specific meaning of the key parameters in the model is described in Table I.

TABLE I. KEY PARAMETERS IN OUR PREDICTION MODEL

Name	Detailed Explanation
k	the sequence number of the key road segments
x_k	the system status (travel time) of the bus in key road segment k
S_{k-1}	The length of the key road segment. It is set to 150 meters except for the last key road segment
v_{k-1}	the weighted speed, computed based on instantaneous speed $v_{c,k-2}$ (key point *k-2* to *k-1*), average speed $v_{l,k-1}$ of the previous bus in the key road segment *k-1* with the same direction, and the historical average speed $v_{h,k-1}$ of the key road segment *k-1*
y_k	the observed or constructed "observed values" of the travel time of the bus within the key road segment k
A	the state transfer coefficient
B	the controlling variable coefficient
H	the observation coefficient
w_k	the systematic error
n_k	the observation error
a, b, c	the weighting factor in speed calculations

The system error w_k and the observation error n_k in the above table are independent of each other. In the calculation, w_k takes a random number, and n_k needs to add a random error in addition to the time difference, and A, B, and H take a unit matrix. The weight calculation is described as follows.

$$w_k^j = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{(y_k = y_k^i)^2}{2\sigma^2}} \tag{4}$$

where w_k^i stands for the weight of each particle on the key road segment k; y_k stands for the observed or constructed "observed values" of the travel time of the bus in the key road segment k; y_k^i stands for the travel time of each particle; and σ stands for the standard deviation of the error.

- *core algorithm in the model*

Step 1: Initializing the algorithm parameters, taking $k = 0$, the number of iterations N, and the number of particles M.

Step 2: Categorizing all preprocessed GPS records by the following procedure.

Step 3: Obtaining all the data of buses that do not arrive at the terminals. Going ahead for each bus trip.

Step 4: Determining whether all the data for a bus has been processed. If the processing is completed, Step 14 is performed; otherwise, go to Step 5.

Step 5: Initializing the particle swarm and taking $k = 0$ to obtain the data of the latest two buses.

Step 6: Calculating the number of the key road section where this bus trip is currently located.

Step 7: Classifying the data of each bus trip according to the number of key road segments. The following process is performed for each key road segment data.

Step 8: Calculating the historical average speed of each bus trip on the key road segment k.

Step 9: Obtaining the x_k and y_k of each particle from the state equation, the observation equation, and the weighted velocity calculation formula.

Step 10: Calculating the particle group expectation. Calculating the final time and saving it.

Step 11: Calculating the weight of each particle based on the weight calculation formula and normalizing it.

Step 12: Calculating the degree of degradation and deciding whether to resample the particle swarm according to such a degree. The criterion is given as following formula:

$$\frac{1}{\sum_{1}^{N}(w_k^i)^2} < \frac{2M}{3} \tag{5}$$

Step 13: Performing the resampling.

Step 14: k=k+1. If the algorithm has been terminated, it should skip to the next step; otherwise, go back to Step 4.

Step 15: Showing the final prediction result.

The specific process of the above resampling process is developed as follows. First, we produce random numbers $\{u_i : i=1, 2,..., M\}$ that are uniformly distributed on [0,1] and M numbers are generated. Then, for each u_i, we search in sample set to find the integer n that satisfies the following condition.

$$\sum_{j=0}^{n-1} w_k^j < u_i < \sum_{j=0}^{n} w_k^j \tag{6}$$

Next, we record the sample x_k^n as a representative sample. Finally, the weights of all particles are reset to $1/M$.

IV. Bus Arrival Time Prediction System Design

The programming language to develop the bus arrival time prediction system is Scala. This section describes the key points design in the prediction process.

A. Calculation of the spark streaming window

In the prediction process, during a fixed period of time, the streaming calculation processes all GPS records starting from the current time and continuing until a fixed period of time. How to set Spark streaming window in the prediction process is shown in Figure 2.

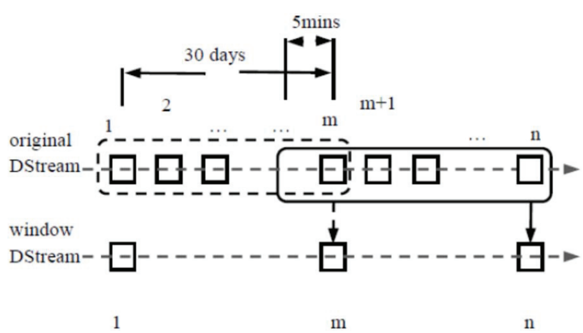

Fig. 2. Prediction window settings in Spark streaming

The window length is set to 60×24×30 (30 days), and the sliding interval is set to 5 minutes. That is, the program will process data one month before the program's running time each time, slide back once every five minutes, and then the program will execute again.

