BCE: A Behavior-learning-based Crowdedness Estimation Mechanism for Crowdsensing Buses

XIAOGUANG NIU1,2, DA SHEN1, ZEZJUN ZHANG1, ZHEN WANG1, JIAWEI WANG1, HAIMING CHEN3

1School of Computer Science, Wuhan University, Wuhan, 430072, China
2Collaborative Innovation Center of Geospatial Technology, 430079, China
3Department of Computer Science, Ningbo University, Ningbo, 315211, China

Corresponding authors: Xiaoguang Niu (xgniu@whu.edu.cn) and Haiming Chen (chenhaiming@nbu.edu.cn)

This work was partially supported by the National Key Research Development Program of China (2016YFB0502201), the National Natural Science Foundation of China NSFC (61872431, 41874031), the Natural Science Fund of Hubei Province (2018CFA007), the Technology Innovation Project of Hubei Province (2018AAA046), the Natural Science Foundation of Zhejiang Province (LY18F020011) and the Natural Science Foundation of Ningbo City (2018A610154).

ABSTRACT This paper aims to develop a method that can accurately estimate the crowdedness level for crowdsensing buses. Multiple related features are reflected by passengers’ moving trajectories at the bus stops. Many state-of-the-art posture recognition approaches have high accuracy, which can ensure that the passengers motion monitoring results are reliable. Through these observations, we propose an improved behavior-learning-based crowdedness estimation mechanism, named BCE, to obtain the crowdedness level of a bus. The motion sequence and gait information of a passenger is obtained via sensors in smartphones and is described by feature vectors. Then the feature vectors are classified as bus crowdedness levels based on gcForest classifier for single-person bus’s crowdedness level estimation and on Recurrent Neural Network (RNN) for multi-people bus’s crowdedness level estimation. Additionally, the moving trajectories and the corresponding crowdedness of the passengers who do not involve in our system can be recognized passively through the motion information of adjacent involved passengers on the bus. The experiments prove that our mechanism achieves an accuracy of 92% overall.

INDEX TERMS activity recognition, hierarchical learning, crowdedness level estimation, crowdsensing, passive perception

I. INTRODUCTION

CROWDEDNESS estimation for buses is mostly addressed in the research field of visual superintendence system [1]–[7], which use a set of cameras or other visual devices to obtain human operations with visual information to quantify the crowdedness of buses. The visual device is efficient in restricted occasions with abundant brightness and wide field of view. Poor brightness and bad vision cause for insufficient and defective visual information, which makes the estimation of crowdedness unacceptable. With these findings, we focus on another research field of visual-free data processing and weak signal processing using mobile devices, aiming to capture informative details in these invisible signal data. Nowadays, travel-related apps offer convenient real-time information (i.e., location, velocity, arrival time, etc.) of buses. However, none of them can provide the crowdedness information of buses. Passengers’ motion (such as the walking time, the average stride while walking, moving trajectory and so on) of getting on the bus can reveal whether it is crowded or not inside the bus. Since the motion of passengers can be recognized with existing high-precision sensors on smartphones, it is attainable to use sensors on the phone to estimate passengers’ behavior and deduce the crowdedness level of a busy bus.

In this paper, we propose a behavior-learning-based crowdedness estimation mechanism for crowdsensing buses, namely BCE, which exploits a trained 3-layer crowdedness classification framework to obtain the crowdedness level of a bus. The participating passengers who are involved in the crowdedness measurement task act as behavior sensors. The
sensors can perceive both participating passengers and the other uninvolved passengers for the crowdedness measurement inside buses. Once passengers get on a bus, sensors begin collecting data. According to different perceived objects, the perception in BCE is divided into active perception and passive perception. As for active perception, sensors collect posture information including motion and gait of participating passengers for BCE. Firstly, considering the monolayer support vector machine (SVM) classifier may not take the semantic relation into account and the hidden Markov model (HMM) [8] can make up for the disadvantages of the SVM classifier. An SVM-HMM classifier is trained to recognize motion. To obtain detailed gait information, a method with a low error rate proposed in [5] is utilized in our system. Then the posture information of an involved passenger from the time when the bus stops to the time when the passenger reaches a stable state at the bus is obtained and will be described by a feature vector. The feature vector of one passenger is the input of the second layer gcForest cascade framework trained in advance to extract the informative contexts between a sequence of postures or a motion. In passive perception, the moving trajectories of uninvolved passengers can be expressed by the behavior fluctuations of the adjacent active involved passengers. When the bus leaves for the next stop, the data of involved passengers on the bus is traversed for a possibly existed behavior fluctuation sequence. Once the sequence is obtained, the posture information of the passively recognized passenger can be calculated based on the attributes of these behavior fluctuations. Eventually, the third layer multi-people Long Short-Term Memory (mLSTM) network will comprehensively utilize the original data concatenated with informative contexts, obtained from the gcForest framework, and passive feature vector obtained from the passive perception to estimate the whole bus crowdedness level.

