Research Article

Haiyang Lyu*, Ke Xu, Donglai Jiao, and Qiqi Zhong

Bump feature detection of the road surface based on the Bi-LSTM

https://doi.org/10.1515/geo-2022-0478
received November 29, 2022; accepted April 12, 2023

Abstract: The road network is the basic facility for transportation systems in the city. Every day, a large number of vehicles move on the road and exert different pressure on the ground, which leads to various problems for the road surface, such as the bump features of the road surface (BFRS). However, traditional methods, such as detecting BFRS manually or with professional equipment, require a lot of professional management and devices. Based on the mobile sensor and the bidirectional long short-term memory (Bi-LSTM), a detection method for BFRS is proposed. The BFRS detection method proposed in this article solves the problem that other BFRS detection methods cannot detect large area road surface efficiently and provides an algorithm idea for efficient detection of large area road surface BFRS. The mobile phone with multi-sensors is carried on vehicles, and the BFRS information is logged during the movements. The orientation of the mobile is computed according to the gyroscope. The actual posture of the acceleration sensor is adjusted with the reference coordinate system, whose *z*-axis is vertical to the ground. This article uses the adjusted acceleration data as the training dataset and labels it according to time stamps and videos recorded by the driving recorder. Finally, the Bi-LSTM is constructed and trained, followed by the BFRS detection. The results show that it can detect BFRS in different regions. The detection accuracy of the campus section and the extended experiment was 92.85 and 87.99%, respectively.

Keywords: bump features of the road surface, Bi-LSTM, smartphone sensors

1 Introduction

The road is the fundamental infrastructure [1,2]. Well-maintained roads can promote social and economic development [3]. However, due to different reasons, such as heavy traffic, overloaded vehicles, adverse weather, and geological impact, the pavement suffered severe damage before reaching its designed life. The road pavement requires substantial maintenance [4]. Such a large road area has brought tremendous pressure on the road department. An efficient, fast, and low-cost bump feature detection method is extremely important for road maintenance work.

Traditional bump feature detection methods, such as the photogrammetry detection method and radar detection method, mainly detect bump features through professional equipment like high-definition and resolution cameras, ground penetrating radar, and light detection and ranging (lidar) [5,6]. These two methods require professional equipment and skills, and the data acquired by these methods are limited to a small area. To improve the efficiency of detecting bump features in a large area, researchers have proposed new methods to detect bump features, including images acquired by usual cameras or different kinds of sensor data to detect bump features [11–22]. The specific method is described in the related research. However, these methods require high-definition images or large amounts of streamed sensor data, which demands huge storage or a high-performance platform. On the other hand, the smartphone is usually equipped with various sensors and has a high sampling rate [7–10]. So, can we use ubiquitous smartphones to detect bump features and solve the above problems?

Hence, the method is proposed to detect bump features based on bidirectional long short-term memory (Bi-LSTM) to obtain road surface information through the built-in sensors of smartphones and is trained to detect bump features in an efficient, fast, and accurate way. As a type of recurrent neural network (RNN), Bi-LSTM has gained significant attention in recent years due to its effectiveness in analyzing sequential data. In particular, Bi-LSTM has been widely used in various applications, such as natural

*** Corresponding author: Haiyang Lyu,** Smart Health Big Data Analysis and Location Services Engineering Research Lab of Jiangsu Province, Nanjing University of Posts and Telecommunications, Nanjing 210023, China; School of Geographic and Biologic Information, Nanjing University of Posts and Telecommunications, Nanjing 210023, China, e-mail: hlyu@njupt.edu.cn

Ke Xu, **Donglai Jiao**, **Qiqi Zhong:** Smart Health Big Data Analysis and Location Services Engineering Research Lab of Jiangsu Province, Nanjing University of Posts and Telecommunications, Nanjing 210023, China; School of Geographic and Biologic Information, Nanjing University of Posts and Telecommunications, Nanjing 210023, China

language processing, speech recognition, and time series prediction. Its popularity is mainly attributed to its ability to capture both forward and backward temporal dependencies in the input sequence, which is achieved by using two separate LSTMs, one running in the forward direction and the other running in the backward direction. This unique architecture allows Bi-LSTM to model long-term dependencies in the input data, making it well-suited for time series analysis. However, how to quickly obtain smartphone sensor data and how to use deep learning will not be extensively discussed in this article.

The remainder of this article is structured as follows. The next section gives an overview of works related to this article. Section 3 provides a detailed description of the composition and implementation of the method in this article. Experiments are conducted. The results are discussed in Section 4, followed by conclusions in Section 5.

2 Related work

The image-based method refers to detecting bump features by analyzing the collected road surface images. Since the situation of the road surface can be intuitively reflected by images, bump features of the road surface (BFRS) can be directly recognized. Pavement cracks are a typical type of bump feature, and a lot of research have been done on related detection methods. Wang et al. [11] proposed a detection method based on the AdaBoost algorithm. It uses OpenCV to train a classifier to identify road features, and the method can detect most pavement cracks successfully. Bump features also include potholes except for pavement cracks. Zhang et al. [12] proposed a pothole detection method based on stereo vision. The method uses an efficient disparity calculation algorithm to generate the disparity map. Then it successfully detects potholes through the distance between the fitted quadratic surface and the road surface. Due to the disadvantage of deep learning requiring a large number of training datasets, Dong et al. [13] proposed a multi-target and few-shot pavement distress detection method based on metric learning. The classification accuracy of both 5-way 1-shot and 5-way 5-shot is highly accurate. Guo [14] developed an intelligent robot based on GPS navigation that combines image recognition and deep learning to detect and locate pavement features. This robot can output video frames containing road diseases. Li et al. [15] used the back propagation neural network (BP neural network) to identify pavement cracks from images. The BP neural network is a multi-layer feed-forward network, and the neural network model is trained by the method of error BP [16,17]. This method divided pavement cracks into three categories: alligator, transversal, and longitudinal cracks. All three crack detection results are highly accurate. Although the image-based methods are accurate and intuitive, they can be affected by factors such as image illumination, occlusion, and road surface water, which limit their ability to identify BFRS. The method proposed in this article can solve this problem very well.

The sensor-based method can detect bump features using spatial-temporal sensor data. Giacomin and Woo [18] proposed a method based on steering wheel vibration feedback to detect road pavement. This method uses an accelerometer and vibration sensor to detect the road pavement, and the results of the experiment can reflect the situation of the road pavement well. Ren et al. [19] selected the data collected by the accelerometer to detect pavement distress. This method is based on the principle of a support vector machine, combined with the accelerometer and GPS sensor, to detect and locate pavement distress, with high detection accuracy. To extend the traditional coverage beyond the automobile vehicle (such as pedestrian streets, bicycle lanes, etc.), Zang et al. [20] used bicycles as a bearing platform, combined with an accelerometer and GPS, to measure the surface roughness. The result of the experiment shows that the International Roughness Index values measured by this method were strongly and positively correlated with those measured by professional instruments. Rather than using the car as a carrier platform for the sensor, Tai et al. [21] chose a motorcycle as a carrier platform. They also combined the accelerometer and GPS sensors to collect data. The data are analyzed to detect road anomalies and evaluate road quality. This method can detect road anomalies successfully and has high accuracy. Li et al. [22] proposed a method to detect road anomalies using continuous wavelet transform (CWT). The basic idea is that: (i) select a mother wavelet that satisfies the conditions, (ii) translate and stretch it to obtain a set of wavelet bases, (iii) use the set of wavelet bases to represent and approximate the signal, and (iv) finally, a time–frequency localized analysis of the signal can be performed. The method proposed by Li et al. can detect road anomalies with high accuracy and can evaluate the size of road anomalies. All these aforementioned methods use sensor data from specially modified or designed platform. However, the platform that has been specially modified or designed is not popular. Using a smartphone as a sensor data acquisition platform can solve this problem very well.

Among these two methods, the image-based detection method is susceptible to external factors, such as weather, lighting, and road surface water. In contrast, the sensor-based detection method requires to embed sensors on the specially modified or designed platform (cars, motorcycles, bicycles, etc.). All those problems will negatively affect the efficiency of large-scale detection of bump features. In order to solve these problems, this article proposes a bump feature detection method based on Bi-LSTM. This method uses smartphone sensors to collect road surface information. Smartphone sensors are more readily available and in greater numbers than sensors mounted on specially modified or designed platforms, enabling more efficient detection of large-scale bump features. At the same time, the performance of the currently used detection methods is poor, and the detection efficiency is not high enough. Bi-LSTM performs well, is suitable for processing time-series information while considering contextual information, and can detect bump features more efficiently and accurately than traditional methods. The contributions of this article are summarized as follows:

(1) The correction and coordinate transformation of attitude data from smartphone sensors.

(2) The processing of sensor data using neighborhood high-dimensional embedding.

(3) The framework of Bi-LSTM-based BFRS detection.

