


Article

Tire–Pavement Contact-Aware Weight Estimation for Multi-Sensor WIM Systems

Zhixin Jia ¹, Kaiya Fu ² and Mengxiang Lin ^{2,*}

¹ State Key Laboratory of Software Development Environment, School of Mechanical Engineering & Automation, Beihang University, Beijing 100083, China; jiazx@buaa.edu.cn

² School of Mechanical Engineering & Automation, Beihang University, Beijing 100083, China; sy1607504@buaa.edu.cn

* Correspondence: linmx@buaa.edu.cn; Tel.: +86-10-82339904

Received: 19 March 2019; Accepted: 24 April 2019; Published: 30 April 2019



Abstract: Accurately estimating the weight of a moving vehicle at normal speed remains a challenging problem due to the complex vehicle dynamics and vehicle–pavement interaction. The weighing technique based on multiple sensors has proven to be an effective approach to this task. To improve the accuracy of weigh-in-motion (WIM) systems, this paper proposes a neural network-based method integrating identification and predication. A backpropagation neural network for signal classification (BPNN-i) was designed to identify ideal samples acquired by load sensors closest to the tire-pavement contact area. After that, ideal samples were used to predict the gross vehicle weight by using another backpropagation neural network (BPNN-e). The dataset for training and evaluation was collected from a multiple-sensor WIM (MS-WIM) system deployed in a public road. In our experiments, 96.89% of samples in the test set had an estimation error of less than 5%.

Keywords: multi-sensor weigh-in-motion system; BP neural network; signal identification; vehicle weight estimation

1. Introduction

Overloading has always been a major concern for road traffic [1]. To date, several weighing techniques have been used for overweight vehicle detection [2,3]. Weigh-in-motion (WIM) is a developing technology that aims at obtaining the gross weight of a vehicle travelling at normal speed [4,5]. Compared with static weighing, two major factors should be considered in designing a WIM system. First, sensors in a WIM system measure dynamic load [6,7] rather than the static weight due to vehicle dynamics [8,9]. Second, the complex interaction between vehicles and pavements leads to difficulties in accurately measuring dynamic load [1,10–13].

These problems have motivated multiple-sensor WIM systems (MS-WIMs) in which an array of sensors installed inside the pavements is used to sense dynamic force varying with a certain frequency applied by a travelling vehicle. In the past few decades, the studies on MS-WIM systems have mainly focused on two aspects: sensor layout and vehicle weight estimation algorithms. The former concerns how to arrange sensors to precisely measure the dynamic response along the travelling direction. The goal of the latter is to precisely compute the vehicle weight that could eliminate the errors caused by the vehicle dynamics as much as possible. The most related work presented recently in the literature is reviewed as follows. M.H. Glover et al. [14] arranged nine strip sensors at regular intervals. The static weight of the vehicle is estimated by the mean value, the median value, or the average of the highest and lowest values obtained by the sensors [15]. David Cebon et al. [16] deployed 96 strip WIM sensors in Indiana and averaged the outputs of all the sensors to estimate the static weight. Piotr Burnoset et al. [17] evenly placed 16 piezoelectric load sensors in the form of 4-m-long strips in

Poland. The static weight of a vehicle was computed as a simple average value of the load samples from successive sensors. Wenbin Zhang et al. [18] used five embedded concrete strain sensors, and the static weight was also computed as a simple average value of all the sensors. Sungkon Kim et al. [19] arranged 12 strain sensors in three rows and four columns at equal intervals in Korea and three sensors in one row at equal intervals in the direction perpendicular to the driving. These strain sensors were mounted on the lower surface of the beam of a bridge. An artificial neural network (ANN) was used to fuse the measurements of multiple sensors to estimate the static weight [19–22]. Ryszard Sroka et al. [23] used 16 polymer piezoelectric load sensors, 8 inductive loop sensors, and 8 temperature sensors in their MS-WIM. The sensors were distributed evenly along the site. The static weight was estimated using a method of cooperative fusion, complementary fusion, and attribute fusion.

Unfortunately, the application of multiple-sensor WIM systems to weight enforcement is still limited due to their poor accuracy [4,24,25]. Multiple factors contribute to this problem [26–28]. Research on vehicle and road interaction has revealed that the instantaneous force applied by a moving wheel can be approximated by a Gaussian distribution. The mean value of the Gaussian distribution is approximately linear with the dynamic load applied by the vehicle, and occurs at the centerline of the wheel path [29,30]. Inspired by these studies, we propose a novel gross vehicle weight estimation method for MS-WIM systems. The signals of the sensors closest to the tire—pavement contact area rather than all signals collected by an MS-WIM system are used to compute the gross vehicle weight of a passing vehicle [31]. Specifically, two kinds of backpropagation neural network are designed to classify signals and predict vehicle weight because they allow us to tackle those tasks that are too difficult to solve by analytical methods. The proposed method is implemented for an MS-WIM system deployed on a public road, and the data obtained from real road traffic are used to train and test our identification network and prediction network.

2. Layout of the Sensors

The layout of the sensors of the MS-WIM system we used is shown in Figure 1. A total of 56 load sensors were arranged in an array of 4 rows and 14 columns. Load sensors were customized strain gauges in a half-bridge configuration with a temperature compensation module. Two loop-detection sensors were placed at both ends of the sensor array. When a vehicle passed through the first loop sensor, the system began to acquire the output of each load sensor, and the signal collection process ended after the vehicle passed through the second loop sensor. At the same time, the average speed of the vehicle was computed, since the distance between the loop sensors was known. The output signal of a load sensor was considered as one sample—that is, 56 samples were collected after a vehicle passed. Seven load sensors in a row were grouped and mounted in a steel box. All steel boxes were embedded 30 cm below the road surface.

