



Real-time vehicle identification using two-step LSTM method for acceleration-based bridge weigh-in-motion system

Yanjie Zhu^{1,2} · Hidehiko Sekiya^{1,3} · Takayuki Okatani^{3,4} · Ikumasa Yoshida¹ · Shuichi Hirano⁵

Received: 9 November 2021 / Revised: 24 March 2022 / Accepted: 12 April 2022
© Springer-Verlag GmbH Germany, part of Springer Nature 2022

Abstract

Recently, accelerometers have been employed for bridge weigh-in-motion (BWIM) systems to provide more durable field measurements comparing with conventional strain-based sensors. As the basis of BWIM system, accurate vehicle identification provides fundamental support for vehicle loads monitoring and overweight traffic detection. However, research efforts on axle recognition in real time are still inadequate, especially for accelerometer-based BWIM system. In this paper, we propose a two-step solution for real-time vehicle identification designed for acceleration measurements. In this method, a sequence-to-label long–short-term memory (LSTM) network is constructed to identify axle-induced responses in a multi-lane system directly. The input sequence is wavelet coefficients after performing wavelet transform on the raw data. Based on the trustworthy axle identification results, an auto-grouping step is then proposed and applied for vehicle-type identification. Model training and method evaluation are conducted using filed measurements from a highway bridge in Tokyo. Two data sets are utilized, i.e., 191 vehicles with 456 axles and 596 vehicles with 1380 axles. Results show that 98% axles can be identified correctly using proposed LSTM method from both data sets, while accuracy of vehicle-type identification is 96% for both data sets, which can demonstrate the robustness of proposed methods. Moreover, the driving lane detection of all detected vehicles is 100% without any failed cases. Comparing with all-in-one deep network using acceleration measurements as input sources directly, the proposed two-step LSTM method requires less training data, hence it is a computationally efficient solution, which would enable its generalization capability for applying on other bridges.

Keywords Bridge weigh-in-motion · Long–short-term memory (LSTM) · Accelerometer · Axle detection · Vehicle identification · Wavelet transform

1 Introduction

Bridge weigh-in-motion (BWIM) system plays an essential role in vehicle load monitoring, which can infer axle weights according to field measurements, to provide supports for overweight vehicle identification and bridge condition assessment [1–4]. Therefore, the accurate vehicle detection is the foundation for BWIM system, which means those axle-induced responses should be captured by transducers properly and determined by reliable techniques precisely.

Since the first introduce of BWIM system by Moses in 1970's [1], the vehicle-detecting transducers have been upgrading from the early on-road sensors, i.e., tape switches and pneumatic tubes, to strain-based sensors [2, 3]. This is because strain gauges are normally installed underneath the bridge to avoid direct exposure to the traffic, hence they are more durable compare to on-road sensors. Moreover, the installation or maintenance of strain gauges do not require

✉ Yanjie Zhu
zhuyanjie2018@163.com

¹ Department of Urban and Civil Engineering, Tokyo City University, 1-28-1 Tamazutsumi, Setagaya-ku, Tokyo 158-8557, Japan

² Department of Bridge Engineering, School of Transportation, Southeast University, Nanjing 210096, China

³ RIKEN Center for Advanced Intelligence Project, 1-4-1 Nihonbashi, Chuo, Tokyo 103-0027, Japan

⁴ Graduate School of Information Sciences, Tohoku University, 6-3-09 Aoba, Aramaki-aza Aoba-ku, Sendai 980-8579, Japan

⁵ Maintenance and Traffic Management Department, Metropolitan Expressway Company Limited, 1-4-1 Kasumigaseki, Chiyoda-ku, Tokyo 100-8930, Japan

traffic interruption. In recent years, acceleration-based sensors have been proposed to replace strain-based detectors for BWIM system [5], because accelerometers are more durable and convenient in terms of installation and maintenance when comparing with strain gauges.

Efforts have been devoted to developing methods of force estimation and displacement calculation for acceleration-based BWIM system [6–8]. However, rarely efforts have been made on automatic and real-time axle identification, which is an essential task for weighting traffic load and overweight traffic recognition. Recently, we have proposed a shallow convolutional neural network-based method for this acceleration-based BWIM system to realize real-time traffic monitoring [9]. In that paper, we adopted the convolutional neural network (CNN) with continuous wavelet transform (CWT), and decomposed the vehicle identification task into three sub-tasks to avoid the high computational cost of using an all-in-one deep network [10]. We first proposed a stable CNN classifier to locate the periods, called valid sequence, containing a single vehicle on one driving lane or multi-vehicles on different lanes individually. After that, an adaptive CWT is developed for valid axle localization, which is based on the amplitude features of both raw measurements and the wavelet coefficients. Finally, the driving lane can be identified through a cross-comparing process among all driving lanes. In that paper, the valid axle localization is the most time-consuming task due to the thresholds having to be manually determined, i.e., we have to investigate the axle-induced vibration features both in the time domain and the wavelet coefficients to set two proper thresholds to find the valid axle. Therefore, for automatic vehicle identification, we, furthermore, present a more simplified and straightforward deep learning method in this paper. The proposed method conducts axles identification by adopting a sequence-to-label long–short-term memory (LSTM) network, and thereafter assort independent axles into detached vehicles automatically.

As demonstrated in previous paper [9], wavelet analysis is an effective tool for time–frequency analysis specially to capture abrupt changes in non-stationary signals, i.e., the axle-induced vibrations. Therefore, the wavelet analysis method is employed to prepare the input sources for LSTM network instead of using raw accelerations data to capture axle-induced features. LSTM networks, proposed by Hochreiter and Schmidhuber [11], have proved its ability in many sequence learning tasks [12], i.e., speech recognition, writing recognition, and sequence anomaly detection. The ability of LSTM in maintaining long-term memory enables it to learn long-term dependencies between time-steps of sequential data. The application of LSTM network can be classified into sequence-to-sequence or sequence-to-label for either prediction or classification. For instance, Malhotra et al. [13] established an sequence-to-sequence

encoder–decoder using LSTM only to detect anomalies in time-series. They trained this prediction model using only normal samples and the model is expected to reconstruct ‘normal’ sequences well with a small error. Hence when an unknown instance being reconstructed with higher reconstruction error, which represents this input sample is not belongs to ‘normal’ data set. Ma et al. [14] employed LSTM directly for traffic speed prediction based on the data recorded by microwave traffic detector. More complicated model can be found by combing CNN and LSTM together. For example, to identify automotive suspension state, Luo et al. [15] employed CNN for feature extraction and LSTM for further damage values prediction. In terms of sequence-to-label learning task, a combination of CNN and LSTM method have also been conducted by Zhou et al. [16] for text classification and by Zhao et al. [17] for speech emotion recognition.

The method we introduced for automatic axle detection is employing LSTM network for sequence-to-label classification. The input sequence is generated by applying wavelet transform on raw acceleration data. Based on axle detection results, an auto-grouping method is further applied to classify those identified axles into detached vehicles. Again, this proposed method is aiming to fulfill the automatic vehicle detection function for real-time traffic monitoring. Comparing with an all-in-one deep network proposed by Kawakatsu et al. [10], the method we proposed is computation efficient due to the task division. Moreover, comparing with our previous proposed method [9], this method is more simplified and streamlined.

The rest of this paper starts with a brief description of the acceleration-based axle identification system, as described in Sect. 2. Then theoretical background and implementation procedures of proposed method is presented in Sect. 3, followed by a case study in Sect. 4. The developed method is assessed using field test data collected from a highway bridge in Tokyo. Details of data preparation, model training with fine-tuning process, and performance evaluation will be described and outlined in Sect. 4. Then Sect. 5 offers a conclusion on this paper.

2 Acceleration-based BWIM system

The configuration of an acceleration-based BWIM system can be found in [5]. In terms of axle identification, two groups of detectors are placed in both entry and exit sides in the longitudinal direction, seeing Fig. 1 as an example of three-lane system. In the transversal direction of entry or exit side, detectors are allocated for each driving lane. When a vehicle is moving across those monitoring lines, the axle-induced vibrations can be captured by those detectors. Figure 1 also provides an example when a four-axle truck

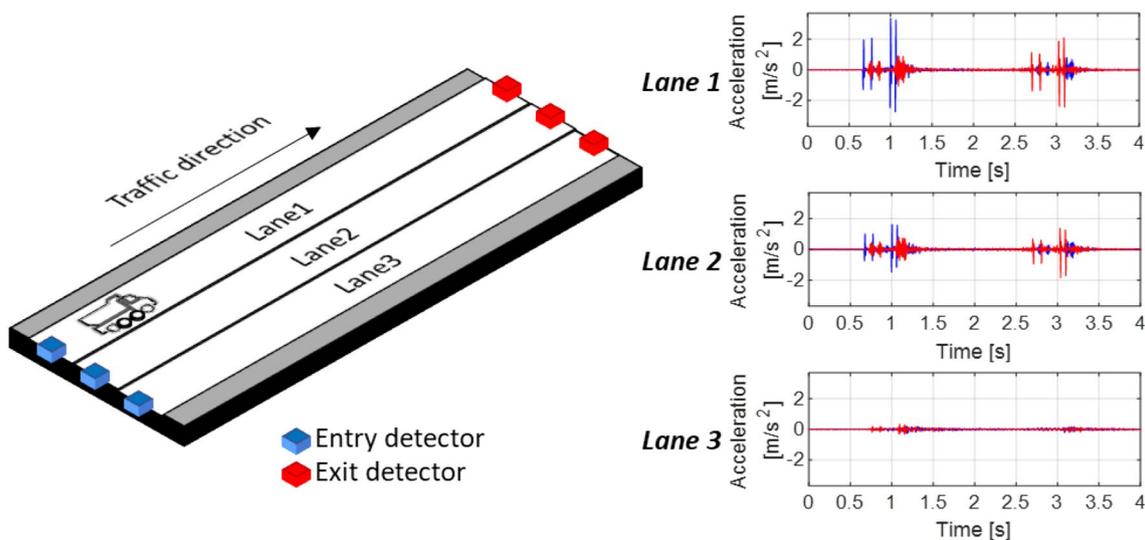


Fig. 1 Schematic diagram of acceleration-based system for traffic identification

driving through the leftmost driving lane. As can be seen, the axle-induced vibration is nonstationary, in which four peaks are apparent from the raw measurements.