The major contributions of this paper consist of the following aspects:

1. Involved passengers are utilized as sensors. It can not only get passengers’ behavior information directly from the participants’ sensing data but also recognize the uninvolved passengers’ information through processing the data provided by involved passengers.

2. We propose the BCE mechanism, an improved trained 3-layer crowdedness classification framework based on SVM-HMM, named gcForest, and LSTM to estimate crowdsensing buses’ crowdedness level accurately with several participants’ sensor data.

3. The BCE performs better than some existing methods in some cases according to the experiments below. It proves the BCE is feasible and efficient to give an estimation of the actual crowdedness level of a bus.

The following sections are organized as follows: Related works and preliminaries are shown in Section 2 and 3 respectively. Section 4 introduces the system design of BCE. In section 5, we evaluate the performance of the BCE. Then the conclusions are made in Section 6.
A deep metric learning-based regression method is proposed by Wang et al. [2] to extract density related features, and learn better distance measurement, simultaneously, the effectiveness of which is proved. Depatla et al. [3] and Liu [21] both achieved crowd counting utilizing WiFi signals. Depatla et al. [3] showed how to characterize the wireless received power measurements as a superposition of renewal-type processes, by borrowing theories from the renewal-process literature, they then showed how the probability mass function of the inter-event times carries vital information on the number of people. Based on the key intuition that now that it is too complex to model the crowd counting using Wi-Fi directly, Liu et al. [21] can use deep learning approaches to construct a complex function to fit the correlation between the number of people and Channel State Information (CSI) values with the accuracy of 82.3% in a rather effective and robust manner. Pipelidis et al. [4] proposed a novel approach for the extraction of reference pressure during the outdoor-to-indoor-transition of the user inside a building, which is identified through sensor fusion. Sindagi et al. [6] proposed a novel end-to-end cascaded network of CNNs to jointly learn crowd count classification and density map estimation. Switching convolutional neural network present by Sam et al. [7] could leverage variation of crowd density within an image to improve the accuracy and localization of the predicted crowd count. The framework combined of patches from a grid within a crowd scene and independent CNN regressors gets better performance compared to current state-of-the-art methods. Qi [22] proposed a submodular method to select the most informative frames from the image sequences of crowds. The method selects the most representative images to guarantee the information coverage, by maximizing the similarities between the group of selected images and the image sequence, which achieves higher accuracy compared with the state-of-the-art regression methods and competitive performance with deep convolutional models. Shami [23] regarded a head detector as a key feature and proposed a state-of-the-art convolutional neural network for the sparse head detection in dense crowd, which behave better on UCF_CC_50, ShanghaiTech, and AHU-Crowd datasets. Hu et al. [24] investigated the use of a deep-learning approach to estimate the number of individuals presented in a mid-level or high-level crowd visible in a single image. Xiong et al. [25] proposed a variant of a recent deep learning model called convolutional LSTM (ConvLSTM) for crowd counting, which fully captures both spatial and temporal dependencies. Vandoni et al. [26] proposed a learning-to-count strategy with a generic detection algorithm which benefits from a counting regressor in order to identify crowded subregions with inadequate head detection performance. The experiment results showed the effectiveness with a count error of less than 5%.

III. PRELIMINARIES

A brief introduction to the crowdsensing bus, followed by definitions of some concept arose in this paper is given in this section.

A. INTRODUCTION ABOUT CROWDSENSING BUS

The crowdsensing bus [17], [27] system consists of a sensor data server, a wireless network and multiple sensor nodes. A sensor node is a mobile phone that passenger possesses on a bus. Each node should be allowed to collect location data and other sensor data and should be willing to send information to the data server. The sensor data server is responsible for the crowdsensing bus system, receiving and processing data from each sensor node.

Once the sensor data is received from participating passengers, the server would cluster the nodes into a group, which means that the bus estimates the degree of crowdedness in bus through all data received from nodes in the same group.

In fact, the amount of sensor data required in a bus crowdedness estimating task is very large. Therefore, data sources are a daunting problem. The crowdsensing bus system can perform more data collection tasks at lower cost and achieve higher quality data.

B. DEFINITIONS

Our method introduces four definitions for further comprehend.

Definition 1(stable state). A stable state means a final state of an involved passenger at the bus stops, which is the state of sitting or standing for more than three seconds. As for passengers-off, they cannot be sensed after they get off the bus. We set the state before they get off as the final stable state.

Definition 2(motion sequence). A motion sequence is a sequence of postures, consisting of different activities including walking, standing, sitting, going downstairs and upstairs, which are listed in chronological order.

Definition 3(bus crowdedness level). Considering the personal crowdedness, levels can only show the congestion situation of an area. To quantify the congested situation of one bus, we define 4 crowdedness levels: Level 1 means almost no passengers or only a few passengers on the bus; Level 2 means the number of passengers is roughly the same as that of the seats; Level 3 means about a quarter of passengers are standing on the bus and walking is affected obviously; Level 4 means it is hard to walk.