3 Methodology

3.1 Motivation and background

When the vehicle passes through BFRS during the movement, it will vibrate and cause abnormal acceleration changes (ACA) that are vertical to the road surface, and the vibration will transmit to the smartphone placed inside the vehicle. Hence, the BFRS can be detected from the spatial-temporal recording of sensors assembled in the smartphone. With this assumption, the BFRS detection method based on Bi-LSTM is proposed. Due to variations in driving habits, the vehicle may actively avoid certain types of BFRS during the driving process. However, this article focuses only on the actual BFRS experienced by the vehicle. The detection method employed in this study primarily utilizes Z-axis acceleration to detect BFRS, while X-axis and Y-axis accelerations are used as supplementary measures. Therefore, the steering behavior of the driver has minimal impact on the test results. To account for differences in driving habits and other factors, the study selected urban roads as the experimental area, and the experimenters endeavored to maintain normal driving habits. As a result, the experimental findings are consistent with typical driving conditions. Nonetheless, due to limitations in the experimental conditions, this article mainly proposes an algorithmic framework, and the effects of various factors, such as driving habits and vehicle performance, on BFRS detection will continue to be explored in future research. Furthermore, the speed of the vehicle in this study was limited to <80 km/h.

In this article, the BFRS are defined as a virtual 3D object on the road surface, with a diameter larger than 20 cm in the horizontal direction and a depression or protrusion larger than 5 cm in the vertical direction. As shown in Figure 1, it is the object passing by vehicles, such as speed bumps and potholes, where the horizontal width of the object is d and the vertical height is h. As denoted in equation (1), if $d \geq 20$ cm and $h \geq 5$ cm, it can be judged as the BFRS in the research. Currently, the most common tire size is 225 mm. Using this as a reference and drawing on previous experience, this article proposes a definition of BFRS. However, due to limited experimental conditions, this article focuses mainly on proposing an algorithm for detecting BFRS rather than specifically investigating the relationship between tire size and BFRS detection. Besides, due to different driving behaviors, the driver will actively avoid certain BFRS while driving. In this condition, the BFRS might not be recorded by the sensors in the smartphone. Hence, only the actual BFRS passed by vehicles is studied in this arcticle.

$$\text{BFRS} = \{(d, h) | d \geq 20\,\text{cm}, h \geq 5\ \text{cm}\}. \tag{1}$$

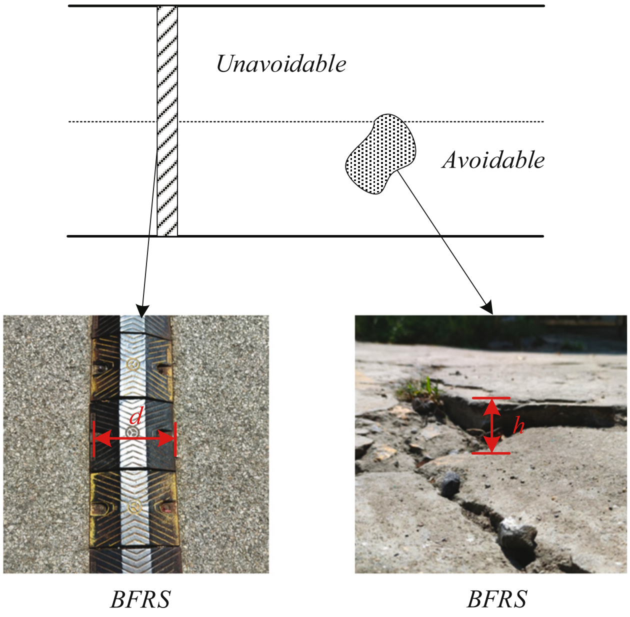


Figure 1: Bump features of the road surface.

The pipeline of the proposed method is illustrated in Figure 2. The sensor data are obtained through the GPS sensor and the IMU sensor assembled in the smartphone, in the form of GPS trajectory, 3D acceleration, and 3D orientation. Then, the input dataset is processed in three steps: (1) acceleration transformation based on the 3D orientation; (2) multi-dimensional neighborhood embedding using the sliding window; and (3) BFRS detection after the training of Bi-LSTM and obtaining the ACA. Finally, project the detected BFRS onto the map using the coherence information between the acceleration and GPS trajectories.

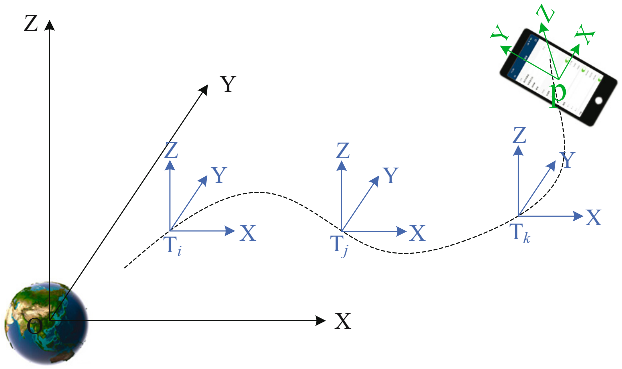


Figure 3: Schematic diagram of coordinate systems.

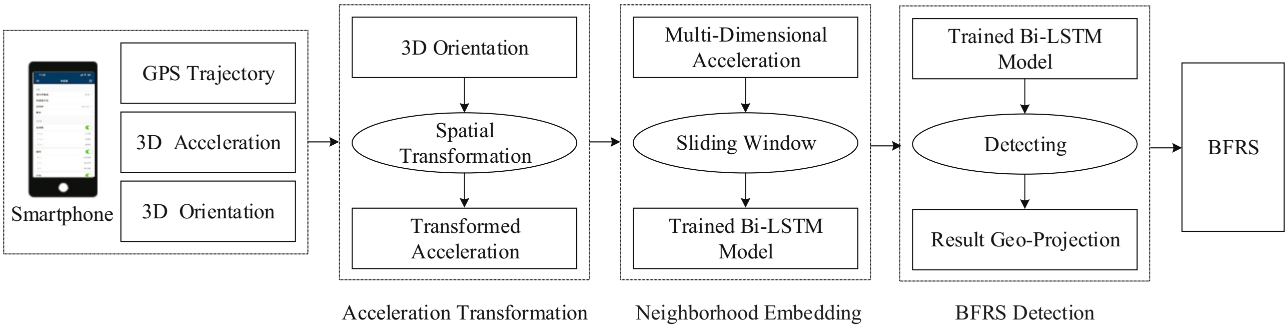


Figure 2: Flow chart of the method.

3.2 Acceleration decomposition based on the orientation transformation matrix

There are three coordinate systems involved in the acceleration decomposition process: the object coordinate system, the inertial coordinate system, and the world coordinate system.

(1) The object coordinate system is set by the sensors assembled within the smartphone in terms of the smartphone's pose, as shown by the green coordinate system in Figure 3.

(2) The world coordinate system is the absolute coordinate system of the geographic world [23]. The world coordinate system used in this article is WGS84. When a point is first defined in the scene, we say its coordinates are specified in world space, i.e., the coordinates of this point are described with respect to a global or world Cartesian coordinate system. The coordinate system has an origin, called the world origin, and the coordinates of any point defined in that space are described with respect to that origin (the point whose coordinates are [0,0,0]).

(3) The inertial coordinate system coincides with the original object coordinate system whose coordinate axes are parallel to the coordinate axes of the world coordinate system (Figure 3) [24].

Therefore, it is necessary to convert the coordinate system of the acceleration sensor into an inertial coordinate system through a rotation matrix and into a world coordinate system through a three-dimensional translation transformation matrix.

Since the mobile phone is not in a flat state when the data are collected, there is a certain angle between the mobile phone and the horizontal plane, which directly causes the acceleration in the numerical direction to be decomposed into the three axes of x, y, and z. The original z-axis acceleration data and vertical acceleration have a difference. In addition, if the absolute value of acceleration is used directly, it will be impossible to distinguish between the horizontal acceleration and the vertical acceleration change when the vehicle is going straight and turning.