3 Method and implementation

3.1 Wavelet analysis of vehicle-induced responses

To detect the nonstationary signals induced by vehicles, it is necessary to keep the time information while considering the frequency spikes. Therefore, wavelet analysis is adopted for data preprocessing before deep learning model. Wavelet transform is a well-known and sufficient tool for frequency analysis, especially to analyze abrupt changes, in time domain [18]. Wavelet transform (WT) on time series, $f(t)$, can be expressed in the following equation:

$$WT(\alpha, \tau) = \frac{1}{\sqrt{\alpha}} \int_{-\infty}^{\infty} f(t) \times \psi\left(\frac{t-\tau}{\alpha}\right) dt \tag{1}$$

where ψ is called mother wave. Two key wavelet concepts that contribute the power of WT are scaling parameter, α , and shifting parameter, τ . A compressed wavelet, with small scaling value, helps in capturing the abrupt changes, while a stretched wavelet helps in capturing the slowly varying changes. We employ continuous wavelet transform (CWT), which means scale parameter are varying continuously, in this paper to provide an overcomplete representation of raw measurements.

In this paper, the morse wavelet [19], seeing Eq. (2), is selected:

$$\psi_{\beta,\gamma}(\omega) = 2\left(\frac{e\gamma}{\beta}\right)^{\beta/\gamma} \omega^\beta e^{-\omega^\gamma} \times U(\omega) \tag{2}$$

where $U(\omega)$ is Heaviside step function, γ controls the symmetry of Morse wavelet, and $P^2 = \beta\gamma$ is time–bandwidth product. More explanations of Morse wavelet can be found in [20]. We use the default values of symmetry parameter ($\gamma = 3$) and time–bandwidth product (as $P^2 = 3.1$) in MATLAB for axle detection.

Figure 2a illustrates wavelet transformed results by applying WT on acceleration measurements, which is recorded by the Lane 1 entry detector, as previously shown in Fig. 1. The zoom in view of one axle is also provided in Fig. 2b.

3.2 Proposed LSTM classifier for axle identification

In this paper, we employ LSTM network to build a sequence-to-label classifier, seeing Fig. 3, for axle detection and traveling lane identification. The core layers of this classifier include a sequence input layer, hidden layers, and a label output layer. In hidden layer, LSTM layer and fully connected layer are stacked. Descriptions of core layers and core components, i.e., loss function, optimizer, assessment criteria, are explained in the rest part of this subsection.

3.2.1 Sequence input layer and label output layer

A sequence input layer holds the sequences prepared for the network. We proposed to input wavelet coefficient sequences instead of the raw vibration data into the network. Hence, the input sequence is a group of wavelet coefficients after performing wavelet transform on acceleration measurements

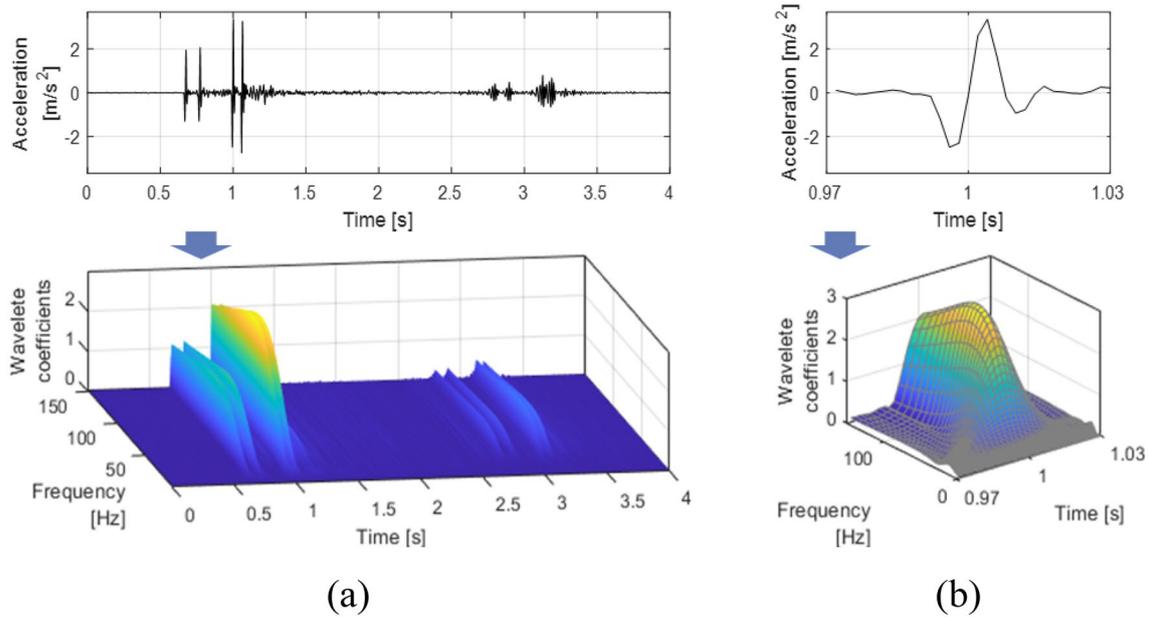


Fig. 2 Example of wavelet analysis on vehicle-induced vibration: **a** Wavelet analysis on lane 1 entry signal shown in Fig. 1; **b** the 3rd axle of (a)

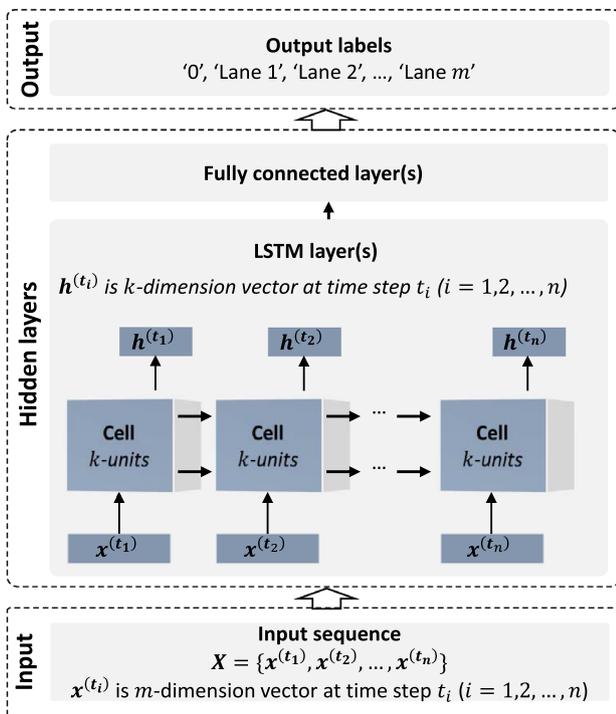


Fig. 3 Configuration of LSTM-based sequence-to-label classifier for axle identification

of all lanes, i.e., m -dimensional vector of readings from m detectors. Considering the complicated traffic conditions in a multi-lane system, we use a short time-period measurement, which can be obtained by shifting a window, size of n , over

a long time-series. The window size is designed to contain only one axle-induced vibrations.

Taking the entry signal previously described in Fig. 1 as an example to depict the detailed process, this four-axle vehicle is passing a three-lane system using driving lane no.1. A small window, with 0.06 s length, containing its third axle are displayed in Fig. 4. The first row of Fig. 4 shows the wavelet transformed results from three lanes. By doing that, the time-dependent wavelet coefficients at a certain frequency level can be selected to formulate the input sequence. This certain frequency level is determined based on the maximum wavelet coefficients, for instance, the outer edge line showing in the second row of Fig. 4. Hence, the final sequence input, X , is the normalized outer edge lines from all three lanes, where $m = 3$, and $t_n = 0.06s$.

The final label output layer is designed to predict the driving lane label. If the input sequence records the vibration due to a valid axle, the target output of the model is to identify the location of this axle, i.e., in lane 1, 2 or m , here $m = 3$. If the output label is '0', it represents the input sequence dose record any valid axle.

3.2.2 Hidden layers

The first major component inside hidden layers is LSTM layer, which is the main module to learn the long-term dependency features of the input sequence. As indicated in Fig. 3, the LSTM cell, with k units, is operated on input data $x^{(i)}, i = 1, 2, \dots, n$. Hence the m -dimension input $x^{(i)}$ can be interpreted into a k -dimension vector, $h^{(i)}$.