Definition 4(behavior fluctuation). Uninvolved passengers’ behavior which cannot be directly sensed will influence and change involved passengers’ state. Involved passengers may sway or move when a uninvolved passenger passed by even they have already reached the stable state. We define this change as behavior fluctuation and quantify it by gyroscope data. If waves simultaneously take place in three axes of the gyroscope sensor, a behavior fluctuation is recognized. The details will be represented in chapter 4.5.
A. SYSTEM OVERVIEW

In this paper, we proposed a behavior-learning-based crowdedness estimation mechanism for crowdsensing buses, namely BCE, to obtain the quantified bus crowdedness level. Figure 1 sketches the architecture of our system. There are two major components.

On the mobile phone side, as shown in Figure 1, the mobile phone carried by the participants at the bus stations can continuously collect sensor data. After performing posture data preprocessing and posture recognition on the mobile phone, the phone will send the individual posture feature vector to the data server. The mobile phone performs data collection, transmission, and part of data processing.

The other part is the data server side. The server is primarily responsible for accepting and processing data sent from participant’s mobile phones. This component consists of passive perception, single-person, and multi-people bus crowdedness level estimation.

In BCE, the collected data from participants can be divided into three moving states including getting on, getting off, and walking on the bus, which can be identified according to the action of going up and down stairs. We refer to three moving states as passenger-on, passenger-off and passenger-walking. First, we process the raw data to get some informative features. Then the features will be selected as the input of the SVM-HMM classifier model for training. After motion recognition and gait recognition, we obtain the moving trajectory (i.e. the motion sequence and the gait information). All the works mentioned above are completed on participant’s mobile phones. And when the bus moves on, the mobile phone will send the obtained individual feature vector $v = [v_1, v_2, \ldots, v_t]$ consisting of posture motion and gait information to the data server. The server will timely cluster the received data into different groups (i.e. different buses) according to the attached location information. Then the server may launch the process of passive perception, as well as the single-person bus crowdedness estimation based on gcForest classifier and multi-person bus crowdedness estimation using the mLSTM network in sequence.

B. DATA PREPROCESS

In this part, we invited eighteen males and eighteen females of different ages between 18 and 31 to collect data. Five actions, including walking, standing, going upstairs, going downstairs and sitting, are performed by every participant. Each process of collecting training data merely contains one action lasting more than 60s. Subsequently, each collected training data set is labeled with corresponding actions. The sampling frequency is set to 200Hz. And the sensor data of this part is collected to train the SVM-HMM model.

Constrained to the ability of process data on mobile phones, a simple algorithm is utilized to eliminate stochastic noise of the raw data before the process of feature extraction. Recursion average filtering method has high smoothness and good effect on recurrent noise. The method is shown as Eq. 1 in which the $x(i)$ means the collected sensor data. Figure 2 demonstrates the effectiveness of this method for disposing the acceleration data.
maximum, RMS (Root Mean Square), standard deviation of every axis (36 features), correlation between each two axis (3 features), VarAPF(1 feature) [9]. After labeling every feature vector with the corresponding action, a SVM classifier is trained for motion recognition.

\[
SMA = \frac{1}{t} \int_{t}^{t+\Delta t} x(t) dt + \int_{t}^{t+\Delta t} y(t) dt + \int_{t}^{t+\Delta t} z(t) dt
\]

The input of the HMM model’s Viterbi algorithm is required to be probability values, while the output of SVM is a definitive number. There is a gap between the SVM classifier and the HMM model in the training process in BCE. Here, we adopt the sigmoid algorithm to map the number to a probability, which is shown in Eq.3.

\[
p(j|x) = \frac{1}{1 + A \exp(-y_i + B)}
\]

where j denotes j-th motion, x denotes the feature vector, yi denotes the SVM output of the i-th kind motion. Parameter A and B can be estimated using cross-validation. In this paper we need to distinguish 5 states, hence 5 SVM classifiers are trained to distinguish one behavior from the other four activities. Every SVM classifier has two outputs (1,-1) to express whether the current sample belongs to this motion. Analogously 5 HMM models must be trained to correspond to 5 motions. While training a HMM model of one motion, a sample data belonging to the motion is inputted into the 5 SVM models then we get 5 integer results. After converting the integer into probability value, a vector is obtained. Using the same handler for all sample data belonging to a motion, a vector sequence is gained. A HMM model corresponding to a motion is trained using the appropriate motion vector sequence. The parameters of every HMM model can be determined by the Baum-Welch algorithm. Thus, the output probability can be calculated by Eq.4 in which n denotes the number of SVM classifiers. In BCE, n = 5.

\[
p(x|S_i) \propto \sum_{j=1}^{n} c_{ij} \cdot \frac{p(j|x)}{p(j)}
\]

After the motion recognition model is trained, the estimation process becomes simpler. At first, input the feature vector of segmented sample data into the SVM models and a series of classification results will be obtained. After converting the integers into probability values and inputting the probability values into the HMM models, the motion which has the biggest probability value is chosen as the final estimation result. We take probabilities of all postures in a time-step as a feature vector \(p_i\) for the following classification. Additionally, we send the data labeled ‘walking’ into the gait recognition method.