B. Implementation details of the prediction program

First, we discuss data access and storage. We use HDFS file directory as a data source for the prediction system and use the output directory of the previous data preprocessing as the data source for the prediction system. The output of the predictive system for one execution is the time required for all buses that did not reach the terminal to reach the subsequent stations. When all buses arrive at each bus stop, the results need to be converted to an RDD type after the prediction is completed, and the Spark streaming will output the stream calculation results.

The more particles there are, the better the distribution of the characterization is. This paper sets the number of particles to 2000. The number of iterations is the maximum number of all key trips in the same direction in the same line. After preliminary statistics on the time that the bus passes through the first key road segment, we find that the value range is (0,700]. Assuming such value obeys homogeneous distribution, all of these values in initial particle groups could be assigned as a random number between [1,700].

The prediction is that each bus starts from the initial stop. During the prediction process, for scenarios in which buses have actually traveled on the road, the particle swarm and the predicted value will take the actual arrival time of the bus. For each key point that has not been reached, the arrival time is predicted using the state equation in the model. However, specific bus locations after discretization cannot be obtained directly. Therefore, it is necessary to obtain the location of the bus at the current moment, and the prediction value before this location takes the true value.

In our prediction model, the travel time of the next critical section is calculated from the key road segment distance and the weighted speed. However, we consider that the instantaneous speed of the bus has limitations for continuously predicting the travel time of multiple road segments. so it is necessary to determine the distance and range of instantaneous speed of buses via experiments.

V. Experiments Analysis and Evaluation

The experimental data used in this paper are actual bus driving data from the Hohhot (moderate city in China), and Qingcheng Line 1 is selected as a representative. The entire line runs through the city center, during which the bus stops are representative of congested roads or squares where the people flows and traffic flows are the greatest. The total length of Qingcheng Line 1 is approximately 9.3 kilometers, and there are 17 stops in the upstream and downstream directions. We predicts the bus arrival time at the peak period and nonpeak period and evaluates the experimental results by the mean absolute error (MAE). MAE can intuitively reflect the degree of deviation between the experimental results and the actual values, and its calculation is given as follows.

$$\mathrm{MAE} = \frac{1}{\mathrm{N}}\sum_{i=1}^{N}|\,\mathrm{y_i} - \hat{\mathrm{y}}_\mathrm{i}\,| \tag{7}$$

where N stands for the number of experimental results, y_i stands for the actual time the bus arrived at stop i, and $y_i^{\wedge}$ stands for the predicted time for the bus to arrive at stop i.

The prediction process actually predicts the travel time in the next key road segment according to the driving condition of the previous key road segment of the bus, on the premise that the driving speed of the bus in these two sequence segments remains unchanged. Therefore, the calculation of the speed in the next segment of the bus will be the key to the prediction. The experiment will take each two specific buses from the sample bus line to validate our model.

A. Experiments setup

The input of the system is the real-time GPS driving record of each route in different driving directions. The system output is the driving time of each bus stop after all the buses have reached the terminal in different directions in different driving directions. In the experiment, HDFS is selected as the data access basis. In practical applications, Kafka is used as a data access tool.

We select four buses with departure times of 16:34 pm, 16:30 pm, 18:51 pm, and 18:39 pm on December 7th, 2016. They are used as the bus that needs prediction and the latest previous bus in the peak time period and nonpeak time period, respectively. In addition, several other bus driving records are selected as historical record data. However, in the actual calculation process, because the latest previous bus may not reach the last stop, it is not unique with respect to the bus to be predicted. In the experiment, predicting the arrival time for the most recent stops, the bus traffic data of the previous two buses are used, while predicting the arrival time for other following stops, historical data of a relatively close bus are used.

The driving records of the above four busses are divided into six parts in sequence and are uploaded to the designated directory in the HDFS at regular time intervals during system operation to simulate the data access of the streaming computing. The paper assumes that the current position of the bus to be predicted is located between the 6th and 7th stations in the initial state, and the last bus is located between the 10th and 11th stations. The historical data are uploaded to the HDFS along with the first driving record. Afterwards, it waits for a stream calculation to be completed at a certain time before uploading the next driving record, so that when all 6 driving records are uploaded to HDFS, the system completes 6 predictions before and after. We take 12 predictions for each of the 6 predictions in the peak time period and nonpeak time period and 77 arrival times (38 in the peak period and 39 in the nonpeak period). The optimal parameter settings for each speed are determined based on the calculated MAE.

B. Determining the best value of the key parameters

The parameters that need to be determined are the speed weight coefficients *a*, *b*, and *c*, which represent the historical speed weights in the next key road segment, the weights of the latest previous bus speed, and the historical speed weights in the following key segments. However, before determining these coefficients values, how to use instantaneous speed to promote better accuracy needs to be experimentally analyzed. So, The this subsection is divided into two parts. First, we determine the optimal use range of instantaneous speed, and then, we determine the optimal values of the speed weight coefficients *a*, *b*, and *c*. Based on these analysis, it is possible to finally determine the time difference that may occur due to different speeds when constructing observations.