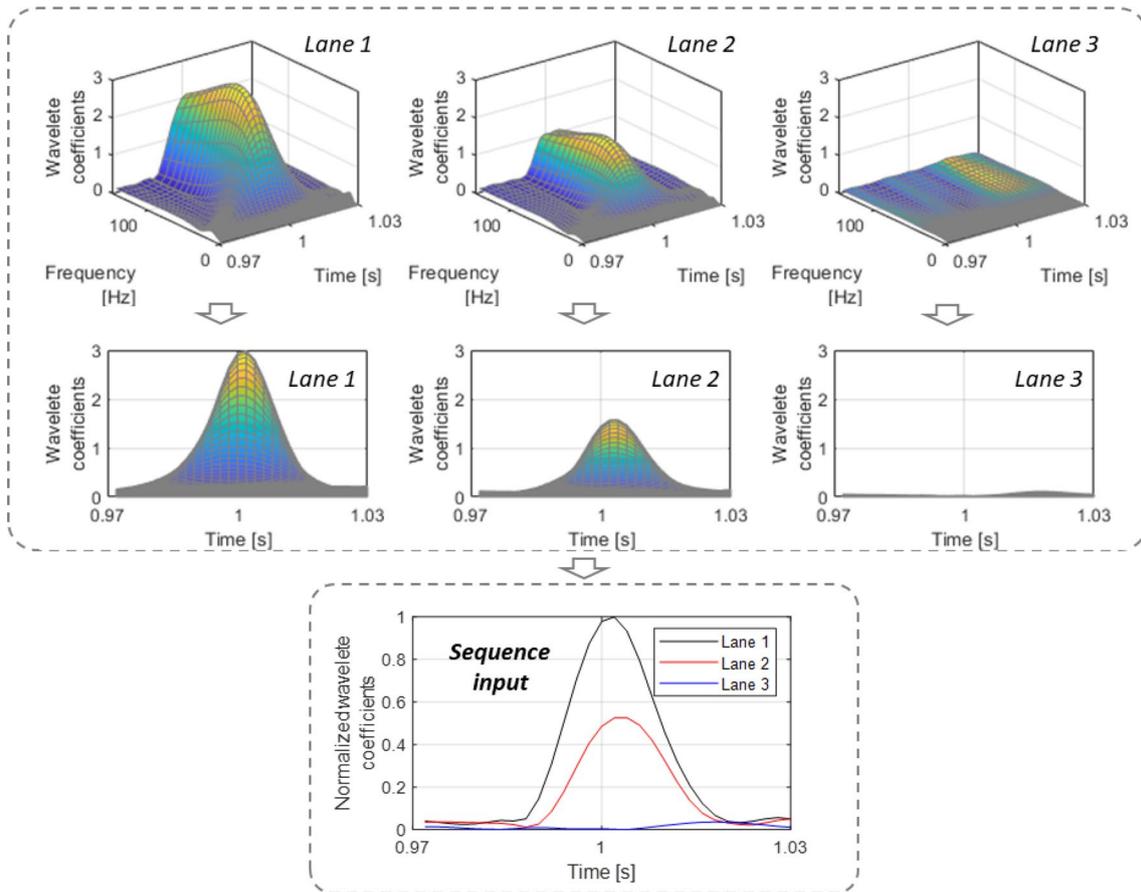


Fig. 4 Demonstration of sequence input for LSTM classifier

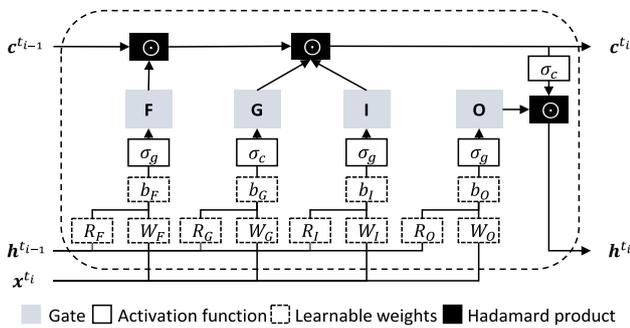


Fig. 5 Structure of LSTM cell

Figure 5 illustrates the structure of a LSTM cell at time step $t_i \in \{t_1, \dots, t_n\}$. The output at time step t_i includes cell state $c^{(t_i)}$ and hidden state $h^{(t_i)}$, which are the accumulation of current input value $x^{(t_i)}$, preceding cell state $c^{(t_{i-1})}$, and preceding hidden state $h^{(t_{i-1})}$. Hence, information can be transferred from previously processed states and affects the final output.

The transfer mechanism is controlled by four gates, named forget gate (F), candidate gate (G), input gate (I),

and output gate (O). The computation steps are represented in Eqs. (3–7) [21], where gate F controls information expurgation from $x^{(t_i)}$ and $h^{(t_{i-1})}$, while gate I and G control information adding and updating from $x^{(t_i)}$ and $h^{(t_{i-1})}$. The hidden representation $h^{(t_i)}$ is then calculated using Eq. (8), which is computed according to current cell state $c^{(t_i)}$ and gate O :

$$F^{(t_i)} = \sigma_g \left(W_F x^{(t_i)} + R_F h^{(t_{i-1})} + b_F \right), \tag{3}$$

$$G^{(t_i)} = \sigma_c \left(W_G x^{(t_i)} + R_G h^{(t_{i-1})} + b_G \right), \tag{4}$$

$$I^{(t_i)} = \sigma_g \left(W_I x^{(t_i)} + R_I h^{(t_{i-1})} + b_I \right), \tag{5}$$

$$O^{(t_i)} = \sigma_g \left(W_O x^{(t_i)} + R_O h^{(t_{i-1})} + b_O \right), \tag{6}$$

$$c^{(t_i)} = F^{(t_i)} \odot c^{(t_{i-1})} + I^{(t_i)} \odot G^{(t_i)}, \tag{7}$$

$$h_t = O^{(t)} \odot \sigma_c \left(c^{(t)} \right), \quad (8) \quad m^{(t)} = \beta_1 \cdot m^{(t-1)} + (1 - \beta_1) g^{(t)} \quad (11)$$

where σ_g is gate activation function and σ_c is the state activation function, \odot represents Hadamard product. Apparently, the learnable weights of an LSTM layer contains input weights $W = \{W_F, W_G, W_I, W_O\}$, recurrent weights $R = \{R_F, R_G, R_I, R_O\}$, and bias $b = \{b_F, b_G, b_I, b_O\}$. The memory flow of LSTM network is thus formed based on the transferred information among units, which enables its ability to process sequences with feature of long-term spatial and temporal dependencies. LSTM layers can be stacked, units inside each LSTM layer can be fine-tuning for different tasks.

The second part of hidden layer is the fully connected layer, which can also be stacked. The function of using fully connected layer is to utilize features learned from LSTM layers for classification. The hyperparameters lie in are the number of stacked layers and the number of neurons in each layer.

3.2.3 Loss function and optimizer for model training

During model training, the categorical crossentropy loss function is adopted to compute the crossentropy loss between ground-truth labels and predicted labels. The target of model training process is to minimize this cost function, seeing Eq. (9):

$$\text{Min Loss} = - \sum_{i=1}^N p(y_i) \cdot \log(q(y_i)) \quad (9)$$

where y_i is the i th segment of a multi-categories data set. p and q refer to the probability distribution of the ground-truth value and its corresponding predicted value, respectively. N is the total number of training segments.

In terms of optimizer, Adam [22] is selected as the optimization algorithm because of its robustness in hyperparameters selection during training process. Adam is designed to compute adaptive learning rates for different parameters based on the estimation of gradients of the objective function for each parameter. For the sake of simplicity, the parameter and time is represented by θ and t , respectively. The objective function for parameter θ_i at time step t_j can be expressed as $J(\theta_i, t_{j-1})$. Hence, the gradient of this objective function can be expressed as in Eq. (10):

$$g_{\theta_i}^{(t)} = \nabla_{\theta_i} J(\theta_i, t_{j-1}) \quad (10)$$

At each time step t_i , Adam is updating the exponential moving averages of the gradients, termed as $m^{(t)}$, and the squared gradients, termed as $v^{(t)}$, seeing Eqs. (11–12):

$$v^{(t)} = \beta_2 \cdot m^{(t-1)} + (1 - \beta_2) \left(g^{(t)} \right)^2 \quad (12)$$

where β_1 and β_2 are the exponential decay rate for the first and second moment estimates, respectively. The parameter is further updated using the following equation:

$$\theta_i^t = \theta_i^{t-1} - \alpha \cdot m^{(t)} / \left(\sqrt{v^{(t)}} + \varepsilon \right) \quad (13)$$

where α is the learning rate and ε is a small constant number to prevent division by zero. The hyperparameter within optimizer which are explored in this paper including learning rate α and constant value of ε for numerical stability.

3.2.4 Assessment criteria

For model assessment, we consider the normal percentage criteria, including accuracy, precision, recall, R1-score. We also evaluate the model using accuracy, precision, recall, F1-score, and Cohen's kappa [23]. As indicated by Cohen, kappa values over 0.81 represents almost perfect agreement.

3.3 Auto-grouping step for vehicle identification

The auto-grouping method is operating on each lane individually to identify vehicles from all detected axles. According to the real driving condition, two vehicles in the same lane must keep a certain distance. Hence, axles in the same lane within a certain period, $t_{limitation}$, should be classified as one vehicle. The value of $t_{limitation}$ are determined based on an investigation of the data set.

