sensors

MDPI

Article

Weigh-in-Motion System Based on an Improved Kalman and LSTM-Attention Algorithm

Baidi Shi [1], Yongfeng Jiang [1,2,*], Yefeng Bao [1], Bingyan Chen [1], Ke Yang [1] and Xianming Chen [1]

[1] College of Mechanical & Electrical Engineering, Hohai University, Changzhou 213022, China
[2] Jiangsu Province Wind Power Structural Research Center, Nanjing 210003, China
* Correspondence: jiangyf@hhuc.edu.cn; Tel.: +86-181-5127-0256

Abstract: A weigh-in-motion (WIM) system continuously and automatically detects an object's weight during transmission. The WIM system is used widely in logistics and industry due to increasing labor and time costs. However, the accuracy and stability of WIM system measurements could be affected by shock and vibration under high speed and heavy load. A novel six degrees-of-freedom (DOF), mass–spring damping-based Kalman filter with time scale (KFTS) algorithm was proposed to filter noise due to the multiple-input noise and its frequency that is highly coupled with the basic sensor signal. Additionally, an attention-based long short-term memory (LSTM) model was built to predict the object's mass by using multiple time-series sensor signals. The results showed that the model has superior performance compared to support vector machine (SVM), fully connected network (FCN) and extreme gradient boosting (XGBoost) models. Experiments showed this improved deep learning model can provide remarkable accuracy under different loads, speed and working situations, which can be applied to the high-precision logistics industry.

Keywords: weigh-in-motion; deep learning; Kalman filter; time-series analysis

Citation: Shi, B.; Jiang, Y.; Bao, Y.; Chen, B.; Yang, K.; Chen, X. Weigh-in-Motion System Based on an Improved Kalman and LSTM-Attention Algorithm. *Sensors* **2023**, *23*, 250. https://doi.org/10.3390/s23010250

Academic Editors: Eric M. Lui and Antonio Concilio

Received: 18 November 2022
Revised: 16 December 2022
Accepted: 19 December 2022
Published: 26 December 2022

1. Introduction

Weigh-in-motion (WIM) balance is of great interest in logistical sorting, which detects the cargo weight during the transport link, with an increasingly intensified contradiction between logistics demand and labor gap [1–3]. In addition, WIM technology is also widely used in vehicle weighing apparatus, dynamic railway scales, and automatic agricultural weight check with full advances of automation [4–6]. In recent decades, the development of WIM with a weigh data process has been one of the most important topics that have attracted the attention of various industries [7,8].

In general, the WIM system can be described in the following steps: (i) data acquisition by using pressure sensors and signal filter [9,10] and (ii) output a filtered signal by using linear regression (LR), an autoregression model [11,12], a machine learning (ML) algorithm [13,14] or a deep learning (DL) algorithm [15,16].

In the first aspect, sensors are usually located under a measurement table to measure the pressure in motion, and the data are sampled by pressure sensors. The signal is mainly sampled by pressure signal; however, its accuracy is affected due to the inevitable noise from motor vibration, measurement error, environmental factors [17], etc. Several achievements have been obtained for filtering noise in motion. A fuzzy logic estimator was used to filter noise during dynamic weighing by allocating a suitable weight to each sampling signal [18]. This method shows good performance during low speed. However, as the speed increases, the noise from the motor increases nonlinearly; thus, fuzzy logic LR models show poor performance under high speed and heavy load conditions. Autoregression (AR) models are used in WIM systems to mitigate this effect. Compared with the LR model, filtered data are decided by the current time and former signal; the filter can extract the feature from the time series. In addition, a system identification method is widely used

in WIM systems [19] by using building a spring damping system. The filtered signal is partially decided by estimating the system identification equation and codetermined via the probability distributions of environmental noise. A Kalman filter (KF), as the typical system identification filter, is widely used in the WIM area [20–22]. The filtered signal is codetermined using a system-state matrix and sensor-sampling value at each time. By using a covariance matrix for dynamic state update, the KF has been proven to be a strictly linear optimal filter and is widely used in linear modeling.