Therefore, it is necessary to transform the coordinate system of the acceleration to make it consistent with the inertial coordinate system and, therefore, to analyze the data of the acceleration sensor to detect BFRS. The steps are as follows:

1. Transform the object coordinate system to the inertial coordinate system: To make the acceleration direction of the sensor the same as the acceleration direction of gravity, this article converts the acceleration of the

sensor in the object coordinate system to the acceleration in the inertial coordinate system. The process uses the pose angle and quaternion of the smartphone pose sensor to derive a conversion matrix $\boldsymbol{R}$, which is then used to convert the object coordinate system to an inertial coordinate system [25]. The quaternion q is a simple hypercomplex number that consists of a real part and three imaginary parts, i, j, and k. The quaternion q can generally be expressed as $a + b \times i + c \times j + d \times k$, where a, b, c, and d are scalars [26–28], as denoted in equation (2). The conversion of the object coordinate system to the inertial coordinate system is denoted in equation (3).

$$\boldsymbol{R} = \begin{pmatrix} 1 - 2b^2 - 2z^2 & 2ab + 2dc & 2ac - 2db \\ 2ab - 2dc & 1 - 2a^2 - 2c^2 & 2bc + 2da \\ 2ac + 2db & 2bc - 2da & 1 - 2a^2 - 2b^2 \end{pmatrix}, \quad (2)$$

$$(\mathrm{Acc}_x, \mathrm{Acc}_y, \mathrm{Acc}_z)_{\mathrm{inertial}} = \boldsymbol{R} \times (\mathrm{Acc}_x, \mathrm{Acc}_y, \mathrm{Acc}_z)_{\mathrm{object}}. \quad (3)$$

2. Aligning the inertial and world coordinate systems: Since the latitude and longitude collected by GPS are based on the world coordinate system, it is necessary to align the sensor data based on an inertial coordinate system to the world coordinate system to obtain the latitude and longitude of each data point. Due to the GPS sampling rate, the data must first be linearly interpolated so that each data point has a corresponding latitude and longitude, and then the inertial coordinate system data must be aligned to the world coordinate system based on the time stamp. The conversion of the inertial coordinate system to the world coordinate system is denoted in equation (4).

$$(\mathrm{Acc}_{\mathrm{world}}, \mathrm{GPS}) = \mathrm{Align}(\mathrm{Acc}_{\mathrm{inertial}}, \mathrm{GPS}). \quad (4)$$

3.3 Multi-dimensional acceleration neighborhood embedding based on the sliding window

When the vehicle carried with a smartphone is passing by the BFRS, ACA can be recorded. However, it is not easy to detect the BFRS directly from the acceleration data since there can be different sizes of the BFRS and the intensity of the ACA can vary from each other, even if the vehicle passing by has the same BFRS at different speeds. To deal with such kinds of situations, multi-dimensional accelerations are applied, i.e., the acceleration in the vertical direction of the inertial coordinate system and the other two accelerations in the horizontal direction of the inertial coordinate system. The vertical ACA is to represent the depth of the BFRS, and the horizontal ACA is to represent the length of the BFRS, as depicted in Figure 1. Therefore, three dimensions of information were used for detection.

Hence, both vertical and horizontal accelerations are taken into account, and the multi-dimensional neighborhood embedding process is performed on the acceleration data. The specific way is implemented by a sliding window. Suppose the sliding window is constructed by two variables: (1) the indicator representing the position of the window, and (2) the length representing the number of acceleration samplings [29,30]. In this article, the center position of the sliding window is set to l, and the length of the sliding window is represented by k data points before and after the center point. Therefore, these $2k + 1$ samplings are taken into the sliding window, and the sliding window starts from $n - k$ to $n + k$. The start and end indexes of the sliding window go along with the time direction until the end of the acceleration samplings, as depicted in Figure 4.

In Figure 4, the three dashes represent the acceleration on the x-, y-, and z-axis, as shown in the data below the dash. The fourth line of data is the feature label. The red box represents the sliding window, which slides from left to right. The final generated training data are a matrix containing acceleration and feature labels.

The embedding process is shown in equations (5) and (6).

$$\begin{aligned}&\text{Neighbor embedding}(\mathrm{Acc}_x, \mathrm{Acc}_y, \mathrm{Acc}_z, \mathrm{Window}_{\mathrm{sliding}}) \\ &\quad = \mathrm{NE},\end{aligned} \quad (5)$$

$$\mathrm{Labeling}(\mathrm{NE}, \mathrm{GPS}_{\mathrm{A}}) = \{\mathrm{NE}_{\mathrm{BFRS}}, \mathrm{lbl}_{\mathrm{BFRS}}\}. \quad (6)$$

In equation (5), the input of the neighborhood embedding algorithm consists of the acceleration components Acc_x, Acc_y, and Acc_z, as well as the length of the sliding window $\mathrm{Window}_{\mathrm{sliding}}$. The output of this algorithm is denoted as NE. In equation (6), NE and position $\mathrm{GPS}_{\mathrm{A}}$ are fed into the geographic labeling algorithm. The outputs of this algorithm are the geographic labeling $\mathrm{lbl}_{\mathrm{BFRS}}$ and the acceleration $\mathrm{NE}_{\mathrm{BFRS}}$ of BFRS in the sensor data.

3.4 BFRS detection based on the Bi-LSTM

Bi-LSTM is a kind of RNN with internal self-connection, which consists of an input layer, hidden layer, and output layer. The state of the hidden layer at the previous moment can be transmitted to the hidden layer at the next moment through this feedback structure, along with the external input variable at the current moment [31,32]. It can judge whether there is BFRS, by focusing on not only the current time series but also the previous and subsequent time series. Bi-LSTM improves the standard LSTM network by

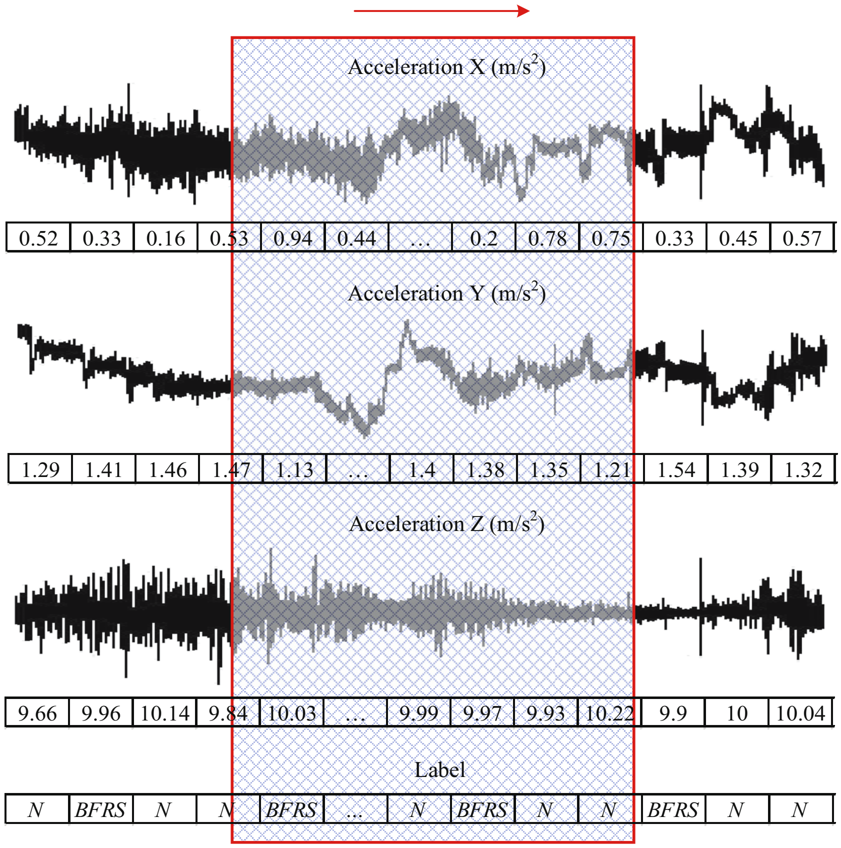


Figure 4: Schematic diagram of the sliding window.

adding another layer of LSTM. The two layers of LSTM are opposite in direction [33], as shown in Figure 5. RNN is well-suited for processing time series data. Bi-LSTM, a variant of RNN, offers significant advantages for analyzing time series data. Detecting a BFRS requires not only analyzing current time series data but also incorporating the time series data prior to and following the detection point to determine whether the point represents a BFRS. This is because, in accordance with the definition of BFRS, a current time series can be defined as a BFRS only if it has significantly diverged from both the preceding and succeeding time series. Therefore, this article selects Bi-LSTM as the analysis technique, as it is capable of simultaneously analyzing the current, preceding, and succeeding time series. This characteristic of Bi-LSTM aligns with the requirements for BFRS detection.

In Figure 5, the bottom layer is the training data of the Bi-LSTM, the middle is the forward and backward LSTM, respectively, and the top layer *y* indicates the output feature labels. The blue box shows the structure of a single

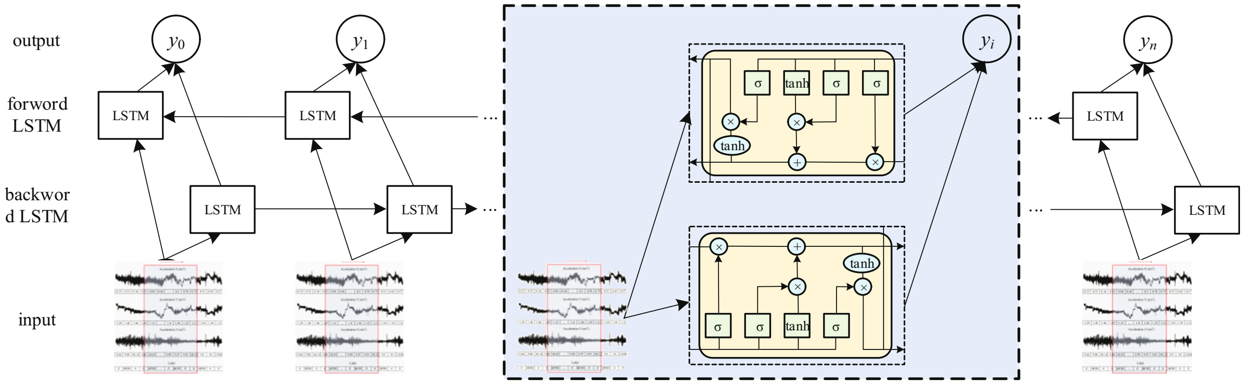


Figure 5: Structure diagram of Bi-LSTM.