First, after performing sufficient experiments, we find that instantaneous speed only plays a role in a very short range, whether it is a peak or a nonpeak period. Therefore, we use the instantaneous speed as the weighted speed only in the first 150 meters, which is the first key road segment.

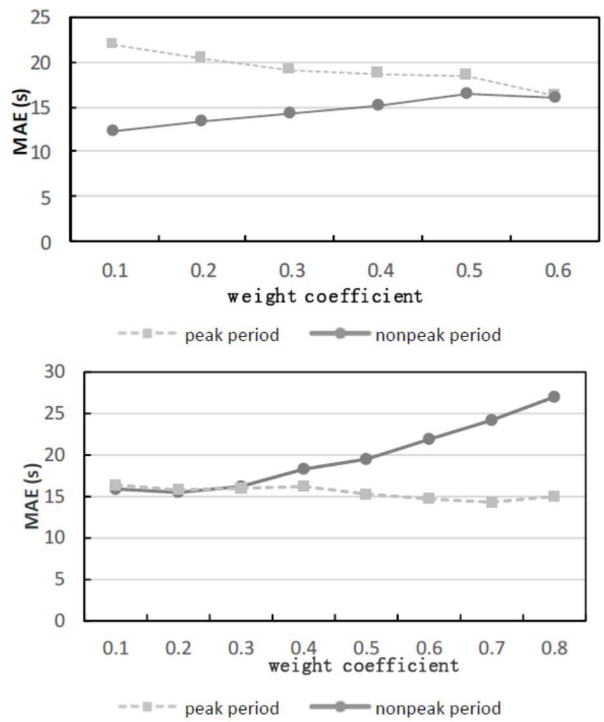

Fig. 3. Prediction error of the first key road segment

Second, from the results shown in Figure 3, it can be seen that within first key road segment, as the instantaneous weight coefficient increases, the prediction error for the peak period gradually decreases and the prediction error for the nonpeak period gradually decreases. To reduce the prediction error during peak hours, we choose to reduce the proportion of historical average speed, so *c* is set to 0.1. On this basis, we find that the best value of *b* is 0.2. Thus, in the first key segment, the optimal value combination of parameters can be finalized as *a*=0.7, *b*=0.2, *c*=0.1.

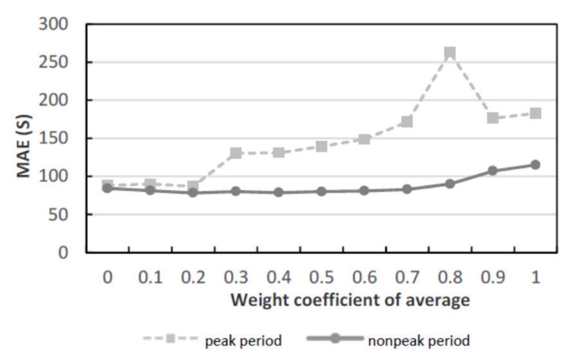

Fig. 4. Prediction error of the following key road segments

Finally, in prediction for the following key road segments after the first one, *a*=0 is determined because the instantaneous speed is no longer used as part of the weighted speed. Based on the above mentioned similar experimental methods and results, as shown in Figure 4, when the historical average speed weight *c* is set to 0.2, the nonpeak and peak time prediction errors take a minimum of 78.2 seconds and 87 seconds, respectively. Therefore, in the remaining road segments, after the first key road segment is determined, the optimal parameter combination is determined to be a=0, b=0.8, and c=0.2.

In addition, in section III-C, we explained that certain time differences must be eliminated, so we uses the average speed from the current position to the first ten key road segments, i.e., 1500 meters, to calculate the time difference.

C. Experiments results and evaluation

According to above coefficients parameters assignments, the prediction results of two buses are analyzed, their respective MAEs are calculated, and the prediction results of the system are compared with the actual data.

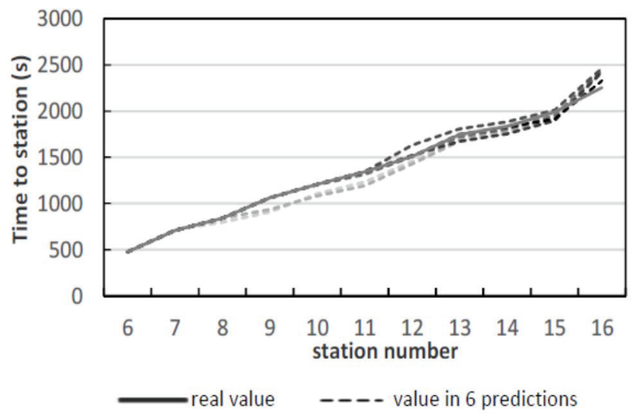

Fig. 5. Prediction results in nonpeak periods

During the nonpeak time period, the comparison between the six predicted values and the actual results is shown in Figure 5. Among all the predicted arrival times (77), the maximum absolute error is 207 seconds, and the MAE of all prediction results is 78.16 seconds. If only the prediction results of the first 6 stations are considered at one time, the MAE is 71.67 seconds. The experimental results are closer to the actual road conditions, and average error of bus arrival time forecasting within 2 minutes is a recognized ideal results.