2) Gait recognition

In BCE, data sensed from acceleration and gyroscope sensor are utilized to calculate the gait information \(g_t=|\text{step number}|\).
ber, average stride, stride variance, average swinging range], which has been demonstrated to be remarkably accurate as indicated in [5].

This section is divided into two parts: one is step counting and stride estimation, the other is swinging times counting. The whole framework of gait recognition is shown in Figure 4.

i) Step counting and stride estimation

For step counting and stride estimation, BCE incorporates the method in [5], with the accuracy of over 94% in step counting and maximum error less than 8.7 cm in stride estimation.

ii) Swinging times counting

Swinging means the turn of moving direction. As is shown in Figure 5, every obvious peak represents a swinging. The height of the peak shows the range of change in some way. In case of the effect of the wrong estimations caused by noise peaks, a proper threshold is adopted to increase the validity of the algorithm. Any peak whose absolute height does not break the threshold will not be accepted as a heading state change motion. So the core of this part is finding and recording the qualified peak numbers and the average of the peaks' absolute values of each step returned by the step counting algorithm.

D. SINGLE-PERSON BUS CROWDEDNESS LEVEL ESTIMATION

Normally the crowdedness level is expressed as the total number of people in a specific area. However, all the existing works about passengers-counting on a bus are based on image recognition, which may have many difficulties to be realized in practice. We find that passengers' walking trajectories can show the crowdedness level in some ways, for example, the bus where a passenger can sit down in a short time must be more spacious than the one where a passenger walks a long time in short steps then stand in the end. Even if the passengers' stable states for all cases are standing, it is obvious that the bus with straight moving trajectories for all the passengers may be less crowded than the bus where the passengers have to make some micro-turns while moving.

So, it's possible to estimate the crowdedness level on the bus by the passenger's motion sequence at bus stops. When there are a few active passengers, gcForest performs better than other neural networks. A brief introduction to gcForest and approach to use it will be described in this section.

1) Data preprocessing

The single-person bus crowdedness estimation in BCE is based on gcForest classifier. After the posture recognition process, we concatenate \( p_t \) and \( g_t \) into a feature vector \( v_t \), and we combine all feature vectors at each time-step into a whole feature vector \( v \) for one single passenger, which will be denoted as \( v \) in Section 4.6.

2) gcForest

A gcForest consists of an input and an output layer, as well as multiple hidden layers. Different from other neural networks, the hidden layers are the cascade forest structure. As illustrated in Figure 6, each level consists of different decision tree forests. In this case, we take two completely-random tree forests and two random forests in one level, and each forest will output a four-dimensional class vector, which is concatenated for the original input feature vector.

In addition, a gcForest contains a multi-grained scanning layer, which can handle sequence data like recurrent neural networks. As Figure 7 shows, sliding windows are used to scan the raw features. In our case, we collect data once 3 seconds and the last 2 minutes. Thus there are 40 nine-dimensional vectors which can be reshaped as one 360-dim vector, and a window size of 26 features are used. A 26-dim feature vector will be generated by sliding the window for one feature and 335 feature vectors are produced totally. All the features vectors are used to train a completely-random tree forest and a random forest, and then the class vectors are generated and concatenated as transformed features, which are input feature vectors in the cascade forest. Additionally, if there is more than one type of sliding window, multiple sliding windows can be used to produce multiple outputs, which are used in the following cascade forests in turn.
3) Implementation details

gcForest has fewer hyper-parameters than other traditional deep neural networks. If we want to apply the gcForest to our feature extracting task, it is still necessary to adjust its parameters to optimize it. Thus, we make some adjustment to gcForest as described below.

1) In multi-grained scanning, formally, we adjust the size and number of sliding window and the type and number of forests to keep the feature vectors produced having proper dimension.

2) In the cascade forest, we can adjust the type and number of forests to make the final loss acceptable.

3) In this case, we select four different sizes of sliding windows, respectively 1*6, 1*9, 1*13, 1*26 and group a random forest and a completely-random tree forest in multi-grained scanning. In cascade forests, two random forests and two completely-random tree forests for each layer are adapted and we set 90% as the final expected estimation accuracy.