LSTM unit. The small light green squares in the figure represent four ordinary neural networks, and the boxes labeled σ and tanh are their activation functions.

A single LSTM cell contains three parts: a forget gate, an input gate, and an output gate.

Forget gate: the forget gate selectively forgets the information in the cellular state from the previous step. The forget gate is implemented through the sigmoid layer. Using h_{t-1} from the previous step and x_t from this step as input, a value between 0 and 1 is output for each number in C_{t-1}, denoted as f_t, which indicates how much information is retained (1 means fully retained and 0 means fully discarded). The process is denoted in equation (7).

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f). \tag{7}$$

Input gate: the input gate is responsible for the selective recording of new information into the cell state. The sigmoid layer (input gate layer) determines the value to be updated, and this probability is denoted as i_t. The tanh layer creates a vector of candidate values $\widetilde{C}_t$ that will be added to the cell states. The old cell state is then updated. The process is denoted in equations (8)–(10).

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i), \tag{8}$$

$$\widetilde{C}_t = \tanh(W_C[h_{t-1}, x_t] + b_C), \tag{9}$$

$$C_t = f_t \times C_{t-1} + i_t \times \widetilde{C}_t. \tag{10}$$

Output gate: the forecast made by the output gate decision. Firstly, the sigmoid layer (output gate) is utilized to determine the part of output native cell state C_t. The cell state is then passed through the tanh layer (so that the value is between −1 and 1), followed by multiplication with the output of the sigmoid layer to obtain the final output h_t. The process is denoted in equations (11) and (12).

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o), \tag{11}$$

$$h_t = o_t \times \tanh(C_t). \tag{12}$$

The training part of the neural network divides the data into two parts: training data and validation data. After the training data are input into the neural network for training, the training results are output, as denoted in equation (13).

$$\{\text{err,itr,model}\} = \text{Bi-LSTM}(\{\text{NE}_{\text{BFRS}},\text{lbl}_{\text{BFRS}}\}_{\text{train}},\text{GPS}_{\text{A}}). \tag{13}$$

3.5 Algorithm of the BFRS detection method

The algorithm of the BFRS method is composed of three stages:

1. Spatial transformation for acceleration using the orientation. The initial acceleration was transformed into the inertial coordinate system. Then, GPS and acceleration were transformed into the world coordinate system.
2. Neighborhood embedding of the multi-dimensional acceleration. The multi-dimensional acceleration was neighborhood embedded with the sliding window. Then, these data were labeled using aligned GPS trajectories.

BFRS training and detection based on the Bi-LSTM. The training dataset was inputted into the Bi-LSTM to train. BFRS was detected by the trained Bi-LSTM and projected back to the geographic space.

The pseudo-code of the proposed method is as follows:

Algorithm 1 The algorithm of BFRS detection based on Bi-LSTM

```
BFRS detection based on Bi-LSTM BFRSDect(GPS, Ori, Acc)
Input: recorded trajectories GPS, orientation of the smart phone Ori, acceleration of the smart phone Acc
Output: detected BFRS
//stage 1: spatial transformation for acceleration using the orientation
T = SpatialMatrix(Orix, Oriy, Oriz);//construct the spatial transformation matrix based on the 3D Orientation
AccT = SpatialTransform(T, Acc);//transform the acceleration direction into the inertial coordinate system
{AccA, GPSA} = Alignment(AccT, GPS);//align GPS and acceleration into the world coordinate system
//stage 2: neighborhood embedding of the multi-dimensional acceleration
NE = NeighborEmbedding(Accx A, Accy A, Accz A, Windowsliding);//neighborhood embedding for the multi-dimensional acceleration
{NEBFRS, lblBFRS} = Labeling(NE, GPSA);//labeling the embedding using aligned GPS trajectories
//stage 3: BFRS training and detecting based on the Bi-LSTM
{{NEBFRS, lblBFRS}train, {NEBFRS, lblBFRS}validate} = Divide({NEBFRS, lblBFRS});//divide training and validating dataset
IF err <= THRESHOLD OR itr >= ITERATION
    {err, itr, model} = Bi-LSTM({NEBFRS, lblBFRS}train, GPSA);//train the Bi-LSTM model
END
BF = Dectect(model, {NEBFRS, lblBFRS}train, GPSA);//detect Bump Features using the trained Bi-LSTM model
BFRS = GeoProject(BF, GPSA);//project the BFRS back to the geographic space
RETURN BFRS
```

4 Experiments and discussion

This section focuses on the performance of the BFRS detection method. The smartphone-based sensor data are collected in the experimental area. The acceleration data are transformed into the inertial coordinate system, and then training and validating datasets are constructed using the sliding window. The Bi-LSTM model to detect BFRS is trained and validated, and comparisons are conducted with different methods. In addition, the capability of the proposed method is further validated in a different area, along with the related discussion.

4.1 Collected datasets and experimental area

In this article, the smartphone is arbitrarily placed on the vehicle to collect sensor data, including 3D acceleration data, 3D orientation data, and GPS data. The sampling frequency of accelerator sensor and orientation sensor is 100 Hz, while that of the GPS sensor is 1 Hz. The first experimental area is the campus with an area of 0.4958 km^2, as shown in Figure 6. In a training section, the road surface is flat, with almost no uphill or downhill. Thus, it can be an ideal location for training neural networks.

In Figure 6, the recorded sensor data are represented by the GPS trajectories (shown by the black line), with GPS sampling, 3D orientation, and 3D acceleration information stored. The related BFRS on the ground is represented by the blue dot and distributed along the road map. In addition, three BFRS (BFRS$_1$, BFRS$_2$, and BFRS$_3$) in different locations are selected and shown in the right part of the figure, and it is the main reason that caused ACA, which can be further used to detect BFRS. The sensor data in the green area is the validation dataset to evaluate the performance of the proposed method and make comparisons with different methods. The total length of the recorded trajectories is 3,522 m, with 33 BFRS and 64,397 accelerometer sampling points, as illustrated in Table 1.

4.2 Acceleration transformation and multi-dimensional neighborhood embedding

Since the smartphone is arbitrarily placed, the direction of recorded 3D acceleration data may not be consistent with

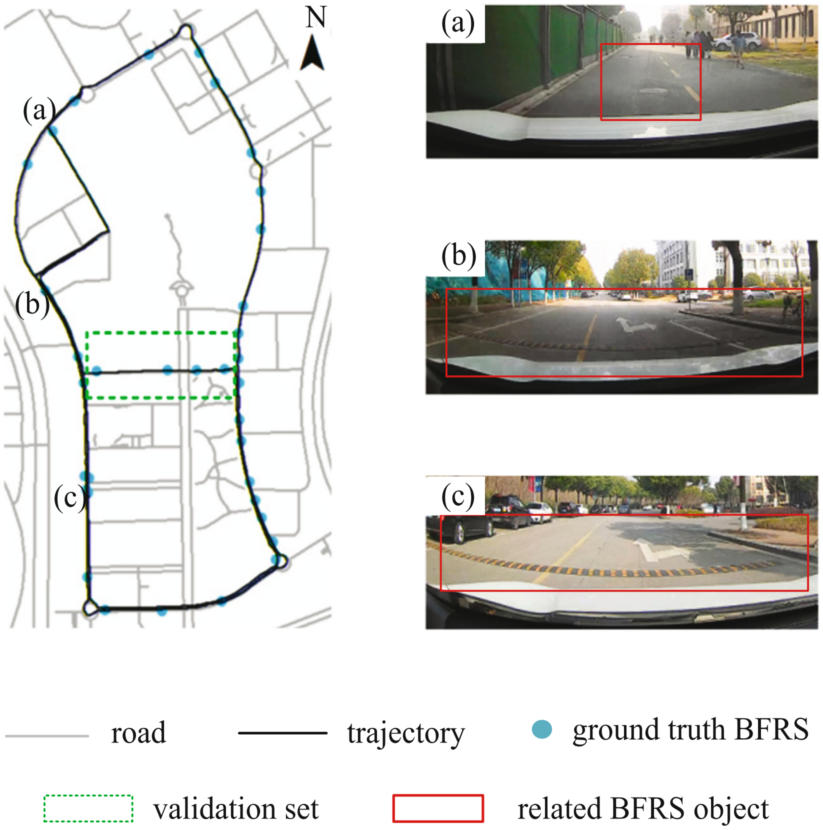
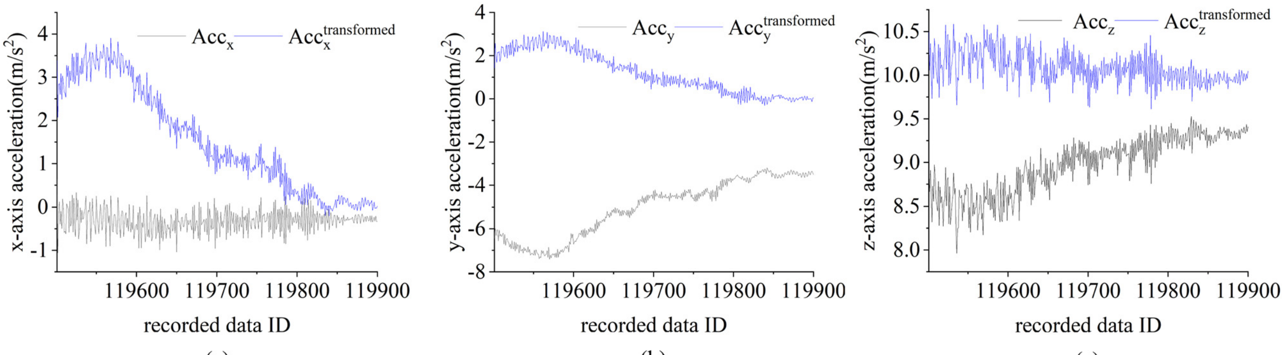


Figure 6: Experimental area of the campus: (a) BFRS$_1$, (b) BFRS$_2$, and (c) BFRS$_3$.