The second problem concerns outputting weight by using the filtered signal. The mean value of the sampled signal is generally directly taken as output weight in low-precision industrial fields. However, it is inappropriate to assign the same weigh for each sampling point for the nonstable process of WIM; the lateral and longitudinal acceleration are uneven. The polynomial and exponential models minimize the fitting error by employing the least squares method [23]. It shows better performance than the traditional LR or the averaging model without adding too much operational complexity and is widely used in low-cost embedded devices. Nevertheless, these models are ineffective for high-speed and accurate WIM fields; the sampling time of the pressure signal is less than the system's steady-state time, which makes the signal unable to reflect the transient response of the object's mass precisely due to the nonstatic contact between the measurement tableboard and object [24]. In addition, the sampled pressure signal exhibits nonlinearity as the increasing of transmission speed and measurement weights. Each pressure sensor's sampling frequency is usually larger than 1024 (Hz) during WIM processing. It is difficult to handle bulk and nonlinear data by using a traditional linear model. The model must have characteristics of nonlinearity and time-series processing capability to handle these problems. Currently, a deep learning algorithm centered on the convolution neural network (CNN), FCN and RNN has better performance than machine learning and statistical models in Big Data analysis, computer vision (CV), natural language processing (NLP) and many other fields [25]. FCN [26] and SVM [27] were used to perform data processing for WIM systems. Additionally, a few forefront achievements of deep learning have been reported. The sampled sequences strictly obey causal conditions [28] during the WIM, i.e., the present state is decided by the former situation and only affects subsequent states. LSTM [29] differs from the common regression models (SVM, XGBoost [30] and FCN); as a nonlinear autoregression structure, the causal features of the signal are considered. The measured data in the sampling interval are taken as input and a serial recursive structure is applied, which can tap the time-domain characteristics of the signal without destroying the temporal continuity of the signal sequence. Owing to these properties, RNN models are widely used in stock prediction, traffic-flow forecasting and time-series tasks.

Attention mechanism is a remarkable achievement in deep learning. The performance of RNN can be greatly improved due to the simulation of human attention distribution. In which, the appropriate weight is allocated to each time during extensive data analysis [31–33].

Six DOF dynamic discrete-response models are built and the acceleration response under different load and belt velocity is analyzed in this paper. The corresponding improved KFTS is built by using the dynamic response system and actual sensor signals as the state estimation and measurement matrix. The key-value, attention-based LSTM data processing model is built and finds that the NAdam optimizer is optimal above SGD, RMSprop and Adamax. The measurement error of the SVM, XGBoost, FCN and attention-based LSTM models is compared. Drawbacks and conclusions are summarized at the end.

2. Establishment of the WIM Filter

2.1. Establishment and Analysis of the Dynamic Model

It is difficult to directly measure the accelerated response of pressure sensors due to the transfer table and kinds of limit protection devices. Normally, analytic models approximate the sensor response as equal to the instantaneous stress suffered by the tableboard. A typical WIM balance is shown in Figure 1.

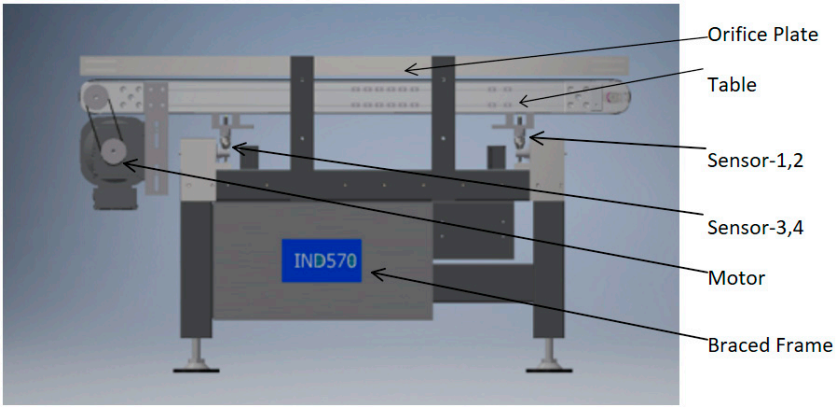


Figure 1. Structure of the WIM scale.

In Figure 1, the main components of WIM scale are shown.

The response of the sensors and the motor's cyclical electric force are used as input. The WIM system as its equivalent model with six degrees of freedom is shown in Figure 2. The system is poweredby one motor, and the measuring module comprises four pressure sensors to detect the instantaneous change in force on the table's vertical direction. Consider the table's steady center of gravity as the origin; the system's generalized coordinate $\boldsymbol{X}$ can be expressed as follows:

$$\boldsymbol{X} = \{x, x_m, x_1, x_2, x_3, x_4\} \tag{1}$$

where $x, x_m, x_1, x_2, x_3, x_4$ are the vertical displacements of table, motor, sensor-1, sensor-2, sensor-3 and sensor-4. Similarly, the velocity matrix and acceleration matrix by can be achieved by discrete differential operation.

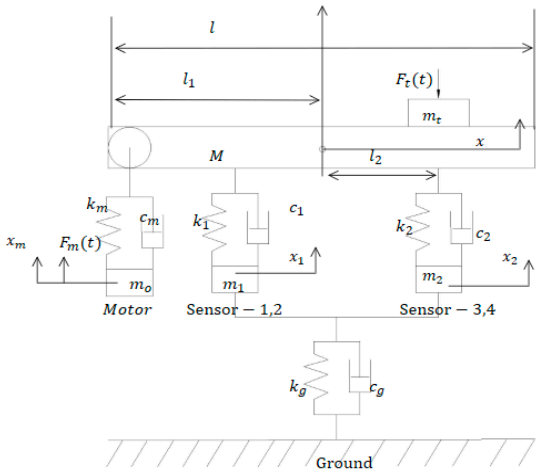


Figure 2. Equivalent model with six degrees of freedom.