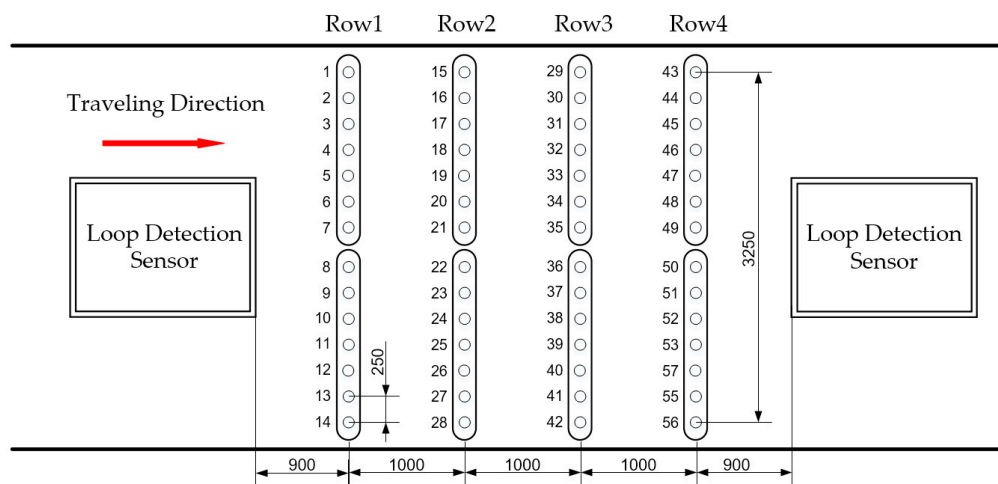


Figure 1. The layout of the sensors.

3. Ideal Sample Identification

Previous studies on the interaction between moving vehicles and roads have revealed that the dynamic load applied by a moving wheel is approximatively Gaussian [29]. As shown in Figure 2a adapted from [29], the strain just below the tire is the mean of the Gaussian distribution and would decay rapidly due to the mechanical properties of roads. In the case of the MS-WIM system, an array of sensors were deployed to sense dynamic load along the travelling direction. However, not all the sensors in the array could properly detect instantaneous forces applied by a moving wheel. Figure 2b shows an example of output signals obtained from our MS-WIM system for a six-axis vehicle. When the wheels passed over the sensors, the closer the sensors were to the tire—pavement contact area, the stronger the output signals were. For those sensors far away from the wheel, the output signals were too weak to be used. We refer to the output signals generated by the former sensors as ideal samples in this work. For our example in Figure 2b, samples generated from sensors 5 and 6 are ideal.

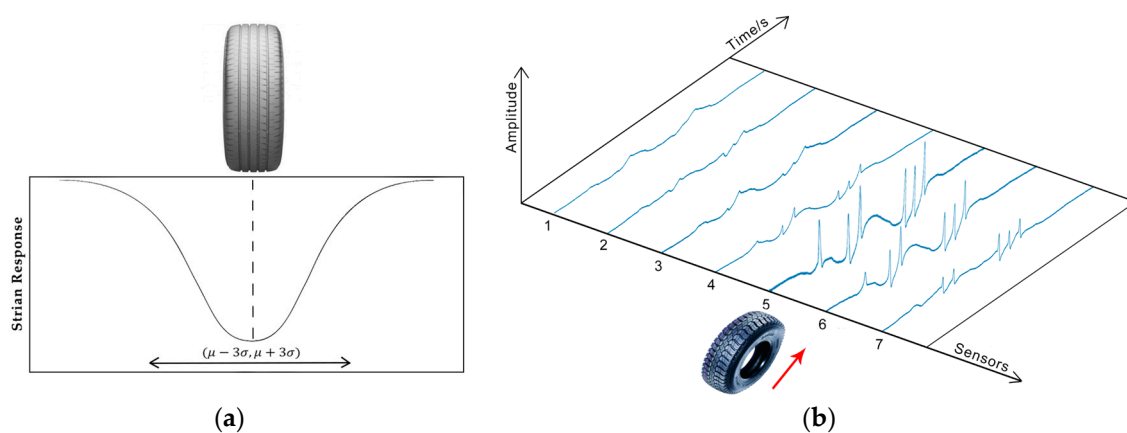


Figure 2. (a) Strain response under a moving wheel adapted from [29]; (b) An example of samples.

The values of ideal samples are measurements of tire—pavement contact stress. Most prior work has used ideal samples implicitly to estimate the vehicle/axis static weight by controlling the trajectory of vehicles [16,29]. However, in reality, we are even unable to know the exact locations of the wheels ahead of time for a vehicle travelling on the road. Therefore, a general solution to identify ideal samples is necessary.

3.1. Data Preprocessing

The signals that were originally acquired needed to be appropriately processed to facilitate subsequent processing. The preprocess proceeded in two steps: normalization and transformation. Normalization is related to the magnitude of signals, which depends on the dynamic force applied and the characteristics of sensors. To eliminate possible biases, the baseline of the original signal was corrected, and the relative magnitude of signals was derived as per Equation (1):

$$A = (A - A_{min}) / (A_{max} - A_{min}), \quad (1)$$

where:

A —the value of the data point to be normalized;

A_{min} —the minimum value of all data points in the signal; and

A_{max} —the maximum value of all data points in the signal.

Figure 3 shows an original signal and its results after normalization. All the data points' values were between 0 and 1, and the relative magnitude and trend of the data points' values were not changed after normalization, but the differences in the peak sizes of different signals were eliminated.