E. PASSIVE PERCEPTION

Before passive perception, we need to obtain behavior fluctuations. According to definition 4, we analyze gyroscope data to find behavior fluctuations. As Figure 8 shown, when $L_2$ norm or the root of the squares sum of the three peak values is over a threshold, we recognize it as a behavior fluctuation. The area is the region between two nearest peak point and we calculate $L_2$ norm of the area blew the three peak values by integration as the magnitude of the behavior fluctuation. The threshold should be chosen to reflect real behavior fluctuations in adjacent participatory passengers. In our experiments, the threshold is 2.4 and we can get four behavior fluctuations pointed in Figure 8.

Once the bus moves on, the server will receive a series of structured data, which can describe the congestion situation of the vicinity of every involved passenger on the bus. Such structured data consists of the individual crowdedness levels of current involved passengers, the credibility of the person estimation result, the occurrence relative time of the estimation result, the waiting time from reaching the stable state to the bus moving on, the relative location of the involved passengers (front, middle or rear), the behavior fluctuation times and the attribute (i.e. occurrence time, fluctuation range, etc.) of every behavior fluctuation. By analyzing the behavior fluctuations from the adjacent involved passengers on the same bus, we can recognize the walking track of the uninvolved passenger. If there exists at least one behavior fluctuation, we regard it that a new uninvolved passenger is recognized. Furthermore, the posture information of the uninvolved can be deduced referring to the definition of feature vector $v_t$ mentioned above. More specifically, the walking velocity and final location of the new uninvolved passenger can be roughly estimated. Since the locations of the involved passengers who have already reach the stable state are known, combining the time differences of the fluctuations’ occurrence time and the known distances of the previous passengers, the stride can be calculated. The walking time, total time and total step number can be obtained using the average stride and the final location. Standing time is set as 0. Swinging times and average swinging range are obtained through the information of the fluctuations which consist of the motion sequence that is regarded as a passively-recognized passenger. Then the vector $g_t$ consisting of step number, average stride, stride variance...
FIGURE 8. the gyroscope data and $L_2$ norm with threshold while behavior fluctuations occur

and the average swinging range is obtained. Considering there exists many behavior fluctuations during the whole moving process, in other words, a behavior fluctuation cannot take place on a bus where all the passengers are sitting, the final stable state of the passively-recognized passenger is set as standing. That is to say, during the process of passive perception, the recognized uninvolving passenger is regarded as walking all the way then stand in the end. Considering the definition of $p_t$ is that the probabilities of all postures, we set the probability value of motion "walk" satisfying normal distribution during the whole passive perception process. Meanwhile, the probabilities of other motions are set as the same value to ensure the sum of the five probabilities is 1.

Then a process of passive perception is completed. Concatenate $p_t$ and $g_t$ into a feature vector $v_t$ and then combine all the $v_t$ as $v_p$, which will be a part of the input in the multi-people LSTM network, as represented in Section 4.6.

FIGURE 9. Basic LSTM

$O = \text{softmax}(Vh_T + c)$

FIGURE 10. The structure of LSTM unit

F. BUS CROWDEDNESS LEVEL ESTIMATION

To make the best of the information hidden in both active and passive feature vectors, we utilize a deep learning framework based on RNN to estimate the bus crowdedness level. Although both based on semantic relation, RNN can take more early information into consideration than the HMM model. In addition, RNN can handle input data of arbitrary length, which is very suitable for our demand of an uncertain number of sensed passengers on one bus. The framework is shown in Figure 9. The underlying network is a standard LSTM network, then the results will be followed by a softmax function to estimate the bus crowdedness level.

LSTM networks have been successfully used in isolated sequences like handwriting [28] and speech [23]. Inspired by this, we build a LSTM based model for our bus crowdedness level estimation problem as well. We develop one LSTM for each active passenger and estimate crowdedness level as shown in Figure 9.

The LSTM unit is shown in Figure 10. In standard RNN, every unit has an identical and simple structure to try its best to make use of the information input before while quickly forget the information hidden in the earliest input although repeat in the meanwhile. Thus, the structure of every repeat unit in LSTM is designed to be different and able to analyze the inputs covering longer time periods. The core of the LSTM unit is the cell state, and the information goes through every unit with little linear interaction to keep its invariability. The forget gate decides which part of the information will be discarded from the current cell state and the update gate decides which part of the new input will be updated into the current cell state. Finally based on the results mentioned above and the tanh and sigmoid layer it completes this update action. Then a result via sigmoid and tanh layer will be the output of this part. The cell state continues to be passed once a new input is received. For this paper, we build our LSTM architecture through this hidden layer function $L$, which is implemented by the following
composite functions (Eq.5, Eq.6, Eq.7, Eq.8, Eq.9):

\[ i_t = \sigma \left( W^i_x x_t + W^i_c h_{t-1} + W^i_b c_{t-1} + b^i \right) \] (5)

\[ f_t = \sigma \left( W^f_x x_t + W^f_c h_{t-1} + W^f_b c_{t-1} + b^f \right) \] (6)

\[ c_t = f_t c_{t-1} + i_t \tanh \left( W^c_x x_t + W^c_c h_{t-1} + b^c \right) \] (7)

\[ o_t = \sigma \left( W^o_x x_t + W^o_c h_{t-1} + W^o_b c_{t-1} + b^o \right) \] (8)

\[ h_t = o_t \tanh (c_t) \] (9)

However, the naïve use of one single LSTM model per person does not capture the interaction of people on a bus. The original LSTM is agnostic to the influence of other passengers. We address this limitation by connecting all LSTMs through a new shared strategy visualized in Figure 10.