In Figure 2, the structures of fixation and connection can be simplified as a spring damping system (c_1, c_2, c_m, c_g, k_1, k_2, k_g, k_m), and each objection's stiffness and damping coefficient have been measured during the design process.

The vertical pressure is not directly affected to avoid damaging the pressure sensor (1~4). The inhibiting devices are installed between the table and sensor, which can be assumed as a spring damping system with a high damping coefficient and low stiffness; m_t is the weight of measurement object. The acceleration of the table in the vertical direction can be measured using an acceleration sensor (sampling frequency is the same as the pressure sensor). The corresponding varying response $F_m(t)$ can be calculated using Newton's second law.

The input force $Fm(t)$ is driven by the motor. Its transient response directly determines the whole system's steady state response as the only power source. The theory electric force of the driving motor in the vertical direction can be defined as follows:

$$F_m(t) = \begin{cases} \frac{T_e}{R} cos\left(\frac{a\pi}{30}t\right) - \frac{i^2\left(L_{min} + \frac{a\pi K}{30}(t-nT)\right)}{\sqrt{b^2\left(\frac{\pi ar}{30}\right)^2\left(\frac{7T}{20} - (t-nT)\right)^2}} \sin\left(\frac{a\pi}{30}t\right), nt \le t \le \left(n + \frac{7}{20}\right)T \\ 0, \qquad\qquad \left(n + \frac{7}{20}\right)Tt \le t \le (n+1)T, (n = 0, 1, 2, \ldots) \end{cases} \quad (2)$$

where R, a, N_r, b, I, K, T_e and L_{min} are inner radius of stator, motor speed, rotor speed, minimum air gap length, winding current, rate of change in winding inductance with respect to position angle, rated torque and winding minimum inductance, respectively.

For the table's shock and vibration in up and down directions, Equation (3) can be achieved based on the model shown in Figure 2.

$$M\ddot{x} = F_0(t) - k_m\,(x - x_m) - 2\,(k_1(x - x_1) - k_2\,(x - x_2) - c_1(\dot{x} - \dot{x}_1) - k_2(\dot{x} - \dot{x}_2)) \quad (3)$$

The motor's shock and vibration in up and down directions can be defined by Equation (4):

$$m_o\ddot{x}_m = F_m(t) - k_m\,(x_m - x) - c_m\,(\dot{x}_m - \dot{x}) \quad (4)$$

The shock and vibration to pressure sensor-1 and pressure sensor-2 in up and down directions can be defined by Equation (5) if the differences in stiffness and damping caused by the assembly are ignored.

$$m_1\ddot{x}_m = -k_1\,(x_1 - x) - c_1\,(\dot{x}_1 - \dot{x}) - k_g\,(x_1 - x) - c_g\,(\dot{x}_1 - \dot{x}) \quad (5)$$

A similar conclusion can be achieved in Equation (6) for sensor-3 and sensor-4:

$$m_2\ddot{x}_2 = -k_2\,(x_2 - x) - c_1\,(\dot{x}_2 - \dot{x}) - k_g\,(x_2 - x) - c_g\,(\dot{x}_2 - \dot{x}) \quad (6)$$

The speed of the belt v (m/min) and weight m (kg) determine the steady-state response of the balance under actual engineering use for the object to be measured. The precise measurement is greatly affected due to object's vibration during its transport. When the object makes contact, the balance table causes self-excited vibration under certain combinations of object weight and speed. This noise, which is a typical high frequency interference noise, greatly disturbs the pressure signal acquisition if it cannot be filtered.

Equations (3)–(6) can be expressed as a vector matrix:

$$\boldsymbol{M}\cdot\ddot{\boldsymbol{x}}(t) + \boldsymbol{C}\cdot\dot{\boldsymbol{x}}(t) + \boldsymbol{K}\cdot\boldsymbol{x}(t) = \boldsymbol{L}\cdot\boldsymbol{U}(t) \quad (7)$$

where $\ddot{\boldsymbol{x}}$, $\dot{\boldsymbol{x}}, \boldsymbol{x}$ are the displacement velocity, accelerated velocity and displacement vectors; $\boldsymbol{M}$, $\boldsymbol{C}$ and $\boldsymbol{K}$ are the matrix of mass, damping and stiffness; $\boldsymbol{L}$ is a constant vector to locate random excitation and $\boldsymbol{U}(t)$ is the nonstationary excitation vector.