Table 1: Experimental area

Length	**$\text{SpRate}_{\text{acc,ori}}$**	**$\text{SpNum}_{\text{acc,ori}}$**	**$\text{SpRate}_{\text{GPS}}$**	**$\text{SpNum}_{\text{GPS}}$**	**BFRS_{num}**
3,522 m	100 Hz	64,397	1 Hz	644	33



(a) (b) (c)

Figure 7: Comparison before and after acceleration transformation: (a) *x*-axis acceleration, (b) *y*-axis acceleration, and (c) *z*-axis acceleration.

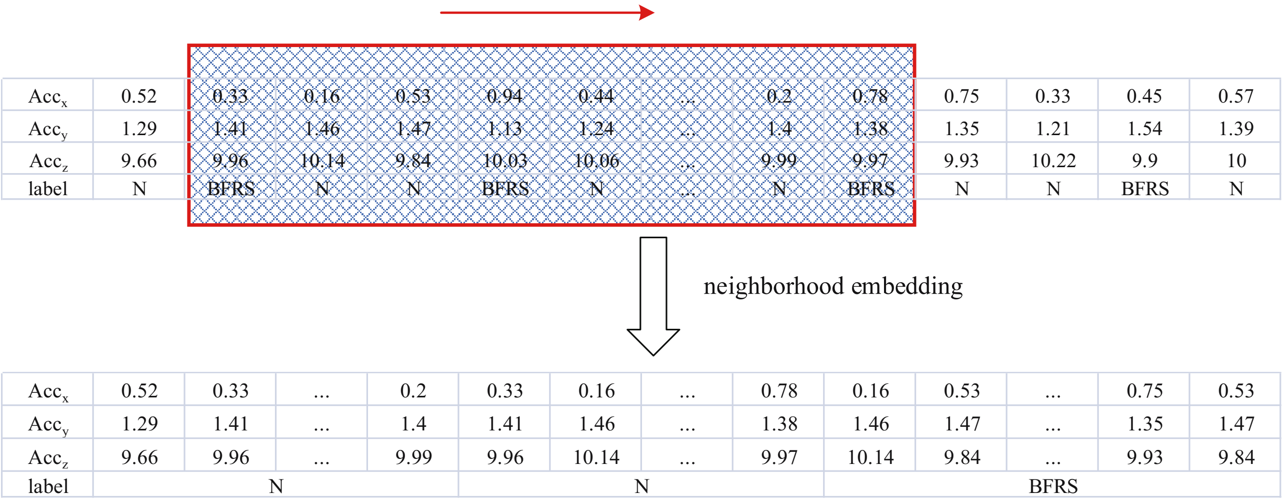
the inertial coordinate system of the vehicle, which can lead to a false judgment of the BFRS, as shown in Figure 7. In the figure, there are three comparisons of acceleration in the *x*-, *y*-, and *z*-axis made before and after the spatial transformation, with the original 3D acceleration and the transformed acceleration changes drawn in gray and blue, respectively. In Figure 7(a), the original acceleration in the *x*-axis keeps steady, while the ACA cannot be directly detected until the transformation has been applied. In Figure 7(b), the ACA remains consistent with the actual cases before and after the transformation. However, in Figure 7(c), the ACA in the original data becomes weak after the transformation. In addition, the average acceleration of the *z*-axis, i.e., the vertical direction of the inertial coordinate system, is 10.08 m/s^2, which is more consistent with the actual case in the world.

The statistics of the selected sensor data in Figure 7 are illustrated in Table 2. There are 400 consecutive samplings in the dataset, and the average acceleration value in the *x*-, *y*-, *z*-axis is −0.31, −5.05, and 8.97 m/s^2, respectively. However, the values in the *x*-, *y*-, and *z*-axis become 1.55, 1.19, and 10.08 m/s^2, respectively. The ACA can be detected in the horizontal direction of the inertial coordinate system based on the changes of acceleration in the *x*- and *y*-axis, while it cannot be detected just using the *z*-axis, since the acceleration changes weakly in the vertical direction. In this condition, there can be a turn at the road intersection, with only horizontal ACA detected and no dramatic changes in the vertical direction. Hence, the spatial transformation of the acceleration can be meaningful, and it can detect the actual BFRS using ACA in different acceleration dimensions.

Table 2: Comparison of average values before and after acceleration transformation

Acceleration	**Before transformation (m/s^2)**	**After transformation (m/s^2)**
x-axis	−0.31	1.55
y-axis	−5.05	1.19
z-axis	8.97	10.08

Following data preprocessing, the neural network is trained using 101-dimensional three-axis acceleration data as input, with the variable *k* set to 50 for multi-dimensional neighborhood embedding. The first column of the data represents data-id, while the second to the 101st columns contain acceleration data processed through multi-dimensional neighborhood embedding. The 102nd column represents the feature label. Ground object annotation is obtained through manual marking of the longitude and latitude of BFRS, which are collected using handheld GPS and marked according to the frequency spectrum image of the acceleration data. This process ensures the accuracy of ground object marking. Initially, the longitude and latitude of high-precision non-stationary ground objects are collected using handheld GPS on the experimental road section and then mapped to the acceleration data. Next, 150 data points are generated before and after BFRS to create a spectrum image, from which data points with significant acceleration changes are extracted. Finally, the extracted data are aggregated, and high-dimensional embedding



Acc_x	0.52	0.33	0.16	0.53	0.94	0.44	...	0.2	0.78	0.75	0.33	0.45	0.57
Acc_y	1.29	1.41	1.46	1.47	1.13	1.24	...	1.4	1.38	1.35	1.21	1.54	1.39
Acc_z	9.66	9.96	10.14	9.84	10.03	10.06	...	9.99	9.97	9.93	10.22	9.9	10
label	N	BFRS	N	N	BFRS	N	...	N	BFRS	N	N	BFRS	N

Acc_x	0.52	0.33	...	0.2	0.33	0.16	...	0.78	0.16	0.53	...	0.75	0.53
Acc_y	1.29	1.41	...	1.4	1.41	1.46	...	1.38	1.46	1.47	...	1.35	1.47
Acc_z	9.66	9.96	...	9.99	9.96	10.14	...	9.97	10.14	9.84	...	9.93	9.84
label	N				N				BFRS				

Figure 8: Neighborhood embedding.

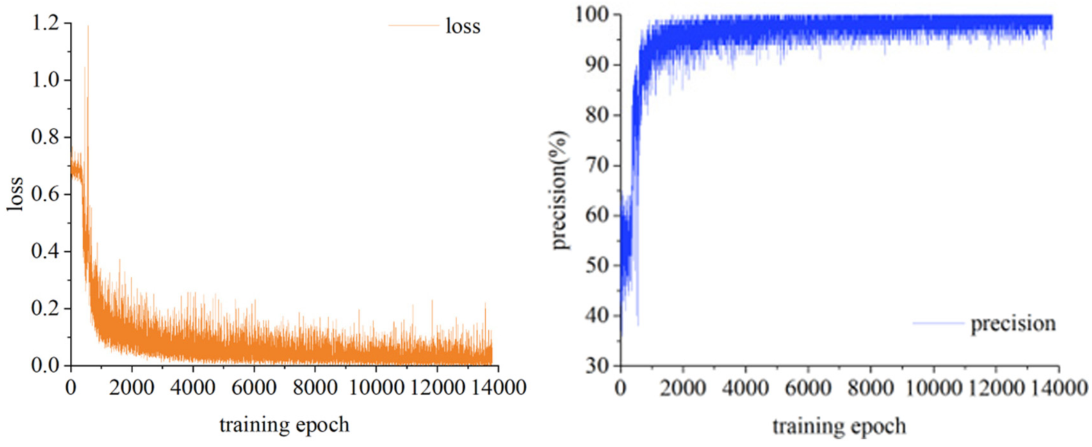
is performed to generate a training set. Figure 8 displays the embedding process.