**Sharing Unit.** We expect the hidden states of an LSTM to capture the time-varying motion-properties and interaction between passengers. In order to obtain the interaction, we share the states between all LSTMs. Hence, we need a compact representation which combines the information from all neighboring states. We handle this by introducing Sharing Units as shown in Figure 11. At each time-step, the LSTM cell receives shared hidden-state information from the LSTM cells of other passengers. While sharing the information, we try to preserve the interaction information as shown below.

The hidden state \( h^i_t \) of the LSTM at time \( t \) captures the latent representation of the \( i \)th active passengers on the bus. We share this representation with other LSTMs by building a shared unit or parameter \( C_t \). Given a hidden-state dimension \( D \), and number of active passengers \( N_p \), Eq. refeq:shared is the presentation of \( C_t \).

\[ c_t = \Phi \left( \sum_{j=1}^{N_p} c^j_{t-1} h^j_{t-1}; W_c \right) \] (10)

where \( h^j_{t-1} \) is the hidden state of the LSTM corresponding to the \( j \)th passengers at \( t-1 \), \( c^j_{t-1} \) is the corresponding crowdedness level, \( c^j_{t-1} = [1, 2, 3, 4] \), \( c^j_{t-1} h^j_{t-1} \) is the contribution made by the \( j \)th active passengers at time-step \( t-1 \), and \( \phi(\cdot) \) is an embedding function with ReLU non-linearity. \( W_c \) is an embedding weight.

We embed the shared unit \( C_t \) and original feature vector into a new input vector \( x^i_t \). These embeddings are concatenated and used as the input to the LSTM cell of the corresponding passengers at time \( t \). This introduces the following representation (Eq.11, Eq.12):

\[ x^i_t = \left[ \Phi(v^i_t; W_v), C_t \right] \] (11)

\[ h^i_t = LSTM \left( h^i_{t-1}, x^i_t; W_l \right) \] (12)

where \( \phi(\cdot) \) is an embedding function with ReLU non-linearity, and \( W_v \) is an embedding weight. The LSTM weights are denoted by \( W_l \).

The parameters of the LSTM model are learned by minimizing the negative log-Likelihood loss \( L \) (Eq.13):

\[ x^i_t = \left[ \Phi(v^i_t; W_v), \Phi \left( \sum_{j=1}^{N_p} c^j_{t-1} h^j_{t-1}; W_c \right) \right] \] (13)

\[ \Phi \left( v^i_t, \sum_{j=1}^{N_p} c^j_{t-1} h^j_{t-1}; W_{v,c} \right) \]

where \( \phi(\cdot) \) is an embedding function with ReLU non-linearity, and \( W_{v,c} \) is generated by concatenating \( W_v \) and \( W_c \).

Thus \( W_{v,c} \) can be recognized with weights of a hidden layer in a LSTM network rather than additional parameters. We jointly back-propagate through multiple LSTMs at every time-step.

**Data Augmentation.** To improve the performance of the mLSTM network and to prevent overfitting, we have artificially enlarged the scale of training datasets by data augmentation. Data augmentation is the application of one or more deformations applied to the labeled data without changing the semantic meaning of the labels. In our case, we add Gaussian noise with appropriate variance to generate a different input feature but with equal semantic information.

**Implementation details.** At each time-step, we use a dimension of 5 for the motion-property and 4 for the gait information before using them as input to the LSTM. Also, we embed the shared unit \( C_t \) into the input. We set the fixed hidden state dimension to be 6 for all the LSTM models. Additionally, we can set \( C_0 \) by calculating all passengers’ contributions through single-person predicting results.

**V. EVALUATIONS**

We will evaluate the accuracy of the BCE in this section. In these experiments, 58 volunteers from Wuhan University were selected to collect data. Each volunteer holds a mobile phone in his hand and collects data. Data was collected including volunteers getting on the bus, getting off the bus, and waiting in line around the bus. Considering that young people give seats to the elderly can significantly affect our experimental results and reduce the accuracy of the method. There are no elderly groups in the volunteers we selected.

We embed the shared unit \( C_t \) and original feature vector into a new input vector \( x^i_t \). These embeddings are concatenated and used as the input to the LSTM cell of the corresponding passengers at time \( t \). This introduces the following representation (Eq.11, Eq.12):

\[ x^i_t = \left[ \Phi(v^i_t; W_v), C_t \right] \] (11)

\[ h^i_t = LSTM \left( h^i_{t-1}, x^i_t; W_l \right) \] (12)

where \( \phi(\cdot) \) is an embedding function with ReLU non-linearity, and \( W_v \) is an embedding weight. The LSTM weights are denoted by \( W_l \).