The dynamic response, Equation (7), can be written in the form of a state equation:

$$\dot{\boldsymbol{X}}(t) = g(\boldsymbol{X}, \boldsymbol{\theta}, f) = \boldsymbol{A}\cdot\boldsymbol{X}(t) + \boldsymbol{B}\cdot\boldsymbol{F}(t) \quad (8)$$

where $\boldsymbol{\theta}$ is the matrix of structure that contains the information of the system's rigidity and damper. $\boldsymbol{A}$ is a $2n$-by-$2n$ matrix, $\boldsymbol{B}$ is a vector with length $2n$, $\boldsymbol{X}$ is a vector with length $2n$

which contains the vector of acceleration and displacement. These can be described by the following equation:

$$\begin{cases} \boldsymbol{A} = \begin{bmatrix} \mathbf{0} & \boldsymbol{I} \\ -\boldsymbol{M}^{-1}\boldsymbol{K} & -\boldsymbol{M}^{-1}\boldsymbol{C} \end{bmatrix} \\ \boldsymbol{B} = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{M}^{-1} \end{bmatrix} \\ \boldsymbol{F}(t) = \boldsymbol{L}\cdot\boldsymbol{U}(t) \\ \boldsymbol{X}(t) = \begin{bmatrix} \boldsymbol{x}(t) \\ \dot{\boldsymbol{x}}(t) \end{bmatrix} \end{cases} \tag{9}$$

Equation (9) is the continuous state equation; using $e^{-\boldsymbol{A}\boldsymbol{t}}$ to multiply both sides can achieve the following relationship:

$$e^{-\boldsymbol{A}\boldsymbol{t}}(\boldsymbol{X} - \boldsymbol{A}\cdot\boldsymbol{X}) = e^{-\boldsymbol{A}\cdot\boldsymbol{t}}\boldsymbol{B}\cdot\boldsymbol{F}(\boldsymbol{t}) \tag{10}$$

Integrating t over the interval (t_0,t) in Equation (10), and substituting the initial conditions of t_0, then the continuous equation of state solution can be achieved:

$$\boldsymbol{X}(t) = \boldsymbol{\Phi}(t, t_0)\cdot\boldsymbol{X}(t_0) + \int_{t_0}^{t} \boldsymbol{\Phi}(t, \tau)\cdot\boldsymbol{B}\cdot\boldsymbol{F}(\tau)\boldsymbol{d}\tau \tag{11}$$

where $\boldsymbol{\Phi}(t, t_0) = e^{\boldsymbol{A}(t-t_0)}$ state transition matrix, which is used as the state estimation matrix in KF; Equation (11) is the standard solution of a continuous state equation, and its essence is equivalent to the Duhamel integral of a dynamical system.

However, in the WIM's six DOF system, the actual signal is described as a discrete matrix and measured in each discrete sampled time by the pressure and acceleration sensors. Equation (11) needs to be transformed into discrete state equations to obtain the measurement matrix and corresponding covariance matrix:

$$\boldsymbol{X}(t_{k+1}) = \boldsymbol{\Phi}(t_{k+1}, t_k)\cdot\boldsymbol{X}(t_0) + \int_{t_k}^{t_{k+1}} \boldsymbol{\Phi}(t_{k+1}, \tau)\cdot\boldsymbol{B}\cdot\boldsymbol{F}(\tau)\boldsymbol{d}\tau \tag{12}$$

where this equation is a discrete equation and determined by the sensor sampling frequency (1024 Hz), defined in the time interval $\Delta t = t_{k+1} - t_k$. Equation (12) can be rewritten as:

$$\boldsymbol{X}_{\boldsymbol{k+1}} = \boldsymbol{\Phi}_k\boldsymbol{X}_k + \boldsymbol{\Gamma}_k\boldsymbol{F}_k, \boldsymbol{\Gamma}_k = \boldsymbol{B}\int_{\mathbf{0}}^{\Delta t} e^{\boldsymbol{A}\tau}d\tau \tag{13}$$

where $\boldsymbol{X}_k$ and $\boldsymbol{F}_k$ represent the state matrix at time t; $\boldsymbol{\Phi}_k$ represents the state transition matrix from t to $t + 1$ and if the Δt is small enough, then $\boldsymbol{\Phi}_k$ can be calculated approximately by the following Taylor expansion:

$$\boldsymbol{\Phi}_k = e^{\boldsymbol{A}\Delta t} = \boldsymbol{I} + \boldsymbol{A}\Delta t + \frac{\mathbf{1}}{\mathbf{2!}}\boldsymbol{A}^{\mathbf{2}}\Delta t^{\mathbf{2}} + \cdots + \frac{\mathbf{1}}{\boldsymbol{k!}}\boldsymbol{A}^{\boldsymbol{k}}\Delta t^{\boldsymbol{k}} \approx \boldsymbol{I} + \boldsymbol{A}\Delta t \tag{14}$$

where $I = A^0$ is a $2n$-by-$2n$ unit matrix; Equation (14) is the discrete state expression.

The discrete differential equation was solved in the Python 3.8.0 environment using the SymPy and SciPy Toolkits. Set the time step Δt as 1/(1024)s, $\boldsymbol{F}_t(t)$ is the real response-accelerated velocity under different belt speed. Figure 3 shows the accelerated velocity response of four pressure sensors under different weight and belt speeds.