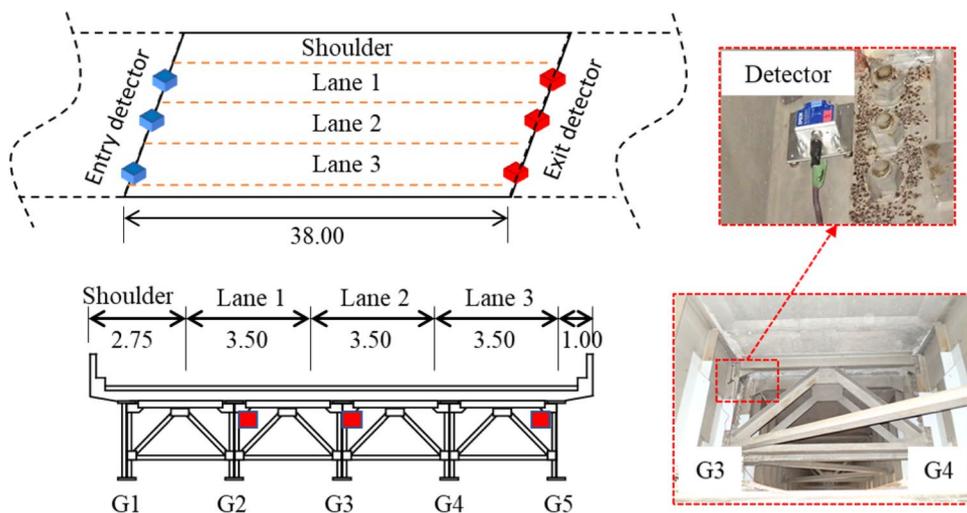
4 Field experiment for method evaluation

To evaluate the proposed vehicle detection method, the acceleration-based BWIM system has been installed to a highway bridge located in Tokyo. It is a 38 m single-span steel bridge with three driving lanes. Figure 6 provides cross-sectional view and sensors configuration for vehicle detection. MEMS accelerometers for vehicle identification are placed at longitudinal edges underneath the bridge deck.

4.1 Data preparation

To prepare the input sequence, the certain frequency level should be decided first. As suggested, the frequency level corresponding to the maximum wavelet coefficients can be selected first. We conducted an examination on 456 axles. The distribution of maximum wavelet coefficients

Fig. 6 Sensor configuration on a highway bridge in Tokyo



corresponding frequency is displayed in Fig. 7a, which shows this certain frequency level 85 Hz.

As aforementioned, small window-sized data are suggested for a multi-lane system to ensure only one valid axle is expected within a window. Hence the window size can be determined according to the vehicle’s limited speed, in this case $t_n = 0.06s$ is considered, and $n = 30$ data points corresponding to 500 Hz sampling rate. By shifting the window along time series, segments for either model training and test can be obtained. The number of segments for model training and test are indicated in Fig. 7b. As can be seen, training set consist 2472 invalid segments with 2484, 3057, 276 samples belonging to ‘Lane 1’, ‘Lane 2’, and ‘Lane 3’, respectively. The test data set contains 456 valid axles belongs to 191 vehicles that travelling in lane 1, 2 or 3, and 275 invalid segments.

Less training samples for category ‘Lane 3’ can be observed according to Fig. 7b due to the training samples

preparation method. For an unknown monitoring system, the easiest way to obtain the training data set is to capture peaks over a certain level, $2.0m/s^2$ in this case, from the whole measurement data bank. Because the amplitude over $2.0m/s^2$ is definitely the axle-induced variations, not the noise-induced vibration. For the application in other bridge, the pre-investigation about the noise-induced vibration should be first taken to determine this threshold. However, the drawback is also obvious. The axle which can induce over $2.0m/s^2$ is normally the heavy vehicles. Lane 3 is the rightmost lane, which means the passing lane in Japan; therefore, fewer heavy vehicles are chosen to drive in this lane. To solve this problem, we will consider the weighted factor for unequal distributions in the following training process. The factor is calculated according to Eq. (14), where N_i is the number of samples of the i th category. For instance, the weighted factor for category ‘0’,

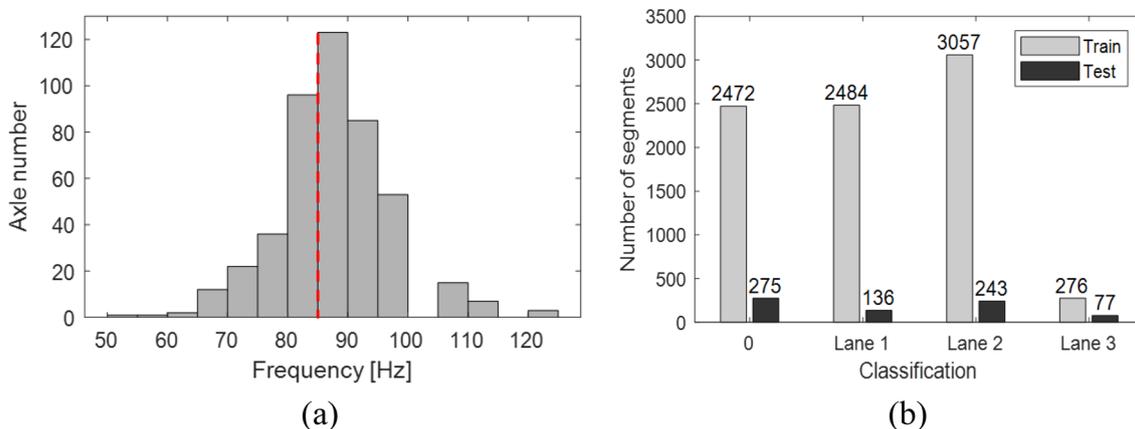


Fig. 7 Data preparation: **a** distribution of frequency level corresponding to the maximum wavelet coefficients; **b** number of segments in training and test data sets

'Lane 1', 'Lane 2' and 'Lane 3' are 1.2, 1.2. 1, and 11.1, respectively, according to Fig. 7b:

$$\text{Weighted factor} = \frac{1}{N_i} \quad (14)$$

4.2 Axle identification

4.2.1 Model training

The investigation on hyperparameters is conducted in terms of hidden layers depth and optimizer settings. The model is trained for 50 epochs and batch size sets to 20, which means 50 full passes on the training data set using a batch size of 20.

We first evaluated the hyperparameter related to the hidden layer depth, including five configurations of various LSTM layer and three configurations of various fully connected layer. Figure 8a displays the five configurations we investigated, while Fig. 8b compares their performance from five aspects, i.e., precision, recall, F1-score, kappa, and time cost. For all model configuration shown in Fig. 8a, the learning rate decays from 10^{-4} using the adaptive decay schedule. The exponential decay rate β_1 and β_2 are set as 0.9 and 0.999. The constant value of ϵ is set to 10^{-7} .

As can be seen from Fig. 8b, model with one-LSTM-layer, i.e., configuration no.1 and 2, shows unstable performance reflecting by the fluctuation in precision, recall, F1-score and kappa score. When deepening the model to two stacked LSTM layers, i.e., configuration no. 3 and 4, positive effect can be observed with higher precision. Moreover,