The parameters of the LSTM model are learned by minimizing the negative log-Likelihood loss \( L \) (Eq.13):

\[ x^i_t = \left[ \Phi(v^i_t; W_v), \Phi \left( \sum_{j=1}^{N_p} c^j_{t-1} h^j_{t-1}; W_c \right) \right] \] (13)

\[ \Phi \left( v^i_t, \sum_{j=1}^{N_p} c^j_{t-1} h^j_{t-1}; W_{v,c} \right) \]

where \( \phi(\cdot) \) is an embedding function with ReLU non-linearity, and \( W_{v,c} \) is generated by concatenating \( W_v \) and \( W_c \).

Thus \( W_{v,c} \) can be recognized with weights of a hidden layer in a LSTM network rather than additional parameters. We jointly back-propagate through multiple LSTMs at every time-step.

**Data Augmentation.** To improve the performance of the mLSTM network and to prevent overfitting, we have artificially enlarged the scale of training datasets by data augmentation. Data augmentation is the application of one or more deformations applied to the labeled data without changing the semantic meaning of the labels. In our case, we add Gaussian noise with appropriate variance to generate a different input feature but with equal semantic information.

**Implementation details.** At each time-step, we use a dimension of 5 for the motion-property and 4 for the gait information before using them as input to the LSTM. Also, we embed the shared unit \( C_t \) into the input. We set the fixed hidden state dimension to be 6 for all the LSTM models. Additionally, we can set \( C_0 \) by calculating all passengers’ contributions through single-person predicting results.

**V. EVALUATIONS**

We will evaluate the accuracy of the BCE in this section. In these experiments, 58 volunteers from Wuhan University were selected to collect data. Each volunteer holds a mobile phone in his hand and collects data. Data was collected including volunteers getting on the bus, getting off the bus, and waiting in line around the bus. Considering that young people give seats to the elderly can significantly affect our experimental results and reduce the accuracy of the method. There are no elderly groups in the volunteers we selected. The bus route map is shown in Figure.12. There were 20 volunteers on each line to collect data, and we collected 2,000
data in one day. The data collection work lasted for two months, and the data covered various car crowdedness situations, taking into account many factors (such as time, traffic, etc.). We tag the data according to the level of crowdedness to detect the accuracy of the BCE method. In addition, we used 70% of the collected data as the training set, 20% as the verification set, and the rest as the test set. The experimental results and related instructions will be shown in the following section.

A. EVALUATION OF POSTURE RECOGNITION
1) Motion Recognition
The accuracy of SVM, HMM-SVM, and SVM-HMM classifiers in this evaluation is shown in Table 1. As shown in Table 1, five motions can be recognized with high accuracy. So, the motion sequence obtained is reliable enough to ensure the validity of the data input into the second layer method.

Since the recognition accuracy for sitting is lower than for standing, the reliability of the samples whose stable state is sitting may be lower than those whose stable state is standing.

### TABLE 1. The results of motion recognition (% of Estimation Accuracy)

<table>
<thead>
<tr>
<th></th>
<th>SVM</th>
<th>HMM-SVM [8]</th>
<th>SVM-HMM(BCE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Walking</td>
<td>95.1</td>
<td>96.5</td>
<td>97.8</td>
</tr>
<tr>
<td>Upstairs</td>
<td>95.4</td>
<td>97.7</td>
<td>99.1</td>
</tr>
<tr>
<td>Downstairs</td>
<td>94.9</td>
<td>96.9</td>
<td>98.3</td>
</tr>
<tr>
<td>Sitting</td>
<td>87.9</td>
<td>91.8</td>
<td>93.2</td>
</tr>
<tr>
<td>Standing</td>
<td>92.8</td>
<td>96.6</td>
<td>98.6</td>
</tr>
<tr>
<td>Overall</td>
<td>93.2</td>
<td>95.9</td>
<td>97.4</td>
</tr>
</tbody>
</table>

2) Gait Recognition
Upon the data is collected, some hidden information like the actual step numbers and swinging times of the volunteers are recorded in the collected data. The evaluation of gait recognition includes step counting, swinging times counting, and stride estimation.

a) Step counting and stride estimation
Since the gait cycle count is the basis for the swinging count and the step size estimation, its accuracy directly affects the subsequent recognition accuracy. The results of step counting are credible with an accuracy over 90%. Howbeit compared with the result in [5], the accuracy of the method we proposed is a little lower. The reason for this situation may be that there is some wrong output from motion recognition, in other words, the data labeled ‘walking’ may not correspond to the motion walk, which results in the extra or missing steps.