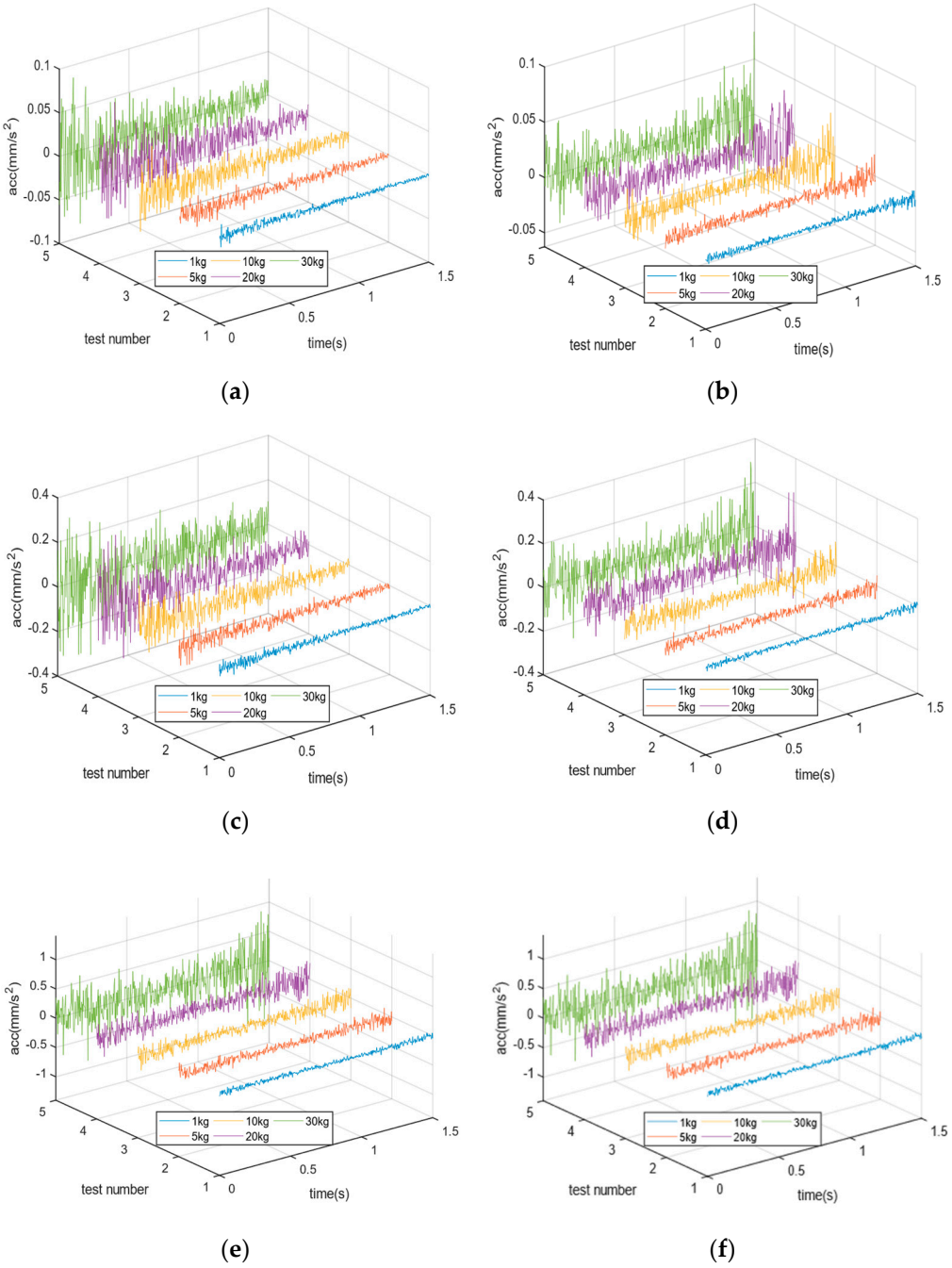


Figure 3. The time-varying accelerated velocity response of pressure sensors under different weight: (**a**) the response of sensor-1 and -2 under v = 45 (m/min), (**b**) the response of sensor-3 and -4 under v = 45 (m/min), (**c**) the response of sensor-1 and -2 under v = 90 (m/min), (**d**) the response of sensor-3 and -4 under v = 90 (m/min), (**e**) the response of sensor-1 and -2 under v = 120 (m/min), (**f**) the response of sensor-3 and -4 under v = 120 (m/min).

In Figure 3, simulation results show:

(1) The valid sampling point is reduced with increasing belt speed. Compared with Figure 3a,b, the signal is mainly influenced by low-frequency noise in low speed. Vibration noise shows more obvious effects, especially under high load due to sensor-1 and sensor-2 being closed to the cargo input side.

(2) The signal indicates a nonlinearity and nonstationary process with increasing belt speed. More seriously, the measuring process is less than the system steady-state time with the decreasing sampling point, as shown in Figure 3c–f.
(3) The pressure sensor is typically oscillatory underdamped; it is crucial to reduce the various internal and external noise from various working conditions. Self-excited vibration is mainly influenced by the genetic frequency, as a high frequency noise, it differs from other signals and can be filtered by a low-pass filter.