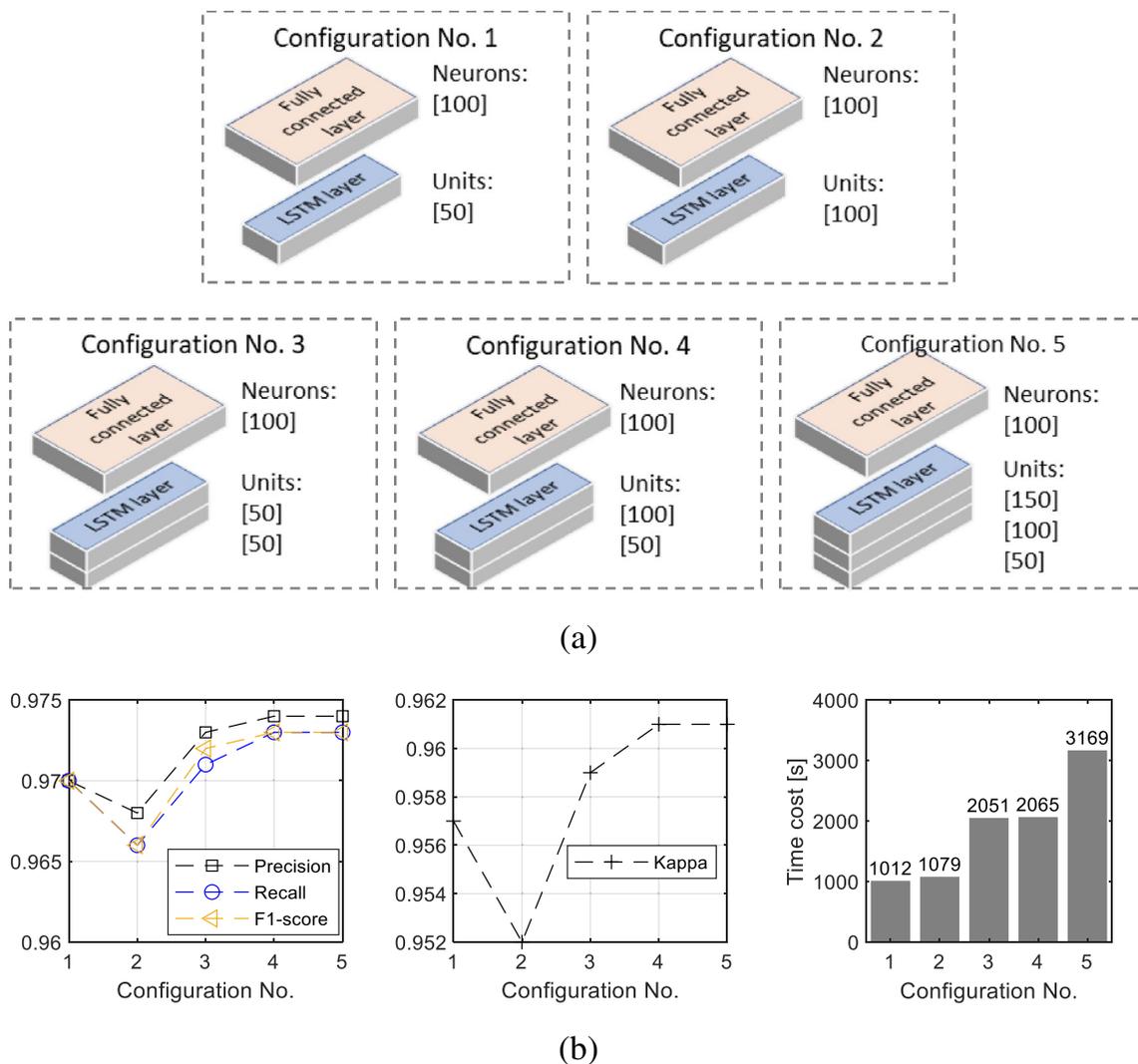


Fig. 8 Hyperparameter examination of LSM layers: **a** five different configurations of LSTM layers; **b** assessment of those five different configurations in (a)

increasing the number of LSTM units can also improve the model performance by 0.001 for precision and F1-score, by 0.002 for recall and kappa score. However, there is no evidence of improvement when stacking three LSTM layers, instead of the training time, which is increasing. Therefore, the suggestion for LSTM layer depth is configuration no. 4.

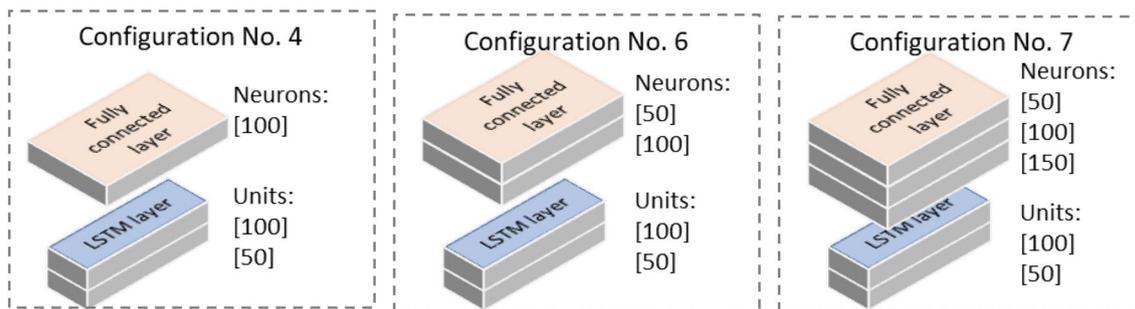
The investigation on fully connected layers are further conducted on three configurations, showing in Fig. 9a, while the LSTM layers are same. Unlike the trend observed from various configurations of LSTM layers, minimal change can be observed among three configurations of fully connected layers, as demonstrated in Fig. 9b. More specifically, the precision didn't show any evident improvement when the fully connected layer is getting deeper, i.e., 0.974 for one-layer model, 0.973 for two-layer model, and 0.975 for three-layer model. In the meantime, the training time, while increasing the stacked fully connected layers is also stable with a slight variation from 2059 to 2090s, which can also reflect the configuration of fully connected layer has a slight effect on the final results. Therefore, the suggested configuration for fully connected layer is no. 7. Exact criteria values of aforementioned seven configurations shown in Figs. 8b and 9b can be found in Table 1.

The hyperparameters within Adam optimizer that are explored in this paper include learning rate and constant

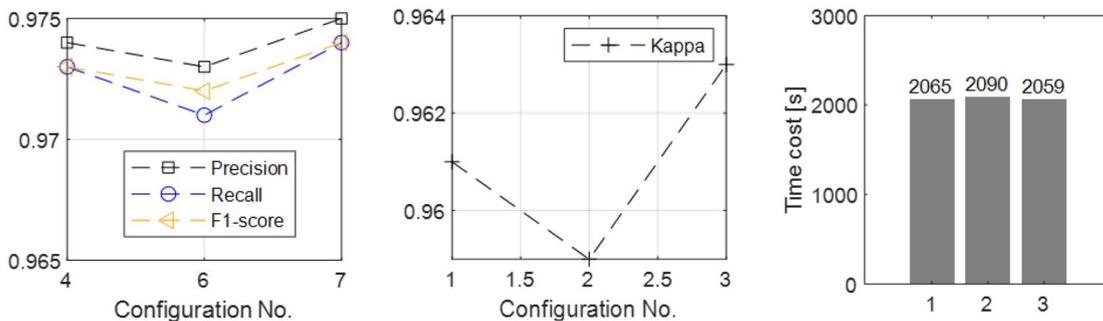
value of ϵ as previously mentioned in Eq. (13). The exponential decay rate β_1 and β_2 are suggested as 0.9 and 0.999 for adaptive decay. When evaluating, the structure of hidden layer includes two stacked LSTM layer (100 and 50 units, respectively) and one fully connected layer with 100 neurons. Figure 10 presents the comparison of seven conditions with various configuration among Adam optimizer, details are listed in Table 2. According to Fig. 10 and Table 2 the best performance can be observed when adopting the initial learning rate as 10^{-4} with adaptive decay schedule, and the constant value of $\epsilon = 10^{-8}$.

Table 1 Investigation of model depth in hidden layer

Configuration no.	Precision	Recall	F1-score	Kappa	Time cost [s]
1	0.970	0.970	0.970	0.957	1012
2	0.968	0.966	0.966	0.952	1079
3	0.973	0.971	0.972	0.959	2051
4	0.974	0.973	0.973	0.961	2065
5	0.974	0.973	0.973	0.961	3169
6	0.973	0.971	0.972	0.959	2090
7	0.975	0.974	0.974	0.963	2059



(a)



(b)

Fig. 9 Hyperparameter examination of fully connected layers: **a** three different configurations fully connected layers; **b** assessment of those three different configurations in (a)

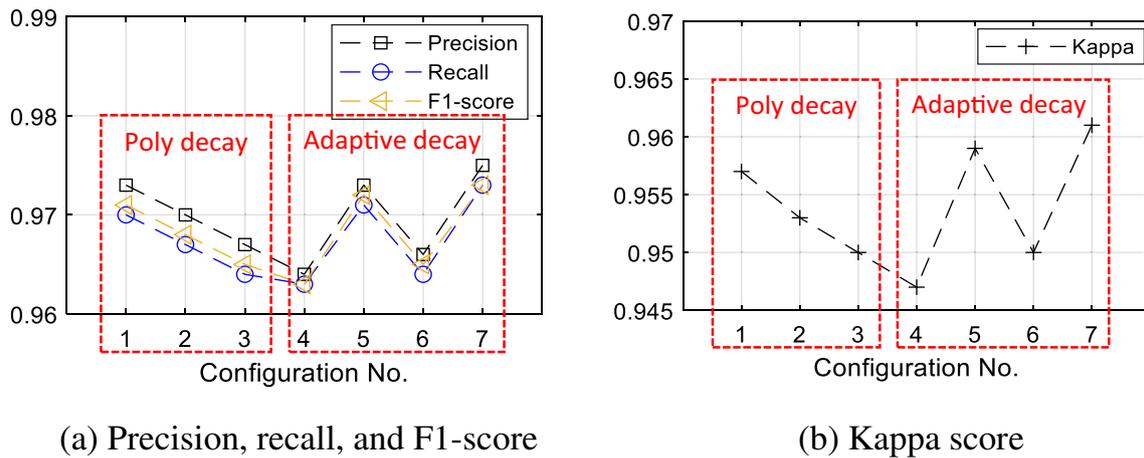


Fig. 10 Assessment results of hyperparameter examination of Adam optimizer

Table 2 Investigation of hyperparameters in Adam optimizer

	Initial α	Learning rate decay	ϵ	Precision	Recall	F1-score	Kappa
1	10^{-3}	Polynomial	10^{-7}	0.973	0.970	0.971	0.957
2	10^{-4}	Polynomial	10^{-7}	0.970	0.967	0.968	0.953
3	10^{-5}	Polynomial	10^{-7}	0.967	0.964	0.965	0.950
4	10^{-3}	Adaptive	10^{-7}	0.964	0.963	0.963	0.947
5	10^{-4}	Adaptive	10^{-7}	0.973	0.971	0.972	0.959
6	10^{-5}	Adaptive	10^{-7}	0.966	0.964	0.965	0.950
7	10^{-4}	Adaptive	10^{-8}	0.975	0.973	0.973	0.961

4.2.2 Model assessment

According to above hyperparameter investigation, the final model is configured as two stacked LSTM layer with 50 and 100 LSTM units, respectively, followed by three fully connected layer with 150, 100 and 50 neurons individually. The Adam optimizer is selected with 10^{-4} initial learning rate and decay adaptively. The constant value is setting as 10^{-8} . Figure 11 shows the axle detection and localization results on 191 vehicles of 456 axles. Apparently, the identification results are 98%, 99% and 97% for classifying segments belongs to 'Lane 1', 'Lane 2', and 'Lane 3'. The overall precision is 0.976, recall and F1-score is 0.974, while Kappa equals 0.963. The accuracy is 98%, while the mis-detected rate is 2%, seeing Table 3.