Everyone’s true stride is measured in advance. Figure 13 shows the stride estimation results. X-axis presents the serial number of participants.

b) Swinging times counting
Based on the gyroscope data for each gait cycle, the swinging times counting algorithm also has the problem mentioned in the step counting algorithm. Figure 14 shows the precision and recall level of this algorithm. X-axis presents the serial number of participants. As shown in this figure, since the corresponding gyroscope data can reflect the swinging motion obviously, the recall accuracy stays 100% for the swinging
B. CROWDEDNESS ESTIMATION

1) Single-person bus crowdedness Level Classification
As shown in Figure 15, the BCE achieves different estimating accuracy in different levels of the bus crowdedness. When the level of the bus crowdedness is 3, the BCE has the highest prediction accuracy. The accuracy of level 1 and level 4 indicates that fewer people and more people may have an impact on the estimation and reduce the accuracy. The variance sizes vary from the bus crowdedness level, which is also the smallest when the level is 3. Figure 16 shows that the accuracy of the crowdedness estimation at four different levels is affected by the passengers’ walking distance. As shown in Figure 16, the relationship between the accuracy and the distance traveled by participants is non-linear. For four different levels of the crowdedness, the highest accuracy occurs at different walking distance. The longer walking distance of the participants results in a lower accuracy when the bus crowdedness level is 1, but when the level is 4, it can form the highest accuracy. The lowest accuracy, which is over 84%, proves the effectiveness of this part.

FIGURE 15. The accuracy of single-person crowdedness level classification

FIGURE 16. Accuracy of single-person crowdedness under four bus crowdedness levels

motion. The accuracy is a bit lower as a result of the ignored and extra step caused by the SVM-HMM model’s wrong judgment.

2) Multi-people Bus Crowdedness Level Classification
According to the experiments result, the average accuracy of the multi-people bus crowdedness level through BCE with passive perception is over 92.8%. Figure 17 shows the accuracy of two multi-people estimations, both are mlSTM networks but one with passive perception involved. As is shown in the figure, the accuracy of mlSTM with passive perception increases more rapidly than the other one when there are several participants. The mlSTM with passive perception performs much better than the other one when the participants are not saturate. With participant people increasing, it also achieves a little higher accuracy than the latter one

FIGURE 17. The accuracy of BCE with passive perception

FIGURE 18. The accuracy for mlSTM with different datasets
VI. CONCLUSIONS AND FUTURE WORK

This paper proposes a behavior-learning-based method, namely BCE, to estimate the crowdedness level on a bus. Compared with the existing equipment-based methods, BCE only relies on motion sensors in the participatory passengers’ smartphone but has better performance. BCE improves HCE by adding passive perception to achieve better accuracy in multi-person bus crowdedness level estimation. The experimental results reveal that the high accuracy motion recognition algorithm based on hierarchical SVM-HMM classifier ensures the input data quality of the gait recognition, which also guarantees the validity of the data input into the gcForest-based single-person crowdedness level estimation.

The uninvolved passenger can be recognized passively from the crowdedness information of the passengers who provide information proactively, then all the passengers’ crowdedness information is utilized in the three layer mLSTM classifier to estimate the multi-person bus crowdedness. We adopted trajectory data sets generated by over 10000 pieces of sensor data to evaluate our mechanism, and the high accuracy of the results proves the feasibility of our system.

In the BCE, we asked participants to hold their phones and keep the phones parallel to the ground. It makes our system less practical because no one will just hold their phones still. Most of them will text messages, browse news or play games with their phones on the bus. In the future work, we need to further preprocess the contaminated mobile phone attitude data to ensure the extracted features high reliability.

In addition, The BCE is only applicable to some traditional buses. The states and postures we defined are constrained to the traditional single-decker buses. We may consider some other activities like going downstairs and upstairs as significant actions if the BCE is built for the double-decker buses.

Also, we plan to extend categories of states and postures with their phones on the bus. In the future work, we need to further preprocess the contaminated mobile phone attitude data to ensure the extracted features high reliability. In addition, The BCE is only applicable to some traditional buses. The states and postures we defined are constrained to the traditional single-decker buses. We may consider some other activities like going downstairs and upstairs as significant actions if the BCE is built for the double-decker buses.

Also, we plan to extend categories of states and postures for different buses to improve the generality and accuracy of BCE.

REFERENCES

JIAWEI WANG received the Beng. degree of computer science in Computer Science School at Wuhan University in 2017. He is currently pursuing the Master degree Computer Science School at Wuhan University. He is the student of Prof. Niu. In this paper, Wang, as one of the researchers, also made parts of contribution. His research interest includes information security, sensor and internet of things and indoor location.

HAIMING CHEN received the BEng and MEng degree in computer engineering from Tianjin University, China, in 2003 and 2006 and the PhD degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences in 2010. He is now an associate professor at the department of computer science, Ningbo University. He has authored or co-authored over 50 papers in areas of wireless, ad hoc, sensor networks and networked embedded computing systems.