The Fourier and Butterworth filters are widely used in WIM systems. These are useful and easy to implement under the certain speed when the cutoff bandwidth is appropriate. However, the responses show different features under different belt speeds and loads, which must be carefully handled to avoid filtering the basic signal. The WIM system based on KF is proposed in the following section.

2.2. Algorithm of KFTS

The filtering steps of the improved KFTS are similar with traditional Kalman: (1) Discrete state equation $\hat{\boldsymbol{X}}_k$ and transition matrix $\boldsymbol{\Phi}_k$ are calculated by the WIM system's estimation matrix. (2) Calculate the Kalman gain $\boldsymbol{K}_k$ according to the actual measurement matrix Z_k, system process noise matrix $\boldsymbol{W}_k$ ~$\boldsymbol{N}$ $(0,Q_k)$ and measurement noise matrix $\boldsymbol{\varepsilon}_k$~$N$ $(0,\boldsymbol{R}_k)$. $\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$ are the corresponding covariance matrices at time t, which are used to describe the environmental and random factor interference.

The filtering process of improved Kalman can be summarized in the following stages:

(1) Prediction: Calculate least-square (LS) state $\hat{\boldsymbol{X}}_k$ based on the state transition matrix $\boldsymbol{\Phi}_{k-1}$ and process noise matrix $\boldsymbol{W}_{k-1}$. The station of $k+1$ can be calculated as follows:

$$\hat{\boldsymbol{X}}_{k+\mathbf{1}} = \boldsymbol{\Phi}_k \cdot \hat{\boldsymbol{X}}_k + \boldsymbol{\Gamma}_k \cdot \boldsymbol{F}_k + \boldsymbol{W}_k \tag{15}$$

where $\hat{\boldsymbol{X}}\ (\boldsymbol{t}_{k-1})$ is the WIM system's state estimation matrix at time t_{k-1}; the state prediction covariance matrix can be described as:

$$\sum_{\hat{\boldsymbol{X}}_k} = \boldsymbol{\Phi}_k \cdot \sum_{\hat{\boldsymbol{x}}_{k-\mathbf{1}}} \cdot \boldsymbol{\Phi}_k{}^T + \boldsymbol{Q}_k / \alpha_{k-1} \tag{16}$$

where $\sum_{\overline{X}_k}$ is the WIM's state-prediction covariance matrix at time t_k; $\sum_{\hat{x}_{k-1}}$ is the WIM's state estimation covariance matrix at time t_{k-1} and α_{k-1} is the adaptive factor at $k-1$.

(2) Measurement: Calculate the error vector $\boldsymbol{i}(t_k)$ based on the pressure sensor's actual signal $\boldsymbol{Z}\ (t_k)$, and $\boldsymbol{i}\ (t_k)$ can be described as:

$$\boldsymbol{i}(t_k) = \boldsymbol{Z}(t_k) - \boldsymbol{J}(t_k) \cdot \overline{\boldsymbol{X}}(t_k) \tag{17}$$

where $\boldsymbol{J}$ is the Jacobian matrix of the measurement signal and can be calculated using a numerical differential.

(3) Calculate Parameter: The theoretical innovation matrix $\boldsymbol{C}_k$, actual innovation matrix $\hat{\boldsymbol{C}}_k$, adaptive factor α_k and Kalman gain $\boldsymbol{K}_k$ can be calculated from the following equations:

$$\hat{\boldsymbol{C}}_k = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{i}_{k-1} \cdot \boldsymbol{i}_{k-1}^T \tag{18}$$

$$\boldsymbol{C}_k = E\left[\boldsymbol{i}_{k-1} \cdot \boldsymbol{i}_{k-1}^T\right] = \boldsymbol{J}_k \boldsymbol{P}_k \boldsymbol{J}_k^T + \boldsymbol{R}_k \tag{19}$$

$$\boldsymbol{\alpha}_k = \frac{c_0}{\|\boldsymbol{i}_k\|} \cdot \left(\frac{\|\hat{\boldsymbol{C}}_k\| / \sqrt{tr(\hat{\boldsymbol{C}}_k)}}{\|\boldsymbol{C}_k\| / \sqrt{tr(\boldsymbol{C}_k)}} \right) \tag{20}$$

where N is the length of time scale, determined by the sampled frequency, belt speed and length of the tableboard, i.e., the number of sampling points; $tr\ (\hat{\boldsymbol{C}}_k)$ is the trace of the innovation matrix. The updated value usually deviates from the actual value

due to the noises that are from model and measurement error. It is necessary to apply the actual innovation matrix $\hat{\boldsymbol{C}}_k$ to codetermine adaptive factor $\boldsymbol{\alpha}_k$. When updating $\boldsymbol{K}_k$, the self-adapting equation is given as Equation (21):

$$\boldsymbol{K}_k = \sum_{\hat{\boldsymbol{X}}_k} \cdot \boldsymbol{J}_k^T \left(\boldsymbol{J}_k \cdot \boldsymbol{P}_k \cdot \boldsymbol{J}_k^T + \alpha_k \cdot \boldsymbol{R}_k \right)^{-1} \tag{21}$$

$\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$ are reversely adjusted matrices to enhance the estimation accuracy. When α_k is a constant value, the KFTS degrades into the traditional extended Kalman filter.