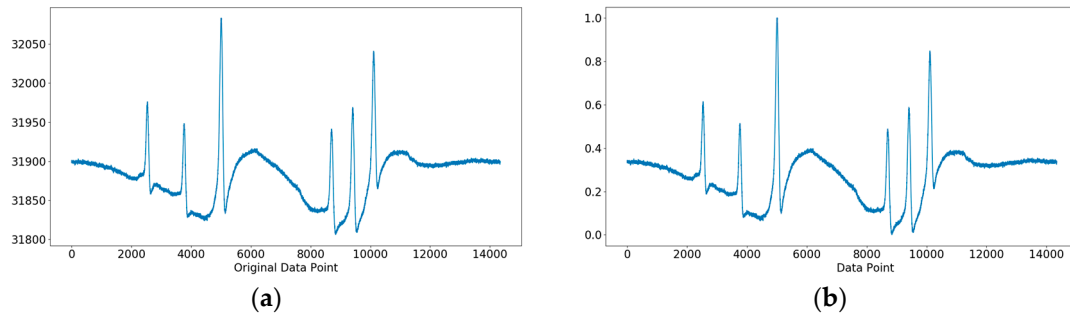


Figure 3. (a) The original signal; (b) The signal after normalization.

Moreover, the sampling frequency of an MS-WIM system is usually fixed. This means that the number of data points in the signals varies with vehicle speed. Therefore, the data needed to be transformed so that all signals had the same length. To reduce the impact of the transformation on signals, the target length of a signal was set as the median of the lengths of all signals. In the case that the length of the signal was greater than the target length, the signal was sparsely sampled at a constant interval. The interval is given in Equation (2):

$$I = \lfloor L_{ori} / (L_{ori} - L_{tar}) \rfloor \quad (2)$$

where:

I —the interval;

L_{ori} —the original length of the signal;

L_{tar} —the target length of the signal; and

$\lfloor \rfloor$ —the round down operation.

Otherwise, the signal was interpolated by inserting a data point at a constant interval. The interval is given in Equation (3), and the value of the data point to be interpolated is given in Equation (4):

$$I = \lfloor L_{ori} / (L_{tar} - L_{ori}) \rfloor \quad (3)$$

$$A_{pos} = (A_{pos-1} + A_{pos+1}) / 2, \quad (4)$$

where:

A_{pos} —the value of the data point to be interpolated;

A_{pos-1} —the value of the previous data point of the data point to be interpolated; and

A_{pos+1} —the value of the next data point of the data point to be interpolated.

Figure 4 shows the samples before and after signal transformation, in which the target length of a signal is 12,200.

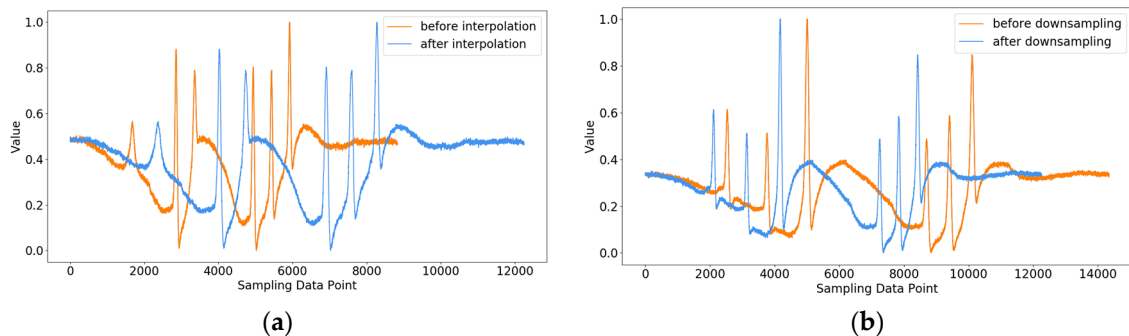


Figure 4. (a) The signal before/after interpolation; (b) The signals before/after down-sampling.

3.2. Short-Time Fourier Transform

The signal generated by a load sensor is a kind of non-stationary time-domain signal [12]. The noise exists during the whole process of data collection, and the effective signal occurs only when the wheel passes over the sensor. In order to obtain the time-domain and frequency-domain characteristics of the effective signal, the short-time Fourier transform was used to extract features for classifying signals.

The basic idea of the short-time Fourier transform is to divide a signal into a number of segments and perform a Fourier analysis segment-by-segment to obtain a local spectrum diagram in different time segments by a window function with an appropriate width to continuously move along the time axis. The definition of the short-time Fourier transform is given in Equation (5):

$$X(\tau, \omega) = \int_{-\infty}^{\infty} x(t) \omega(t - \tau) e^{-j\omega t} dt, \quad (5)$$

where:

$x(t)$ —the time-varying signal, that is, the signal to be transformed;

t —the time variable;

ω —the angular frequency;

$\omega(t)$ —the window function;

τ —the window time position of window function; and

$X(\tau, \omega)$ —the time-frequency function, which reflects the spectral amplitude of the component in which the frequency is ω of $x(t)$ at time t .

For the short-time Fourier transform, the selection of the window function is important because the result of the short-time Fourier transform is mainly determined by the width and shape of the window function. According to the characteristics of the signals, Hamming Window was selected as the window function, as defined in Equation (6):

$$\omega(n, \alpha) = (1 - \alpha) - \alpha \times \cos \frac{2\pi n}{N - 1}, \quad (6)$$

where:

α —the scale factor, which is generally 0.46 [32,33];

n —the width of the window function, and $0 \leq n \leq N - 1$.