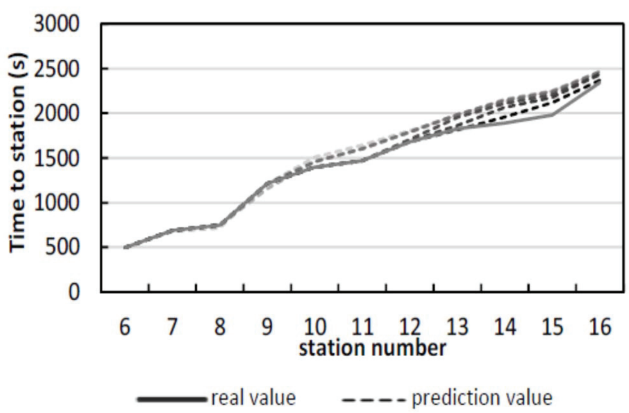

Fig. 6. Prediction results in peak periods

During the peak time period, the comparison between the six predicted values and the actual results is shown in Figure 6. Among all the predicted arrival times (77), the maximum absolute error is 270 seconds, and the MAE of all the prediction results is 112.23 seconds. If only the prediction results of the first 6 stations are taken considered, the MAE is 87.61 seconds. The experimental results are closer to the actual road conditions, and the average absolute error of the prediction results is still within 2 minutes, which satisfies the actual application requirements of passengers.

VI. Conclusion

In this paper, the bus arrival time prediction problem is studied based on the revised particle filter algorithm. By introducing the latest bus speed for collaborative data analysis, a new prediction model is constructed, and then predictive errors and particle optimization problems are solved. Based on the model and the Spark platform, a real-time bus arrival time prediction software system is designed and implemented as a prototype. The experimental results show that our proposed model and prediction system have high prediction accuracy of bus arrival time and fully meet the daily needs of passengers. For future work, we hope to make more efforts to address real-time prediction using a better algorithm [9] and put our system into a larger-scale actual application to improve effectiveness and efficiency.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (No.61662051).

References

[1] J. L. Li, J. Gao, Y. Yang, et al., “Bus Arrival Time Prediction Based On Mixed Model,” China Communications, vol.14, no. 5, pp. 38-47, 2017.

[2] C. Y. Kee, L. P. Wong, A. T. Khader, et al., “Multi-Label Classification of Estimated Time of Arrival with Ensemble Neural Networks in Bus Transportation Network,” in Proc. of the 2nd IEEE International Conference on Intelligent Transportation Engineering, Singapore, 2017, pp.150-154.

[3] F. Xie, J. H. Gu, S. Q. Zhang, “Predicting model of bus arrival time based on MapReduce clustering and neural network,”. Journal of Computer Applications, vol. 37, no.1, pp. 118-122, 2017.

[4] B. Yu, Z. Y. B. Yang, “Bus Arrival Time Prediction Using Support Vector Machines,” Journal of Intelligent Transportation Systems, vol. 10, no. 4, pp.151-158, 2007.

[5] X. Chen, H. Gong, J. Wang, “BRT Vehicle Travel Time Prediction Based on SVM and Kalman Filter,” Journal of Transportation Systems Engineering & Information Technology, vol. 12, no. 4, pp. 29-34, 2012.

[6] B. Dhivyabharathi , B. A. Kumar , L Vanajakshi, “Real Time Bus Arrival Time Prediction System under Indian Traffic Condition,” in Proc. of the 1st IEEE International Conference on Intelligent Transportation Engineering, Singapore, 2016, pp.18-22.

[7] Z. Matei , C. Mosharaf, J. F. Michael, et al., “Spark: Cluster Computing with Working Sets,” in Proc. of the 2nd USENIX Workshop on Hot Topics in Cloud Computing, Boston, 2010, pp.1-7.

[8] F. S. Wang, M. Y. Lu, Q. J. Zhao, et al., “Particle Filter Algorithm,” Chinese Journal of Computers, vol. 37, no. 8, pp. 1679-1694, 2014.

[9] B. A. Kumar, R. Jairam, S. Arkatkar, et al., “Real Time Bus Travel Time Prediction using K-NN Classifier,” International Journal of Transportation Research, vol.20, no. 2, pp. 132-139, 2017.