We further evaluate the proposed LSTM classifier on the second data set contains 596 vehicles with 1380 axles, the accuracy can also reach 98%, which can demonstrate the convinced performance of proposed LSTM classifier for axle identification.

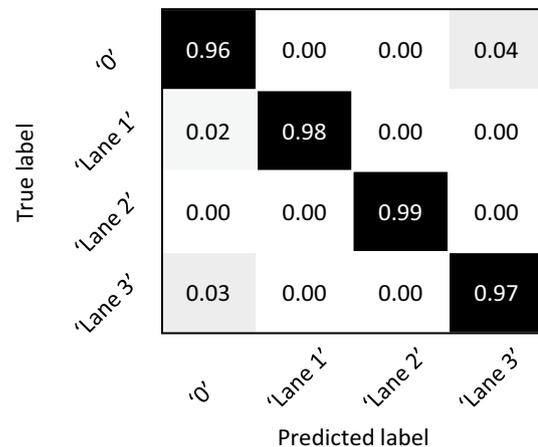


Fig. 11 Confusion matrix for axle identification

4.3 Vehicle identification

Figure 12 provides the visualization of axle detection and localization results using proposed LSTM classifier. Detected axles are independent. The following task is to group axles on the basis of vehicles. The process of the

Table 3 Axle identification results on two data sets

	Axle identification results on data set 1	Axle identification results on data set 2
Expected number	456	1380
Detected number	447	1353
Fail to detect	9	27
Accuracy	98.0%	98.0%
Failure detection rate	2.0%	2.0%

auto-grouping step has been described in previous section. According to the investigation on 191 vehicles, this certain period, $t_{limitation}$, can be set to 0.752 s, which related with the vehicle length and vehicle speed. Hence, continuous axles within 0.752 s will be automatically grouped as a single vehicle.

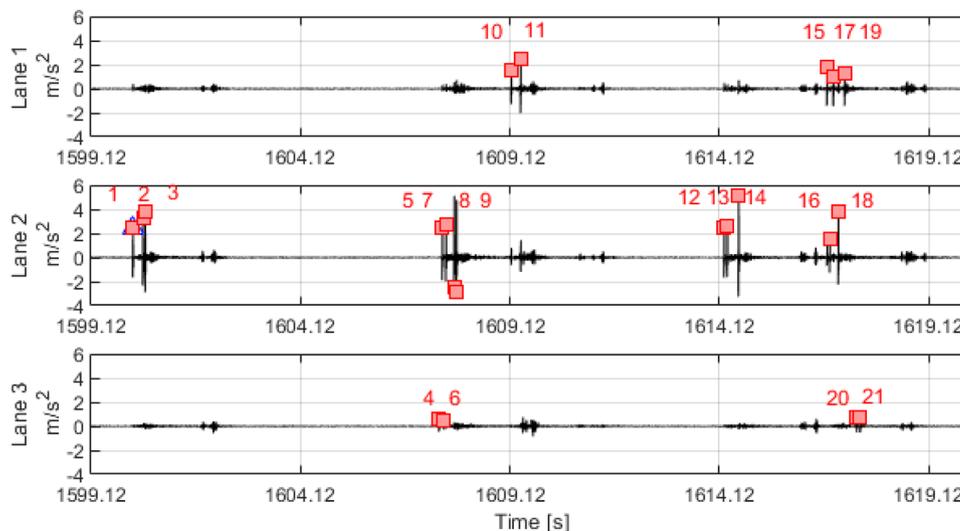
Due to the implementation of weighed factor for lane 3 segments during training process, the LSTM model is too sensitive to data from lane 3. Consequently, there have chances to detect fake axles, as shown in Fig. 13. After auto-grouping axles into independent vehicles, those detected fake axles can be classified into two groups. The first type is reflected by a vehicle with only one axle, i.e., the first fake axle shown in Fig. 13. The solution is that if the axle-induced vibration is less than $0.3m/s^2$, this detected axle will be ignored and removed. This threshold can be determined through the pre-investigation about the noise-induced vibration level when applying this method on the other bridges. This rule is also applied for the other two lanes. The second type of fake axles exists in a vehicle with multi-weight axles. For instance, the second and third fake axles shown in Fig. 13 are treated as

two axles of the detected vehicle. That vehicle should be 2-axle vehicle, instead of a 4-axle vehicle. The vibration induced by those detected four axles are $2.4m/s^2$, $0.5m/s^2$, $5.2m/s^2$, and $0.6m/s^2$, respectively. Apparently, the considerable disparity among those four axles is inconsistent with facts. Hence, an extra restriction will be applied as the post-processing step for the second type of fake axles. For a single vehicle, the deviation among all axle-induced vibrations should within 70%, where this 70% threshold is come from the evaluation results of 191 vehicles. Therefore, $0.5m/s^2$ and $0.6m/s^2$ can be identified as fake axle-induced variations.

Vehicle detection results using the first data set with 191 vehicles are presented in Fig. 14, in respect of vehicle-type identification and driving lane detection. Table 4 also summarizes the detailed results. According to Fig. 14a and Table 4, vehicle identification can achieve 96% precision, with 0.2% vehicles cannot be identified. The mis-judgement rate is 3.8% due to failure detection of eight axles which belong to eight two-axle vehicles. There also have two two-axle vehicles being mis-judged as three-axle vehicles and one is totally missed. As can be seen from Fig. 14b, the accuracy of driving lane detection, the accuracy is 100% for all detected vehicles.

The examination results on the second data set with 596 vehicles are presented in Fig. 15 and Table 5. The accuracy for vehicle-type identification is robust keeping at 95.9%, while failure detection rate is 0.3% due to four two-axle vehicles cannot be identified. The mis-judgements can also be observed, especially for two-axle vehicles with lightweight axles. The average mis-judgement rate is 3.9%. Reasons will be analyzed in the subsequent subsection. For all identified vehicles, the lane detection accuracy is 100%.