After the short-time Fourier transform, the features of signals, i.e., the local spectrograms of each signal in different time periods, were obtained, which would further be fed into a neural network for classification.

3.3. Signal Classification

A three-layer forward neural network (BPNN-i) was designed to classify the signals. As shown in Figure 5, x_i ($1 \leq i \leq m$) are the input features of the BPNN-i, b_{1j} ($1 \leq j \leq n$) is the bias values of neurons in the hidden layer, b_{21} and b_{22} are the bias values of the output neurons, w_{mn} is the weights between the input layer and the hidden layer, and w_{n2} is the weights between the hidden layer and the output layer. The number of neurons in the hidden layer was set to 512. The outputs of the output layer neurons were the probability that the sample belongs to each category. The number of neurons in the output layer was the number of categories, that is, two, where one represents an ideal sample and the other is not.

The number of neurons in the input layer of the BPNN-i depended on the dimension of the spectral amplitude of a signal. However, the dimension of the spectral amplitude obtained by the short-time Fourier transform was high due to the high sampling frequency of the MS-WIM system.

Principal component analysis (PCA) was used first to reduce the dimension of the signal spectral amplitude into a few irrelevant integrated components.

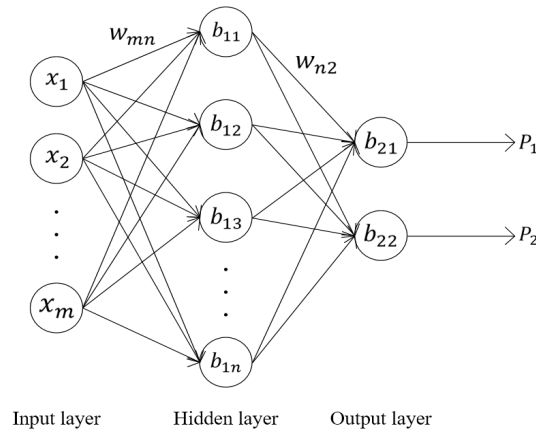


Figure 5. The structure of backpropagation neural network for identification (BPNN-i).

The activation function of neurons in the hidden layer was set as a sigmoid function, defined as Equation (7):

$$f_1(x) = \frac{1}{1 + e^{-x}}. \quad (7)$$

So, the output of the neurons in the hidden layer was:

$$H = f_1(W_1^T X + B_1), \quad (8)$$

where:

$H = [h_1, h_2, \dots, h_n]$ —the output of the hidden layer;

$X = [x_1, x_2, \dots, x_m]$ —the input of BPNN-i, that is, the spectral amplitude after PCA;

$W_1 = \begin{bmatrix} w_{11}^1 & \cdots & w_{1n}^1 \\ \vdots & \ddots & \vdots \\ w_{m1}^1 & \cdots & w_{mn}^1 \end{bmatrix}$ —the weights between the input layer and the hidden layer;

$B_1 = [b_{11}, b_{12}, \dots, b_{1n}]$ —the bias values of neurons in the hidden layer.

The activation function of neurons in the output layer was set as a linear function, which is defined as Equation (9):

$$f_2(x) = x. \quad (9)$$

So, the output of BPNN-i is:

$$P = f_2(W_2^T H + B_2), \quad (10)$$

where:

$P = [p_1, p_2]$ —the output of the output layer;

$W_2 = \begin{bmatrix} w_{11}^2 & w_{12}^2 \\ \vdots & \vdots \\ w_{n1}^2 & w_{n2}^2 \end{bmatrix}$ —the weights between the hidden layer and the output layer; and

$B_2 = [b_{21}, b_{22}]$ —the bias values of neurons in the output layer.

4. Gross Vehicle Weight Estimation

So far, ideal samples and information about vehicle axles were obtained after a vehicle passed. Our ultimate goal was to estimate vehicle weight by ideal samples. Like the previous signal identification, we relied on BP networks in which features such as crests were network inputs.

4.1. Data Preprocess

In order to extract crests in a signal accurately, we had to increase the difference between crests and non-crests so as to reduce the effect of noise. To this end, crest sharpening was applied by Equation (11):

$$y' = y - f_1 \frac{d^2 y}{dx^2} + f_2 \frac{d^4 y}{dx^4}, \quad (11)$$

where:

y —the original signal;

y' —the processed signal;

f_1 —the scale factor of the second derivative of the original signal; and

f_2 —the scale factor of the fourth derivative of the original signal.

Because the magnitude of the crest was relevant to the weight of the vehicle, the signals generated by vehicles of different weights were quite different. Z-score normalization was applied to reduce this kind of difference. The result is shown in Figure 6.

$$y' = \frac{y - \bar{y}}{\sigma}, \quad (12)$$

where:

y —the original signal;

y' —the signal after normalization;

\bar{y} —the average of the original signals; and

σ —the standard error of the original signals.

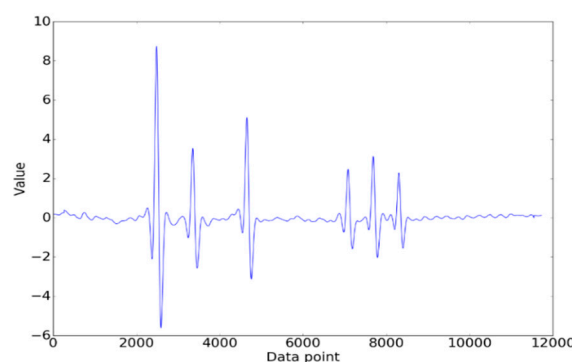


Figure 6. The signal after normalization.