4.3 BFRS detection model training and comparison with different methods

With the prepared training and validating datasets, the BFRS detection model based on Bi-LSTM is trained and evaluated. The hardware for training is the NVIDIA CPU RTX 3060. The training parameters of the model in this article refer to the trainNetwork function in matlab: {150 epoch, training 92 iterations per epoch, learning rate 0.001}. The layer is configured as follows: it has three features, 100 hidden units, and a many-to-one output mode with one class. The number of hidden units was determined through multiple experiments, while the remaining parameters were determined based on the dimensions of the input and output data. Finally, the BFRS detection model is constructed. The training process is shown in Figure 9, and the related confusion matrix is shown in Figure 10.

As seen in Figure 9, the loss of the model keeps decreasing and the precision keeps increasing. Hence, the model becomes converged after 150 epochs. In the confusion matrix of Figure 10, there are two kinds of output, i.e., BFRS and normal surface, and 2,019 BFRS labels are properly predicted among the 2,053 training datasets, with a precision of 97.63%, a recall of 98.34%, and an *F*-score of 97.98%.

To make an evaluation of the proposed method, the dataset in the green area of Figure 6 is selected, and the



(a) (b)

Figure 9: Training process for the BFRS detection model: (a) loss and (b) precision.

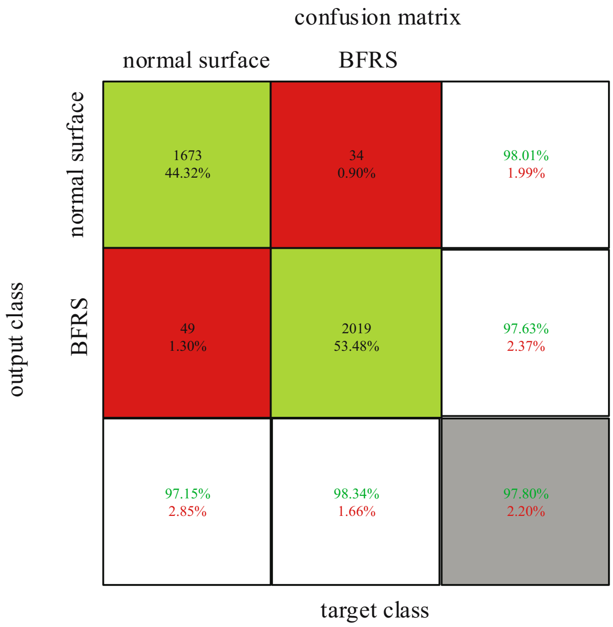


Figure 10: The confusion matrix of BFRS detection results.

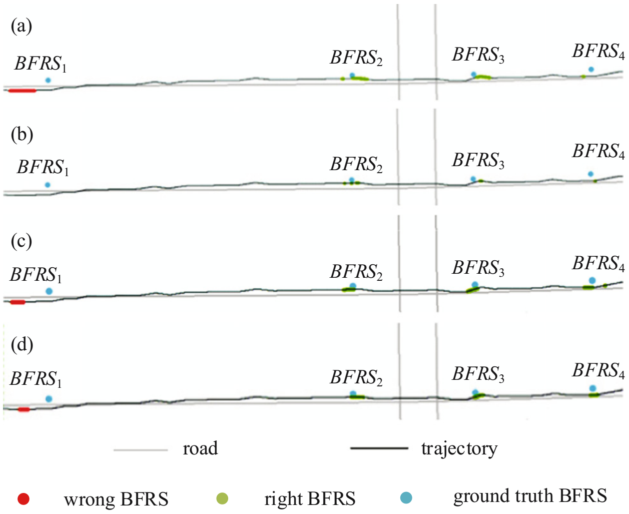


Figure 12: Comparison of BFRS detection results by different methods: (a) the proposed method, (b) CWT, (c) BP, and (d) ISCHANGE.

detailed information is shown in Figure 11. A vehicle just drove through the road (the gray line), with a trajectory (black line) passing by BFRS and information stored in the trajectory. Four BFRS objects ($BFRS_1$, $BFRS_2$, $BFRS_3$, $BFRS_4$) are spatially distributed along the road, and both of them are typical objects that can cause obvious ACA.

With the trained BFRS detection model, BFRS is detected in the selected area, as depicted in Figure 12. The detected GPS samplings are assigned related labels, i.e., "BFRS" or "normal surface." If the GPS sampling point is within the buffer area (the buffer distance of 10.00 m) of the ground truth BFRS (blue dot), it can be taken as "right BFRS," as shown by the green dot. However, the red dot is considered to be the "wrong BFRS," which is out of the reach of the buffer area. Figure 12(a) shows the BFRS detection result based on the proposed method. Three BFRS are successfully detected with green dots inside the buffer area of ground truth BFRS, and one BFRS is undetected and labeled in red. To make a comparison, the methods of CWT [22], BP [15], and ISCHANGE [34] are applied in this article, with related results depicted in Figure 12(b)–(d). In Figure 12(b), few samplings are properly detected in each BFRS, and even no result is detected for BFRS$_1$. In Figure 12(c) and (d), there are some samplings properly detected in BFRS$_2$, BFRS$_3$, and BFRS$_4$, while BFRS$_1$ is not successfully detected, similar to that in Figure 12(a). To obtain a specific evaluation, a statistical analysis of BFRS detection results by different methods is conducted, as depicted in Table 3.

In Table 3, it is the statistical result of the recall, precision, and *F*-score for each method. The precision is

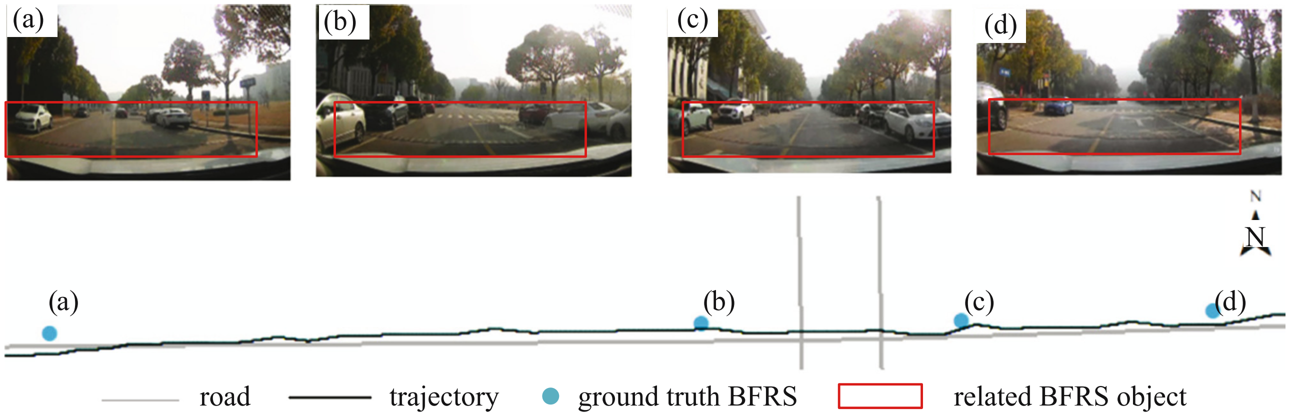


Figure 11: The ground truth of the comparison area. (a) BFRS$_1$, (b) BFRS$_2$, (c) BFRS$_3$, and (d) BFRS$_4$.

Table 3: Statistics of BFRS detection results by different methods

Different methods	Recall (%)	Precision (%)	*F*-score (%)
The proposed method	93.29	92.41	92.85
CWT	7.03	100.00	13.14
BP	22.68	73.96	34.71
ISCHANGE	19.49	85.92	31.78

computed by dividing the number of correct predictions by the number of all predictions that are true. In contrast, the recall is computed by dividing the number of predictions correctly predicted to be true by the number of all predictions that are actually true. The *F*-score is computed by multiplying the recall rate by two times the precision rate and dividing it by the recall rate plus the precision