Fig. 12 Automatic axle identification



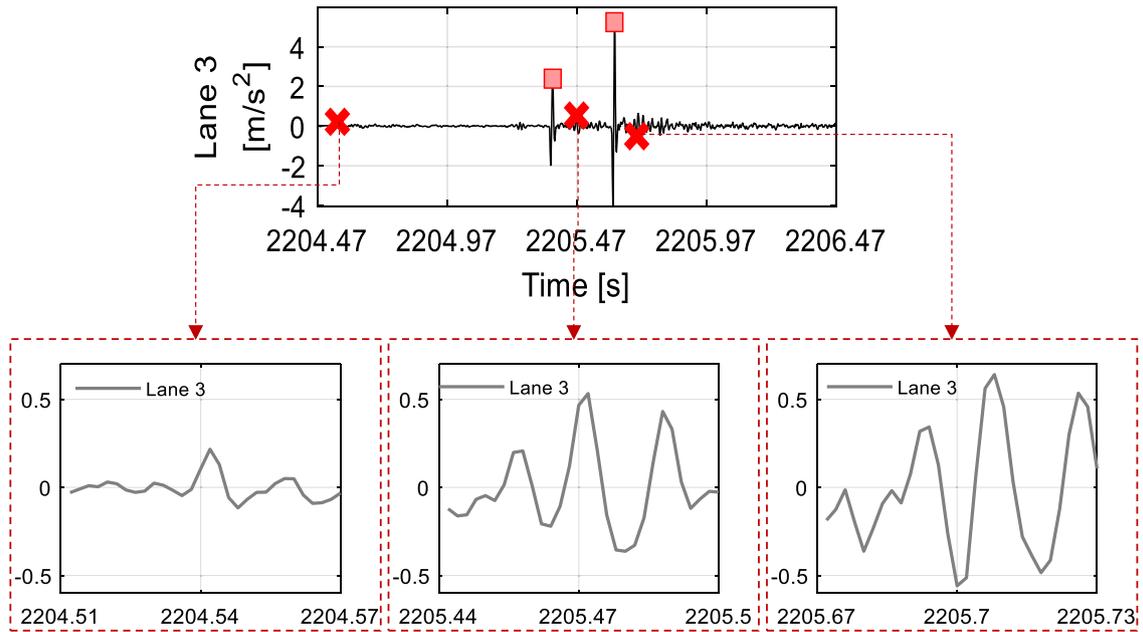


Fig. 13 Error detection due to model too sensitive to data from lane 3

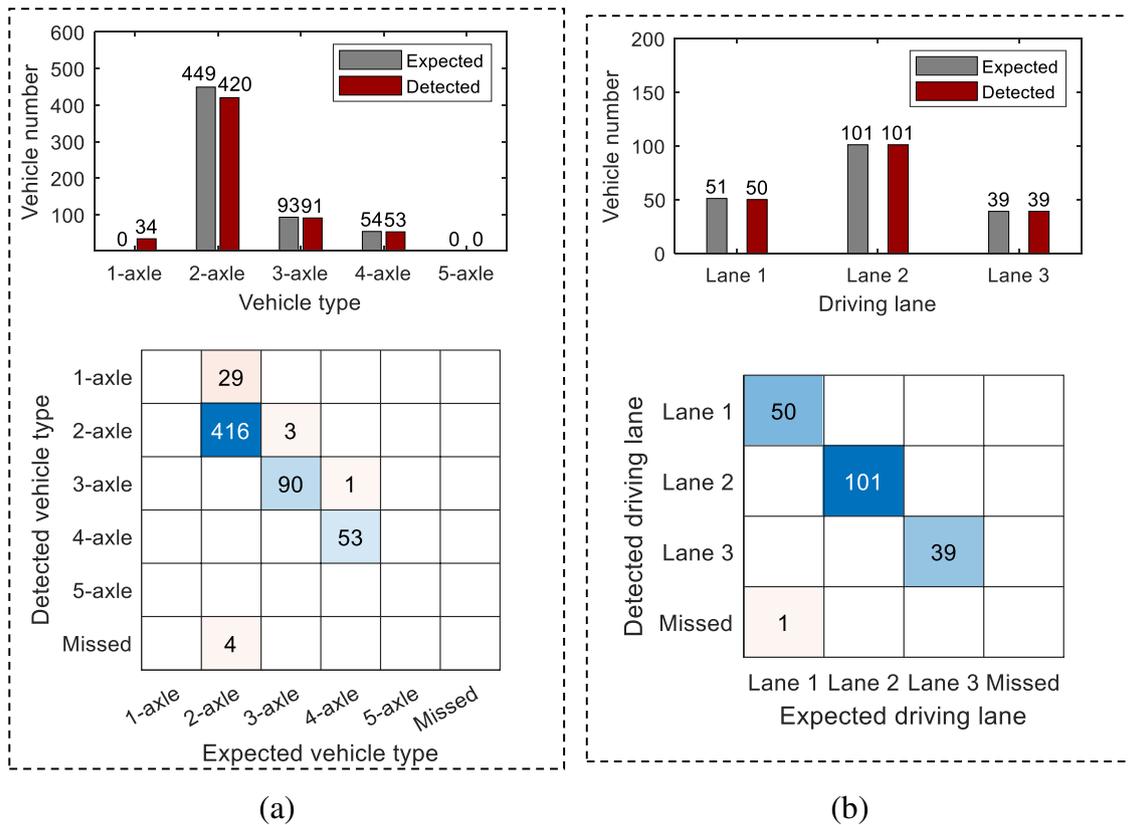


Fig. 14 Detection results on 191 vehicles: a vehicle-type identification; b driving lane detection

Table 4 Vehicle detection results on 191 vehicles

	Vehicle-type identification				Driving lane detection		
	2-axle	3-axle	4-axle	5-axle	Lane 1	Lane 2	Lane 3
Expected number	137	32	21	1	51	101	39
Detected number	128	32	19	1	50	101	39
Fail to detect	1	0	0	0	1	0	0
Mis-judge	8	0	2	0	0	0	0
Accuracy	96.0%				99.3%		
Failure detection rate	0.2%				0.7%		
Mis-judgement rate	3.8%				0.0%		

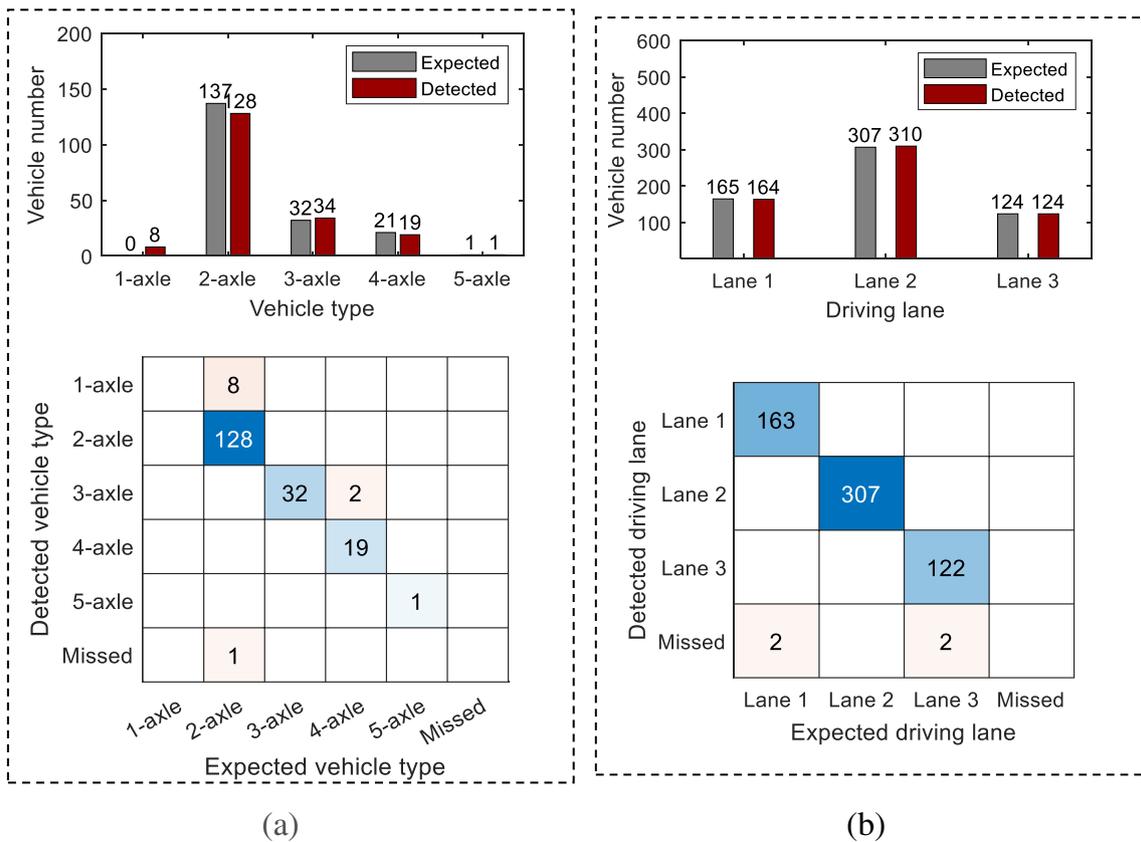


Fig. 15 Detection results on 596 vehicles: **a** vehicle-type identification; **b** driving lane detection

Table 5 Vehicle detection results on 596 vehicles

	Vehicle-type identification			Driving lane detection		
	2-axle	3-axle	4-axle	Lane 1	Lane 2	Lane 3
Expected number	449	93	54	165	307	124
Detected number	416	90	53	163	307	122
Fail to detect	4	0	0	2	0	2
Mis-judge	29	3	1	0	0	0
Accuracy	95.9%			99.1%		
Failure detection rate	0.3%			0.9%		
Mis-judgement rate	3.9%			0.0%		

4.4 Method limitation discussion

The proposed method is able to identify 98% axles; however, there are still 2% axles cannot be detected. Those failure cases will result in the mis-judgement of vehicle-type identification.

Two failure conditions can be summarized for these 2% undetected axles. The first typical failure occurs when axles induce bridge vibrations within 0.03 s. The proposed method is unable to identify the lighter axle. Figure 16a gives an example when two axles on adjacent lanes (lane 1 and 2). The third axle of the vehicle in lane 1 and the first axle of vehicles in lane 2 are inducing bridge vibration within 0.004 s, the proposed method can identify the heavy axle, i.e., the axle in lane 1, but will fail to identify the axle in lane 2. Same reason for missing the first axle of vehicle in lane

3 due to the time difference is 0.014 s. The second failure condition exist when the vehicles of interest is too light. The method is not sensitive enough to identify those small-scaled axles. An example is presented in Fig. 16b, where the second axle of vehicle in lane 3 is unable to be uncovered.