4.2. Extraction of Crest

When the wheel of a vehicle passes over a sensor, contact stress causes an obvious fluctuation in the sensor output. To locate the signal wave crest, a local search algorithm was used. Specifically, we referred to the trough point on the left side of the crest as the *contact point*, while the trough points on the right side of the crest were referred to as the *leaving point*, which indicate the positions where the wheel starts pressing the sensor and leaves the sensor, respectively. The signal between the contact point and the leaving point is the crest to be extracted.

In the ideal situation, the extraction result is shown in Figure 7. That is, the number of crests extracted was equal to the number of axles of the vehicle. However, as shown in Figure 8a, some crests could have been missed due to complicated circumstances during sampling. Crest extraction correction was therefore introduced. Firstly, the number of axles was determined according to the number of crests occurring mostly for all signals collected. The signals whose number of crests extracted was equal to the number of axles were regarded as reference signals, and others were signals to be corrected. Secondly, the missing crests in a signal were further complemented by matching crests with the reference signal. The result after crest matching is shown in Figure 8b.

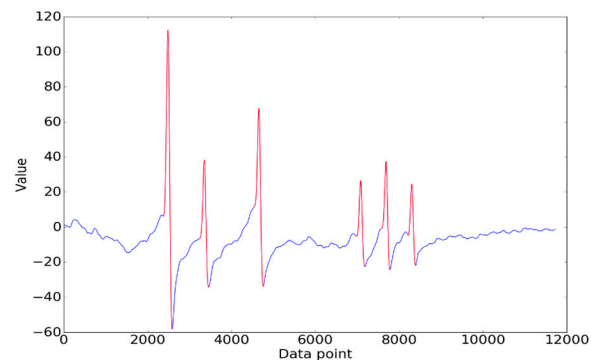


Figure 7. The ideal result of crest extraction.

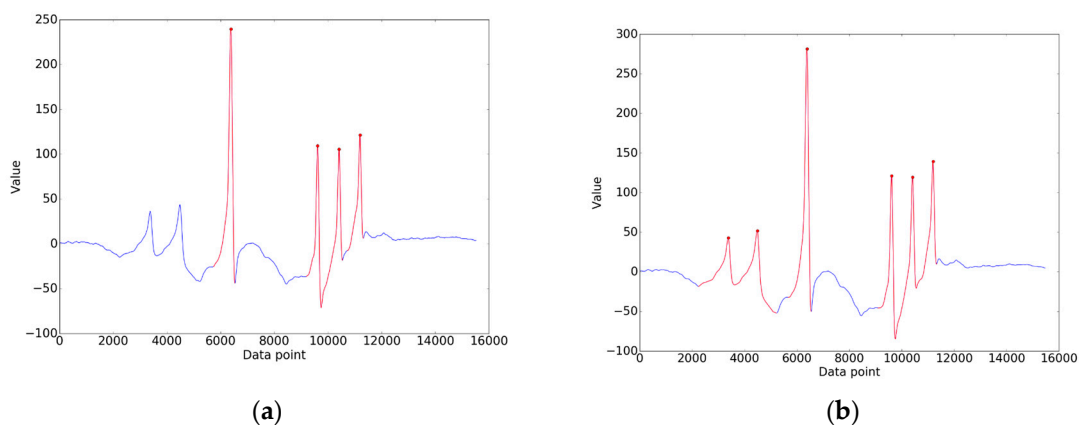


Figure 8. (a) The result with missed crests; (b) The result after crest correction.

4.3. Estimation of Vehicle Weight

A multi-layer forward neural network (BPNN-e) was designed to compute the gross weights of vehicles using data collected by the sensors of each row in the WIM system. Therefore, as shown in Figure 9, there were four BPNN-es corresponding to four rows of sensors, and the weight of a vehicle was the average of the outputs of all BPNN-es.

The structure of a BPNN-e used for weight regression is depicted in Figure 10, containing an input layer, an output layer, and three hidden layers. Specifically, the number of neurons in the output layer was 1, and the number of neurons in the three hidden layers was 32, 16, and 8, respectively. It is worth noting that the dimension of the input of a BPNN-e varied with the number of vehicle axles. Let us take a six-axle vehicle as an example. The six-axle vehicle would generate six crests on the signal, so the inputs of the BPNN-e would include 2×6 areas of each crest, 2×5 distances between crests, and one vehicle speed (i.e., the dimension of the input is 23).

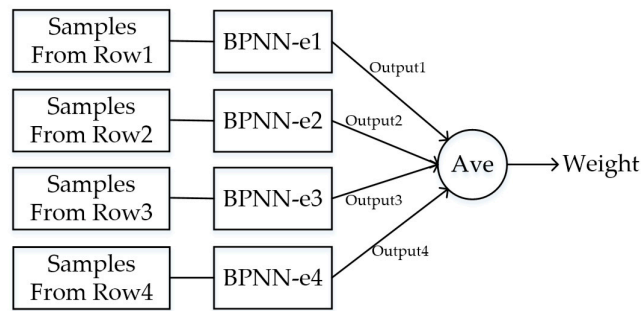


Figure 9. Weight calculation on the neural networks for estimation (BPNN-e).

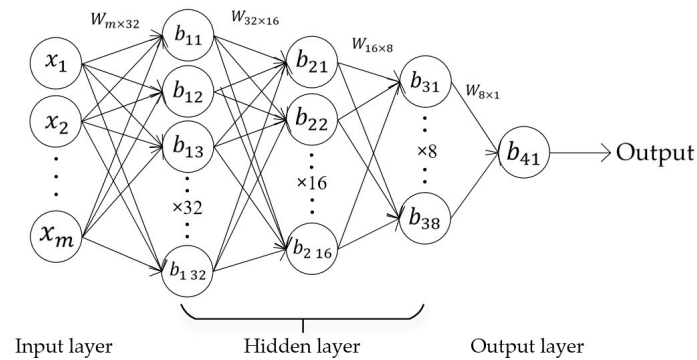


Figure 10. The structure of a BPNN-e.