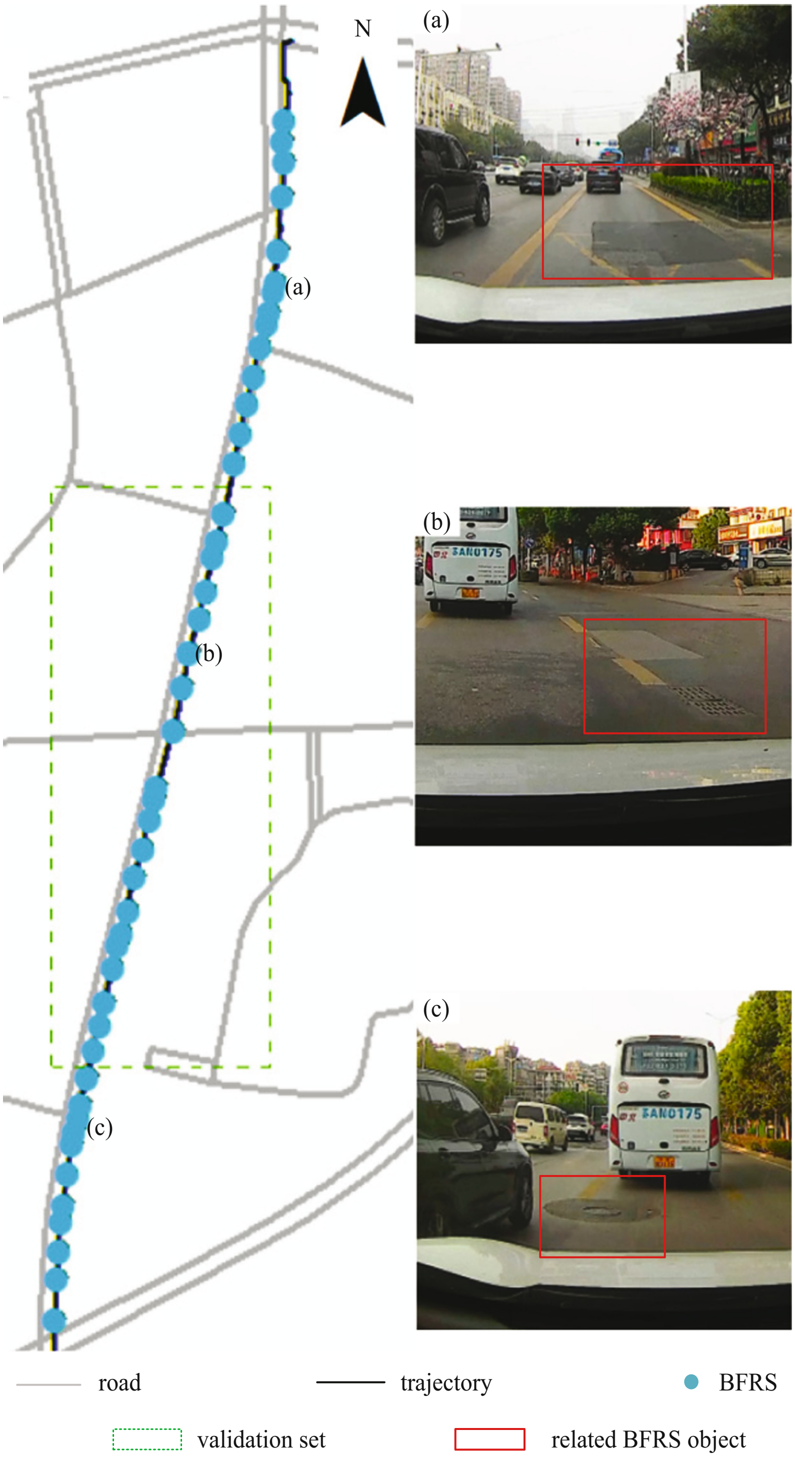


Figure 13: The ground truth of the complex area. (a) $BFRS_1$, (b) $BFRS_2$, and (c) $BFRS_3$.

Table 4: Statistics of the complex area

Length	SpRate$_{acc,ori}$	SpNum$_{acc,ori}$	SpRate$_{GPS}$	SpNum$_{GPS}$	BFRS$_{num}$
1,225 m	100 Hz	22,587	1 Hz	226	49

rate. The results of different methods are shown in Table 3. Based on the computation standard, the precision of BFRS for the proposed method is 92.41%, and the precision of BFRS for CWT, BP, and ISCHAGE is 100.00, 73.96, and 85.92%, respectively. Although the precision of CWT and ISCHAGE is higher than the proposed method, the recall is a little low. Hence, the *F*-score for each of the methods is 92.85, 13.14, 34.71, and 31.78%, respectively, and the proposed method holds the highest *F*-score among other methods. In combination with Figure 12, the proposed method can detect BFRS and outperform other compared methods.

4.4 BFRS detection and comparison in the complex area

In order to evaluate the capability of the proposed BFRS detection method, a complex area with more BFRS and different ACA intensities is selected, as depicted in Figure 13. It is a road in a residential area, and the sub-area in the green rectangle is used for zoomed-in views in comparison. The BFRS in this area is not as typical as that in Figure 6, since the road surface is not so flat and the height or depth of each BFRS is not so apparent (illustrated by BFRS_1, BFRS_2, and BFRS_3 in Figure 13). Although BFRS_1 and BFRS_2 are located in relatively flat areas, it is not easy to catch the ACA in recorded sensor data compared with BFRS_3. The detailed information is shown in Table 4.

As depicted in Table 4, the total length of the recorded trajectory is 1,225 m, and the number of stored samplings of the acceleration and orientation sensor is 22,578 with a sampling rate of 100 Hz, while that for the GPS samplings is 226 with a sampling rate of 1 Hz. In addition, there are 49 BFRS in the area. The stretch of road is mostly flat but includes a downhill section at the beginning due to it descending from an elevated bridge. Therefore, some parts of the road surface slope downward.

Using the BFRS detection model trained in Section 4.3, experiments are conducted in the complex area, and the results are shown in Figure 14. Since the road condition in this area is quite complicated, it is difficult to manually label each BFRS sample coherent with the actual case. Hence, the ground truth BFRS is generated using the video

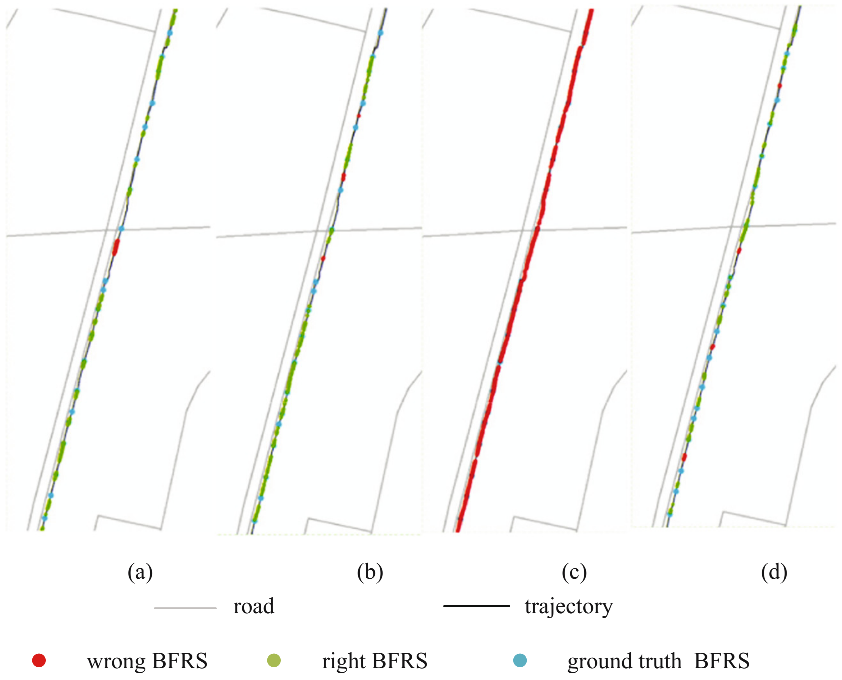


Figure 14: Comparison of BFRS detection results in the complex area: (a) the proposed method, (b) CWT, (c) BP, and (d) ISCHANGE.

Table 5: Statistics of BFRS detection results in the complex area

Different methods	Recall (%)	Precision (%)	*F*-score (%)
The proposed method	86.27	89.79	87.99
CWT	78.43	81.63	80.00
BP	—	—	—
ISCHANGE	83.67	80.39	82.00

recording during the driving process, and related BFRS samplings are constructed based on the buffer of the ground truth BFRS, i.e., the GPS samplings within 10 m of the ground truth BFRS are taken as the right BFRS. Figure 14(a) shows the BFRS detection result based on the proposed method. There are 44 BFRS detected out of 49 ground truth BFRS (blue dots), with five BFRS undetected. The properly detected GPS samplings are shown by the green dot, and the wrong BFRS are in red. The results in Figure 14(b) and (d) seem good compared with those in Figure 14(c), since the BFRS detection result in Figure 14(c) is extremely poor, as labeled by the red dot. In Figure 14(b), the CWT method detected 51 BFRS, of which 40 BFRS were correctly detected and nine BFRS were wrong. The BFRS detection result based on ISCHANGE is shown in Figure 14(d). The maximum number by the ISCHANGE function is set to 800. Finally, 51 BFRS are detected, and 41 are correctly detected. To obtain a detailed result of the comparison, the recall, precision, and *F*-score are computed, as shown in Table 5.

In Table 5, the performance of the proposed BFRS detection method is compared with CWT, BP, and ISCHAGE. Since there are too many wrongly detected BFRS in the result of BP, as shown in Figure 14(c), the recall, precision, and *F*-score are not computed. In addition, the BP method seems only effective in the training dataset, but it cannot effectively detect BFRS when the area is changed. The precision of the proposed method is 89.79%, which is higher than CWT, BP, and ISCHANE. The precision of the CWT is higher than that of the ISCHANGE, while the recall of the CWT is a little high; hence, the *F*-score of the CWT is 80.00%, which is lower than that of the ISCHANGE. The proposed method holds the highest *F*-score among the compared methods.