5 Conclusions

This paper presents a real-time and efficient two-step solution of axle identification especially designed for acceleration-based bridge weigh-in-motion (BWIM) system. We first introduced a sequence-to-label long-short-term memory (LSTM) classifier for axle detection and driving lane recognition. Depending on detected axles, an auto-grouping step is further applied for vehicle-type classification. Based on

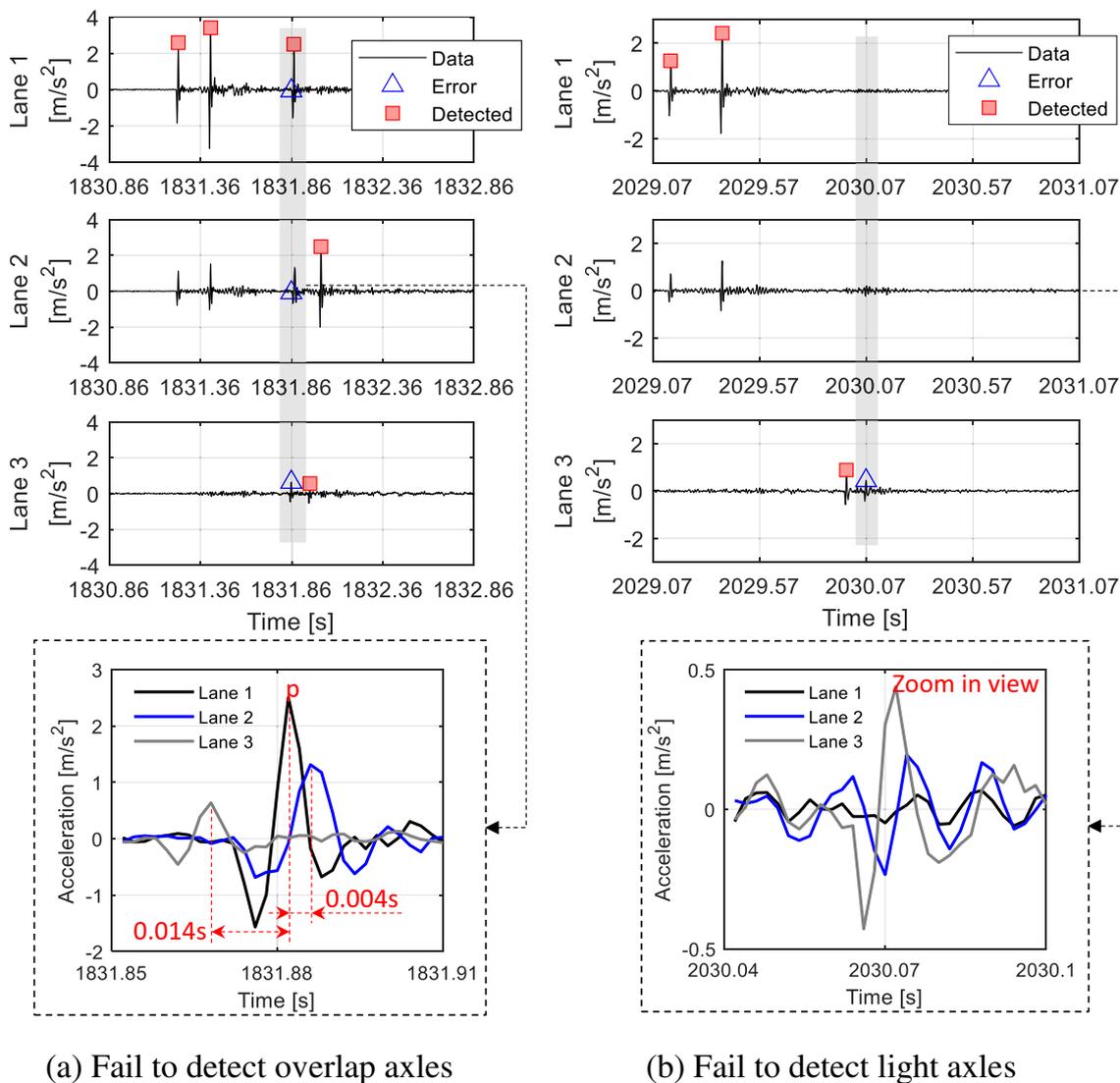


Fig. 16 Method limitation

the obtained results, highlights of this paper can be drawn as follows:

- Proposed LSTM classifier is robust to identify axle with 98% accuracy.
- The accuracy of vehicle-type identification can achieve 96% with high robustness.
- For all identified vehicles, the driving lane detection is 100% without any mis-judgements.
- The proposed method requires less training data and is a computationally efficient solution if comparing with an all-in-one deep network. Moreover, there is no restriction on bridge types, which enables its generalization capability.

However, limitations of proposed method still remain, i.e., 2% axles cannot be detected in our experiment. We have discussed those cases and classified them into two conditions. First is due to more than one axle are inducing vibrations within 0.03 s. In this case, the relative lightweight axle will be masked. The other failure occurs when the axle is too light to induce evident vibrations, i.e., the axle-induced vibration is only two times of the noise-induced vibrations.

In the future, more sensitive method should be considered for those lightweight axles. We will also make efforts on over-weight axle identified in real-time based on those detected axles.

Acknowledgements This study was carried out as cooperative research with the Tokyo Metropolitan Expressway Co., Ltd., Shutoko Engineering Co., Ltd., and the Highway Technology Research Center. The acceleration measurements were supported by Seiko Epson. This work is partial supported by the National Natural Science Foundation of China (Grant No. 52108118).

Data availability Some or all data, models, or code generated or used during the study are proprietary or confidential in nature and may only be provided with restrictions. The field measurements, i.e., acceleration data, were collected by Tokyo City University, based on a collaboration with the Metropolitan Expressway, Shutoko Engineering, and the Highway Technology Research Center. These data can be provided by the second author, if all partners agree with the provision. These data can only be used for research, and publishing results based on those data should be approved by the second author in advance.

References

- Moses F (1979) Weigh-in-motion system using instrumented bridges. *Transp Eng J ASCE* 105:233–249. <https://doi.org/10.1061/TPEJAN.0000783>
- Yu Y, Cai CS, Deng L (2016) State-of-the-art review on bridge weigh-in-motion technology. *Adv Struct Eng* 19:1514–1530. <https://doi.org/10.1177/1369433216655922>
- Lydon M, Taylor SE, Robinson D et al (2016) Recent developments in bridge weigh in motion (B-WIM). *J Civ Struct Heal Monit* 6:69–81. <https://doi.org/10.1007/s13349-015-0119-6>
- Lydon M, Robinson D, Taylor SE et al (2017) Improved axle detection for bridge weigh-in-motion systems using fiber optic sensors. *J Civ Struct Health Monit* 7:325–332. <https://doi.org/10.1007/s13349-017-0229-4>
- Sekiya H, Kubota K, Miki C (2018) Simplified portable bridge weigh-in-motion system using accelerometers. *J Bridge Eng* 23:04017124. [https://doi.org/10.1061/\(ASCE\)BE.1943-5592.0001174](https://doi.org/10.1061/(ASCE)BE.1943-5592.0001174)
- Sekiya H (2019) Field verification over one year of a portable bridge weigh-in-motion system for steel bridges. *J Bridge Eng* 24:1–13. [https://doi.org/10.1061/\(ASCE\)BE.1943-5592.0001411](https://doi.org/10.1061/(ASCE)BE.1943-5592.0001411)
- Morichika S, Sekiya H, Zhu Y et al (2021) Estimation of displacement response in steel plate girder bridge using a single MEMS accelerometer. *IEEE Sens J* 21:8204–8208. <https://doi.org/10.1109/JSEN.2021.3051697>
- Mohammed YM, Uddin N (2019) Acceleration-based bridge weigh-in-motion. *Bridge Struct* 14:131–138. <https://doi.org/10.3233/BRS-190143>
- Zhu Y, Sekiya H, Okatani T et al (2021) Acceleration-based deep learning method for vehicle monitoring. *IEEE Sens J* 21:17154–17161. <https://doi.org/10.1109/JSEN.2021.3082145>
- Kawakatsu T, Aihara K, Takasu A, Adachi J (2019) Deep sensing approach to single-sensor vehicle weighing system on bridges. *IEEE Sens J* 19:243–256. <https://doi.org/10.1109/JSEN.2018.2872839>
- Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Comput* 9:1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Wang J, Chen Y, Hao S et al (2019) Deep learning for sensor-based activity recognition: a survey. *Pattern Recognit Lett* 119:3–11. <https://doi.org/10.1016/j.patrec.2018.02.010>
- Malhotra P, Ramakrishnan A, Anand G et al (2016) LSTM-based encoder-decoder for multi-sensor anomaly detection
- Ma X, Tao Z, Wang Y et al (2015) Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transp Res Part C Emerg Technol* 54:187–197. <https://doi.org/10.1016/j.trc.2015.03.014>
- Luo H, Huang M, Zhou Z (2019) A dual-tree complex wavelet enhanced convolutional LSTM neural network for structural health monitoring of automotive suspension. *Measurement* 137:14–27. <https://doi.org/10.1016/j.measurement.2019.01.038>
- Zhou C, Sun C, Liu Z, Lau FCM (2015) A C-LSTM neural network for text classification
- Zhao J, Mao X, Chen L (2019) Speech emotion recognition using deep 1D & 2D CNN LSTM networks. *Biomed Signal Process Control* 47:312–323. <https://doi.org/10.1016/j.bspc.2018.08.035>
- Yu Y, Cai C, Deng L (2017) Vehicle axle identification using wavelet analysis of bridge global responses. *J Vib Control* 23:2830–2840. <https://doi.org/10.1177/1077546315623147>
- Olhede SC, Walden AT (2002) Generalized Morse wavelets. *IEEE Trans Signal Process* 50:2661–2670. <https://doi.org/10.1109/TSP.2002.804066>
- Lilly JM, Olhede SC (2009) Higher-order properties of analytic wavelets. *IEEE Trans Signal Process* 57:146–160. <https://doi.org/10.1109/TSP.2008.2007607>
- Gers FA, Schmidhuber J, Cummins F (2000) Learning to forget: continual prediction with LSTM. *Neural Comput* 12:2451–2471. <https://doi.org/10.1162/089976600300015015>
- Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. In: 3rd Int Conf Learn Represent ICLR 2015—Conf Track Proc, pp 1–15
- Cohen J (1960) A coefficient of agreement for nominal scales. *Educ Psychol Meas* 20:37–46. <https://doi.org/10.1177/001316446002000104>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.