5. Implementation and Evaluation

The MS-WIM system with the layout shown in Figure 1 was deployed on a public road. Meanwhile, a static weighing system was equipped on the same road so that the real weight of a vehicle was available. To evaluate our method, we used data collected from 28 August 2017 to 13 September 2017 as our dataset, which includes 324,408 samples generated by 5793 vehicles that passed. It is worth noting that samples collected by several failed sensors were deleted. As mentioned before, the average speed of a passing vehicle was computed by utilizing two loop sensors. The range of vehicle weight in our dataset is 5000–63,951 kg, and the speed range is 6.1–57.9 km/h. The distributions of all the samples over weight and speed are shown in Figure 11.

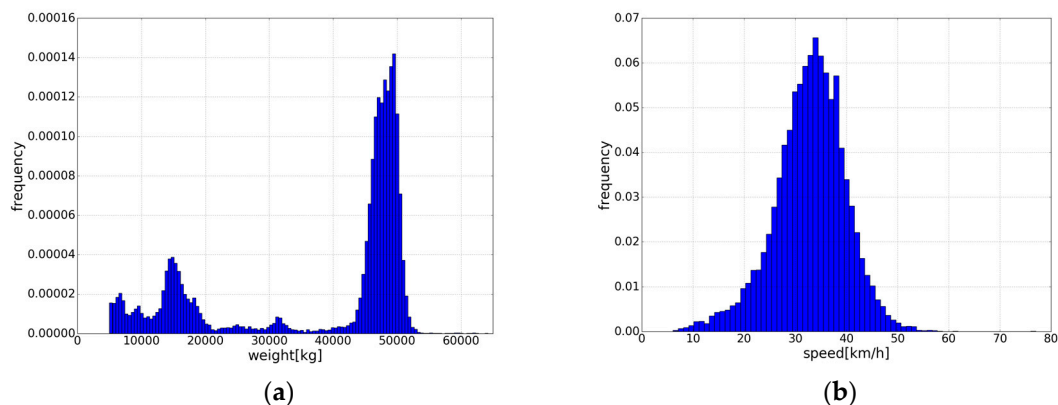


Figure 11. (a) The sample distribution over weights; (b) The sample distribution over speeds.

Furthermore, we implemented our method with the help of some tools. In particular, the short-time Fourier transform algorithm *stft* in *scipy* and the PCA algorithm in *scikit-learn* were used in our implementation. The width of the window function was set as 100, that is, each time 100 data points were selected as a small segment for Fourier transform. The dimension of the spectrum amplitude

obtained by the short-time Fourier transform of the scaled signal was 12,597. The standard of the number of principal components to select was that the contribution of the cumulative variance of the principal component selected was up to 99%. After PCA dimensionality reduction, the dimension of spectrum amplitude was reduced from 12,597 to 8602 (i.e., the dimension decreased by 46.44%).

Both BPNN-i and BPNN-e were built on the top of *Tensorflow*. The number of neurons in the input layer was set to 8602 for BPNN-i and to 23 for BPNN-e. The weights of BPNN-i and BPNN-e were randomly initialized with random values satisfying a normal distribution, in which the mean was 0 and the variance was $\sqrt{2/n}$ (n is the number of the training data). All the bias values were initialized to zero. For the network configuration of BPNN-i, a small batch gradient descent method was used for training in which the batch size was 100 and the loss function was cross entropy. The activation function was sigmoid. For BPNN-e, the loss function was MSE (mean squared error), and Adaptive Moment Estimation (Adam) was chosen as the optimization algorithm. The activation function of BPNN-e is Rectified Linear Units (RELU). In addition, in order to prevent the neural network from overfitting, L2 regularization was added during the training for both BPNNs.

BPNN-i and BPNN-e were trained separately to achieve their respective goals. All experiments were carried out on a PC with i7-7500 and 8 G memory (Lenovo, Beijing, China). First, the BPNN-i was trained to select ideal samples. Eighty percent of samples in the dataset (i.e., 259,526 samples) were randomly selected as the training set; the remaining were used as the testing set. The samples in the dataset were manually labeled as ideal or not. The accuracy of identification was the ratio of the number of the samples with correct classification to the total number of the samples. The changes of identification accuracy in the training set and the testing set are shown in Figure 12. Eventually, the identification accuracy in the testing set was 92.04%.

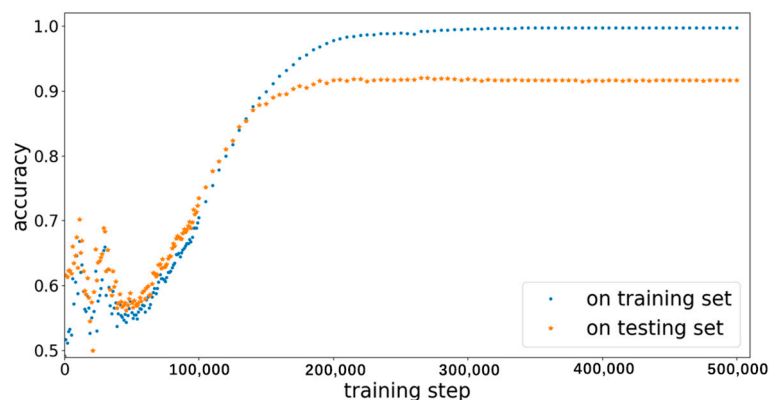


Figure 12. The changes of classification accuracy in the training and testing sets.