Through these experiments, it can be observed that most of the BFRS can be properly detected based on the proposed method, and the proposed method can perform well in different areas compared with other methods.

5 Conclusions and future work

The detection of BFRS has an important impact on the maintenance of the road network. Traditional BFRS detection methods require professional equipment and skills, which lead to limited efficiency. To solve the problem, the BFRS detecting method based on Bi-LSTM is proposed, and the road information is collected through the built-in sensors of smartphones. The 3D acceleration, 3D orientation, and GPS sampling are recorded in the arbitrarily placed smartphone during the movement of vehicles. Then, the spatial transformation matrix is constructed using the 3D orientation information, and the 3D acceleration is transformed into the inertial coordinate system. To detect the true BFRS through the acceleration information, a multidimensional training dataset is generated using the sliding window. The dataset is divided into the training dataset and the validating dataset. After the preparation process, the Bi-LSTM is trained and validated to detect the bump features based on the ACA information, and the detected features are further aligned with GPS samplings. The BFRS is projected onto the world coordinate system and represented in the road map. Using the proposed method, BFRS in different areas is experimented, and the results are compared with different methods. The recall, precision, and *F*-score are analyzed and computed, and the *F*-score of the proposed method is higher than 87%, which holds the best performance among the compared results.

Due to the limitations of experimental conditions, the experimental area in this article was located within the city, and only one car and one driver were used for data collection. The speed of the car is limited to under 80 km/h. The main purpose of this article is to propose an algorithmic approach for detecting non-smooth road surfaces. In future research, crowd-sourced data will be used as a data collection method for this approach, which will eliminate the effects of driver behavior and vehicle performance on detection results through a large amount of data. Future research will focus on the refinement and extension of the proposed method with the crowd-sourced sensor data from different kinds of smartphones, where an end-to-end BFRS detection model is required. Besides, different kinds of BFRS also require to be categorized and detected separately. Hence, more training and validating datasets are demanded, and the capacity of the proposed method also needs to be improved.

Funding information: This research was funded by the National Natural Science Foundation of China (No. 42101466) and the Postgraduate Scientific Research Innovation Program of Jiangsu Province (Grant No. KYCX21_0761).

Author contributions: H.L. performed the theory analysis and methodology and contributed to drafting the manuscript. K.X. analyzed the data, design, and coding. D.J. performed the

literature reviews, provided the background knowledge, and improved the writing. Q.Z. collected the data and conducted the statistics. All authors have read and agreed to the published version of the manuscript.

Conflict of interest: The authors state no conflict of interest.

Data availability statement: Publicly available datasets were analyzed in this study. The data can be found at: https://doi.org/10.6084/m9.figshare.21437883.

References

[1] Oliskevych M, Taran I, Volkova T, Klymenko I. Simulation of cargo delivery by road carrier: Case study of the transportation company. Sci Bull Natl Min Campus. 2022;2:118–23.

[2] Gong X, Wu F, Xing R, Du J, Liu C. LCBRG: A lane-level road cluster mining algorithm with bidirectional region growing. Open Geosci. 2021;13(1):835–50.

[3] Tsimoshynska O, Koval M, Kryshtal H, Filipishyna L, Koval V. Investing in road construction infrastructure projects under public-private partnership in the form of concession. Sci Bull Natl Min Campus. 2021;2:184–92.

[4] Hassan N, Siddiqui I, Mazhar S, Hameed H. Road anomaly classification for low-cost road maintenance and route quality maps. In: 2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Kyoto, Japan; 11–15 March 2019.

[5] Chen Z, Zhang J, Tao D. Progressive LiDAR adaptation for road detection. IEEE/CAA J Automatica Sin. 2019;6(3):693–702.

[6] Giannopoulos A. Modelling ground penetrating radar by GprMax. Constr Build Mater. 2005;19(10):755–62.

[7] Majumder S, Deen MJ. Smartphone sensors for health monitoring and diagnosis. Sensors. 2019;19(9):2164.

[8] Shoaib M, Scholten H, Havinga PJ. Towards physical activity recognition using smartphone sensors. In: 2013 IEEE 10th International Conference on Ubiquitous Intelligence and Computing and 2013 IEEE 10th International Conference on Autonomic and Trusted Computing, Vietri sul Mare, Italy; 18–21 December 2013.

[9] Jia Y, Jin S, Savi P, Yan Q, Li W. Modeling and theoretical analysis of GNSS-R soil moisture retrieval based on the random forest and support vector machine learning approach. Remote Sens. 2020;12(22):3679.

[10] Jianhua L, Guoqiang F, Jingyan L, Danqi W, Zheng C, Nan W, et al. Mobile phone indoor scene features recognition localization method based on semantic constraint of building map location anchor. Open Geosci. 2022;14(1):1268–89.

[11] Wang S, Yang F, Cheng Y, Yang Y, Wang Y. Adaboost-based crack detection method for pavement. In IOP Conference Series: Earth and Environmental Science, Qingdao, China; 23–25 November 2018.

[12] Zhang Z, Ai X, Chan CK, Dahnoun N. An efficient algorithm for pothole detection using stereo vision. In: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy; 4–9 May 2014.

[13] Dong H, Song K, Wang Q, Yan Y, Jiang P. Deep metric learning-based for multi-target few-shot pavement distress classification. IEEE Trans Ind Inform. 2021;18(3):1801–10.

[14] Guo W. Intelligent detection device of pavement disease based on image recognition technology. In Journal of Physics: Conference Series, Guilin, China, 26–28 March 2021; 2021, April.

[15] Li L, Sun L, Ning G, Tan S. Automatic pavement crack recognition based on BP neural network. PROMET-Traffic Transp. 2014;26(1):11–22.

[16] Liu L, Chen J, Xu L. Realization and application research of BP neural network based on MATLAB. In 2008 International Seminar on Future BioMedical Information Engineering, Wuhan, China, 18 December 2008; 2008, December.

[17] Sun J, Wang G. Geographic information system technology combined with back propagation neural network in groundwater quality monitoring. ISPRS Int J Geo-Inf. 2020;9(12):736.

[18] Giacomin J, Woo YJ. A study of the human ability to detect road surface type on the basis of steering wheel vibration feedback. Proc Inst Mech Eng Part D: J Automobile Eng. , 2005;219(11):1259–70.

[19] Ren Y, Wen G, Li X. An SVM based algorithm for road disease detection using accelerometer. TELKOMNIKA Indones J Electr Eng. 2013;11(9):5169–75.

[20] Zang K, Shen J, Huang H, Wan M, Shi J. Assessing and mapping of road surface roughness based on GPS and accelerometer sensors on bicycle-mounted smartphones. Sensors. 2018;18(3):914.

[21] Tai YC, Chan CW, Hsu JYJ. Automatic road anomaly detection using smart mobile device. In Conference on technologies and applications of artificial intelligence. Hsinchu, Taiwan; 18–20 November 2010.

[22] Li X, Huo D, Goldberg DW, Chu T, Yin Z, Hammond T. Embracing crowdsensing: An enhanced mobile sensing solution for road anomaly detection. ISPRS Int J Geo-Inf. 2019;8(9):412.

[23] Kozhevnikov N, Danilova V. Human development in the world coordinate system on the basis of limit equilibria. Agathos. 2018;9(1):135–43.

[24] Russell CT. Geophysical coordinate transformations. Cosmic Electrodyn. 1971;2(2):184–96.

[25] Shoemake K. Animating rotation with quaternion curves. In Proceedings of the 12th annual conference on Computer graphics and interactive techniques. New York, United States; 1985, July.

[26] Eberly D. Quaternion algebra and calculus. Magic Software Inc. 2002;26:1–8.

[27] Palais B, Palais R, Rodi S. A disorienting look at Euler's theorem on the axis of a rotation. Am Math Mon. 2009;116(10):892–909.

[28] Murty MR, Weatherby C. A generalization of Euler's theorem for ζ (2 k). Am Math Mon. 2016;123(1):53–65.

[29] Senft M. Suffix tree for a sliding window: An overview. In: WDS, Prague Czech; 2005, June.

[30] Koç CK. Analysis of sliding window techniques for exponentiation. Comput Math Appl. 1995;30(10):17–24.

[31] Schmidt RM. Recurrent neural networks (RNNs): A gentle introduction and overview. arXiv preprint arXiv:1912.05911; 2019.

[32] Azad A, Wang X. Land use change ontology and traffic prediction through recurrent neural networks: A case study in Calgary, Canada. ISPRS Int J Geo-Inf. 2021;10(6):358.

[33] Ahmed S, Saif AS, Hanif MI, Shakil MMN, Jaman MM, Haque MMU, et al. Att-BiL-SL: Attention-based Bi-LSTM and sequential LSTM for describing video in the textual formation. Appl Sci. 2021;12(1):317.

[34] Killick R, Fearnhead P, Eckley IA. Optimal detection of changepoints with a linear computational cost. J Am Stat Assoc. 2012;107(500):1590–8.