(4) Output: Calculate the filtered signal at time k:

$$\boldsymbol{X}_{k+1} = \hat{\boldsymbol{X}}_{k+\mathbf{1}} + \boldsymbol{K}_k \cdot \left(\boldsymbol{Z}(t_k) - \boldsymbol{J}(t_k) \cdot \overline{\boldsymbol{X}}(t_k) \right) \tag{22}$$

The traditional KF estimate of the state of the linear system is based optimally on the principle of recursive least variance estimation. Unlike the former, the extended Kalman filter extends this algorithm to nonlinear systems. The extended Kalman filter algorithm linearizes the model via the high-order Taylor expansion, which avoids the fitting error.

2.3. Performance Comparison

The system's state matrix is defined by the system's dynamic equation of mechanical structure (Section 2.1) and amended via the sampled signals in our improved KFTS algorithm. Figure 4 shows the performance under the different speeds (60, 90 and 120 m/min):

In Figure 4, our algorithm has been marked with the sign "*". The fluctuation amplitude of the original gradually presents a nonlinear growth trend with the increase in transmission speed under the calibration weight of 20 kg. The acceleration is unavoidable during measurement due to the unevenness of the belt and balance, and the transmission error of the motor. The feature of the peak is used in the EMD filter, but the original state of the signal is excessively filtered and not applicable in the WIM system. The wavelet filter with a different base function shows poor performance under the high-speed state for the same reason. In addition, the effective measurement time(s) is directly affected by the belt speed (v), and the sampling time meets the following formula:

$$s = (l - 2d)/v \tag{23}$$

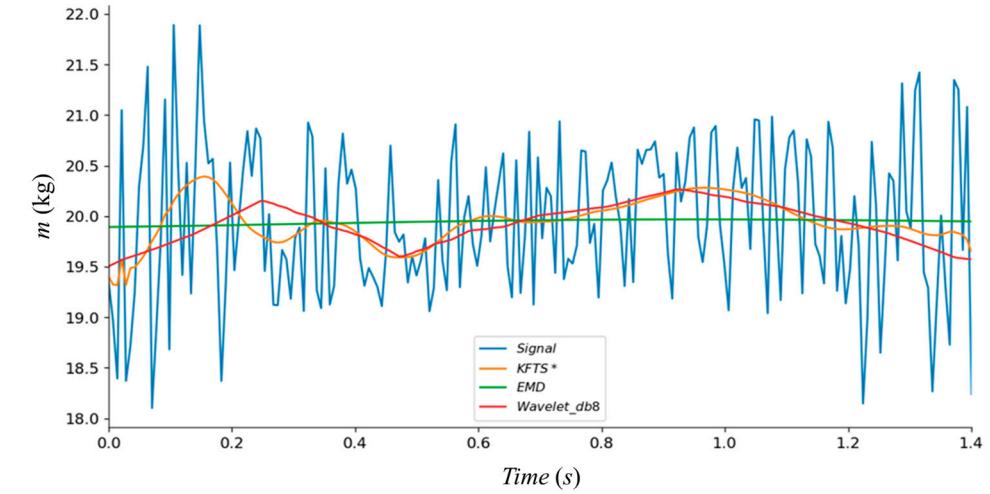
where l is the length of balance and d is the length dimension in the speed direction.

Traditionally, the average value of filter data is taken as the output weight in general industrial area, the corresponding data handling process can be described as the following formula:

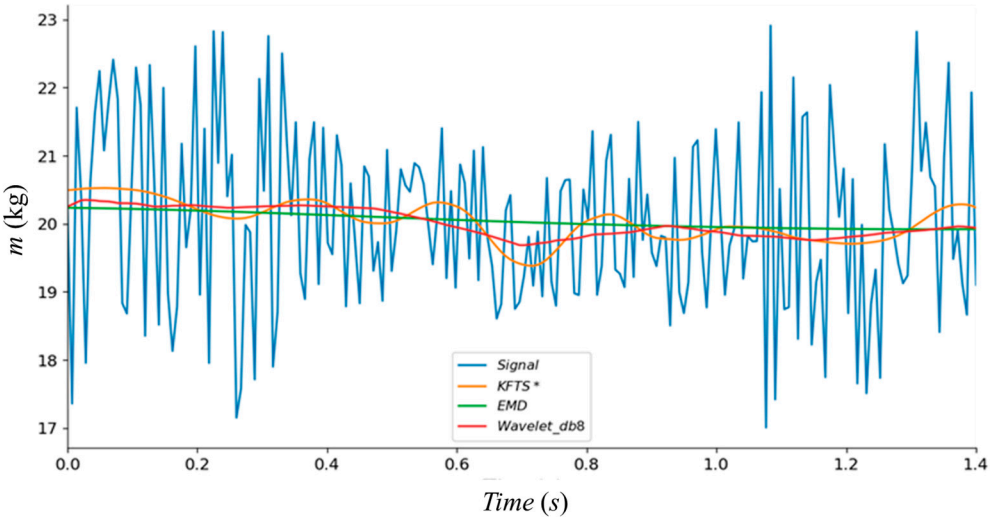
$$out = \sum_{i=1}^{s} \alpha_i m_i \tag{24}$$