To train and test the prediction network BPNN-e, 55,000 ideal samples were used, which were divided into a training set and a testing set according to a ratio of 4:1. To measure the accuracy of estimation in the testing set, the relative error defined by Equation (13) [9] was used:

$$E = \frac{|W_r - W_e|}{W_r} \times 100\%, \quad (13)$$

where:

E is the relative error;

W_e is the gross vehicle weight predicated by the trained BPNN-e; and

W_r is the real vehicle weight computed by the static weighing system.

Overall, the relative error of 96.9% samples in the testing set was less than 5%. For each vehicle, the total time spent engaged in the data pre-process, identification, and predication was less than 3 s on average. We further investigated the relationship between weight, speed, and accuracy. The accuracies

of estimation for different vehicle weights and speeds are shown in Tables 1 and 2 respectively. As seen in Table 1, larger errors occurred at vehicle weights between 20,000–40,000 kg, while there were fewer training samples in this interval. However, as shown in Table 2, the average relative error had no obvious relationship with the number of training samples, and only slightly increased as the vehicle speed increased. The result indicates that more training data with different weights are needed to improve accuracy further.

Table 1. Average relative errors with different weights.

| Gross Vehicle Weight (t) | 10–20 | 20–30 | 30–40 | 40–50 | 50–60 |
|----------------------------|-------|-------|-------|--------|-------|
| Number of Training Samples | 5850 | 447 | 595 | 35,004 | 2104 |
| Average Relative Error | 2.02% | 4.68% | 6.37% | 1.19% | 2.87% |

Table 2. Average relative errors at different speeds.

| Speed (km/h) | 0–10 | 10–20 | 20–30 | 30–40 | 40–50 | 50–60 |
|----------------------------|-------|-------|-------|--------|--------|-------|
| Number of Training Samples | 54 | 1213 | 7858 | 23,932 | 10,225 | 718 |
| Average Relative Error | 1.68% | 1.22% | 1.45% | 1.48% | 1.47% | 1.64% |

6. Conclusions

In this paper, we presented a neural-network-based method to estimate gross vehicle weight for an MS-WIM system. The main contribution of this work is the integration of ideal sample identification into weight estimation. Experiments were conducted on particular populations of data obtained from real road traffic, and our results demonstrate the overall effectiveness of our method. For an MS-WIM system with sensors embedded under the pavement, the change of pavement characteristics will inevitably affect dynamic loads measured by sensors, which could subsequently increase estimation bias. To this end, both identification and predication networks need to be regularly retrained on new data so as to respond to the changes.

Like other learning-based methods, the accuracy performance of our method heavily relied on training data. Further evaluation will require more comprehensive data with larger variations in weight and speed. In addition, a new implementation that requires much less training data and transferring a trained network to a new MS-WIM system are also worthy of further study.

Author Contributions: Conceptualization, M.L.; Methodology, Z.J., K.F. and M.L.; Software, Z.J. and K.F.; Data preparation, K.F.; Writing—Original Draft Preparation, Z.J., K.F., and M.L.

Funding: This research is supported by SKLSDE-2019ZX-03.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Jacob, B.; La Beaumelle, V.F.-D. Improving truck safety: Potential of weigh-in-motion technology. *IATSS Res.* **2010**, *34*, 9–15. [\[CrossRef\]](#)
- Jacob, B.; Cottineau, L.-M. Weigh-in-motion for Direct Enforcement of Overloaded Commercial Vehicles. *Transp. Res. Procedia* **2016**, *14*, 1413–1422. [\[CrossRef\]](#)
- Fitzgerald, P.C.; Sevillano, E.; Obrien, E.J.; Malekjafarian, A. Bridge weigh-in-motion using a moving force identification algorithm. *Procedia Eng.* **2017**, *199*, 2955–2960. [\[CrossRef\]](#)
- Oubrich, L.; Ouassaid, M.; Maaroufi, M. Dynamic loads, source of errors of high speed weigh in motion systems. In Proceedings of the 2017 14th International Multi-Conference on Systems, Signals & Devices (SSD), Marrakech, Morocco, 28–31 March 2017; pp. 354–359.
- Oubrich, L.; Ouassaid, M.; Maaroufi, M. Reconstruction of axle load signal, measurement basis of static load of vehicle axles through the high speed weigh in motion system. In Proceedings of the 2018 4th International Conference on Optimization and Applications (ICOA), Mohammedia, Morocco, 26–27 April 2018.

6. Highway Innovative Technology Evaluation Center (HITEC). *Evaluation of Measurement Specialties, Inc. Piezoelectric Weigh-In-Motion Sensors*; Report No. CERF No. 40587; American Society of Civil Engineers: Reston, VA, USA, 2001.
7. Burnos, P.; Gajda, J.; Yang, S.X. Thermal Property Analysis of Axle Load Sensors for Weighing Vehicles in Weigh-in-Motion System. *Sensors* **2016**, *16*, 2143. [[CrossRef](#)] [[PubMed](#)]
8. Al-Qadi, I.; Wang, H.; Ouyang, Y.; Grimmelsman, K.; Purdy, J.E. *LTBP Program's Literature Review on Weigh-In-Motion Systems*; Office of Infrastructure Research and Development Federal Highway Administration: McLean, VA, USA, 2016.
9. Qin, T.; Lin, M.; Cao, M.; Fu, K.; Ding, R. Effects of Sensor Location on Dynamic Load Estimation in Weigh-in-Motion System. *Sensors* **2018**, *18*, 3044. [[CrossRef](#)] [[PubMed](#)]
10. Gonzalez, A.; Green, M.F.; O'Brien, E.J.; Xie, H. Theoretical testing of a multiple-sensor bridge weigh-in-motion algorithm. In Proceedings of the 2002 7th International Symposium on Heavy Vehicle Weights and Dimensions, Delft, The Netherlands, 16–20 June 2002.
11. Park, D.W.; Papagiannakis, A.T.; Kim, I.T. Analysis of Dynamic Vehicle Loads using Vehicle Pavement Interaction Model. *KSCE J. Civ. Eng.* **2014**, *18*, 2085–2092. [[CrossRef](#)]
12. Sun, L. An overview of a unified theory of dynamics of vehicle–pavement interaction under moving and stochastic load. *J. Mod. Transp.* **2013**, *21*, 135–162. [[CrossRef](#)]
13. Jacob, B. (Ed.) *Weigh-In-Motion of Axles and Vehicles for Europe (WAVE)*; General Report of 4th Framework Programme Transport; Laboratoire Central des Pontes et Chaussées: Paris, France, 2001.
14. Glover, M.H.; Newton, W.H. *Evaluation of a Multiple-Sensor Weigh-In-Motion System*; Working Paper WP VED/90/69; Transport and Road Research Laboratory: Crowthorne, Berkshire, UK, 1989.
15. Chen, N.; Li, Q.; Li, F.; Jia, Z. A Data Processing Algorithm Based on Vehicle Weigh-in-motion Systems. In Proceedings of the 2013 Ninth International Conference on Natural Computation (ICNC), Shenyang, China, 23–25 July 2013.
16. Cebon, D.; Winkler, C.B. *Multiple-Sensor Weigh-in-Motion: Theory and Experiments*; TRB: Washington, DC, USA, 1991; pp. 70–78.
17. Burnos, P.; Gajda, J.; Piwowar, P.; Sroka, R.; Stencel, M. Accurate weighing of moving vehicles. *Metrol. Meas. Syst.* **2007**, *14*, 507–516.
18. Zhang, W.; Suo, C.; Wang, Q. A Novel Sensor System for Measuring Wheel Loads of Vehicles on Highways. *Sensors* **2008**, *8*, 7671–7689. [[CrossRef](#)] [[PubMed](#)]
19. Kim, S.; Lee, J.; Park, M.-S.; Jo, B.-W. Vehicle Signal Analysis Using Artificial Neural Networks for a Bridge Weigh-in-Motion System. *Sensors* **2009**, *9*, 7943–7956. [[CrossRef](#)] [[PubMed](#)]
20. Mangeas, M.; Glaser, S.; Dolcemasclo, V. Neural networks estimation of truck static weights by fusing weight-in-motion data. In Proceedings of the Fifth International Conference on Information Fusion, Annapolis, MD, USA, 8–11 July 2002; pp. 456–462.
21. Ruder, S. An overview of gradient descent optimization algorithms. *arXiv* **2016**, arXiv:1609.04747.
22. Kinga, D.; Adam, J.B. A method for stochastic optimization. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
23. Sroka, R.; Gajda, J.; Burnos, P.; Piwowar, P. Information fusion in weigh in motion systems. In Proceedings of the 2015 IEEE Sensors Applications Symposium (SAS), Zadar, Croatia, 13–15 April 2015; pp. 1–6.
24. Haugen, T.; Levy, J.R.; Aakre, E.; Tello, M.E.P. Weigh-in-Motion Equipment—Experiences and Challenges. *Transp. Res. Procedia* **2016**, *14*, 1423–1432. [[CrossRef](#)]
25. Gajda, J.; Sroka, R.; Stencel, M.; Zeglen, T.; Piwowar, P.; Burnos, P.; Marszalek, Z. Design and accuracy assessment of the multi-sensor weigh-in-motion system. In Proceedings of the IEEE International Instrumentation and Measurement Technology Conference, Pisa, Italy, 11–14 May 2015; pp. 1036–1041.
26. Scheuter, F. Evaluation of factors affecting WIM system accuracy. In Proceedings of the Second European Conference on COST, Lisbon, Portugal, 14–16 September 1998; Volume 323, pp. 14–16.
27. Pedersen, T.; Haugen, T. Vehicle weight estimation based on piezoelectric sensors used at traffic enforcement cameras experiences from the norwegian system. In Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, 16–19 October 2017.

28. Grakovski, A.; Pilipovecs, A. Multi-purpose fibre optic system for automated vehicle's weighting-in-motion and classification in applications of intelligent transport systems. In Proceedings of the 2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), Naples, Italy, 26–28 June 2017.
29. Xue, W.; Wang, L.; Wang, D. A Prototype Integrated Monitoring System for Pavement and Traffic Based on an Embedded Sensing Network. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 1380–1390. [[CrossRef](#)]
30. Arraigada, M.; Partl, M.N.; Angelone, S.M.; Martinez, F. Evaluation of accelerometers to determine pavement deflections under traffic loads. *Mater. Struct.* **2009**, *42*, 779–790. [[CrossRef](#)]
31. Jia, S.; Li, Q.; Li, X. Signal acquisition and processing of the moving vehicle weighing system. *WSEAS Trans. Signal Process.* **2010**, *3*, 113–122.
32. Harris, F.J. On the use of Windows for Harmonic Analysis with the Discrete Fourier Transform. *Proc. IEEE* **1978**, *66*, 51–83. [[CrossRef](#)]
33. Nuttall, A.H. Some Windows with Very Good Sidelobe Behavior. *IEEE Trans. Acoust. Speech Signal Process.* **1981**, *29*, 84–91. [[CrossRef](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).