where out is the final output weight, α_i is the weight of each sampling point (in the direct averaging method, the value is a constant $1/s$).

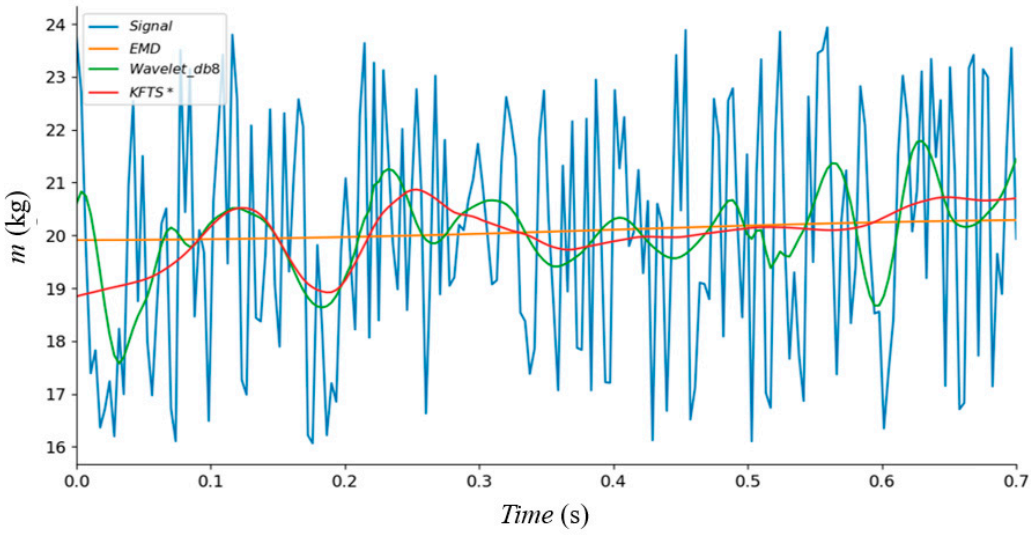
The signal's characteristics under different speeds show nonlinearity, according to Figure 4. Traditional linear algorithm is inappropriate to extract data feature and output exact weight. A deep learning-based model is built to handle the bulk signal and achieve an accurate weigh-in-motion result.



(**a**)



(**b**)



(**c**)

Figure 4. Comparison of filtering effects: (**a**) under the speed of 60 (m/min), (**b**) under the speed of 90 (m/min), (**c**) under the speed of 120 (m/min).

3. Building the Deep Learning Model

3.1. Training Dataset

Deep learning is a typical sample-learning model with feature self-organization. The completeness, adequacy and comprehensiveness of learning samples directly influence model performance. Factors of belt speed (v), load (m) and temperature C(T) are taken in our dataset to comprehensively contain various WIM's working conditions. A three-factor and five-level (L_{35}) orthogonal table was designed and is shown in Table 1.

Table 1. The L_{35} orthogonal test table.

	A—Belt Speed (m/min)	B—Load Weight (kg)	C—Temperature (°C)
$A_1B_1C_1$	45	1	−10
$A_1B_2C_2$	45	5	0
$A_1B_3C_3$	45	10	10
$A_1B_4C_4$	45	20	20
$A_1B_5C_5$	45	30	40
$A_2B_1C_2$	60	1	0
$A_2B_2C_3$	60	5	10
$A_2B_3C_4$	60	10	20
$A_2B_4C_5$	60	20	40
$A_2B_5C_1$	60	30	−10
$A_3B_1C_3$	90	1	10
$A_3B_2C_4$	90	5	20
$A_3B_3C_5$	90	10	40
$A_3B_4C_1$	90	20	−10
$A_3B_5C_2$	90	30	0
$A_4B_1C_4$	120	1	20
$A_4B_2C_5$	120	5	40
$A_4B_3C_1$	120	10	−10
$A_4B_4C_2$	120	20	0
$A_4B_5C_3$	120	30	10

Table 1 contains 25 combinations of the of v, m and T factors. Five hundred tests were conducted under each combination to avoid the randomness of a single measurement. Totally, 10,000 sample data were obtained. The entire experiment was conducted within the China Coal Research Institute to avoid environmental disturbances.

3.2. Residual Connection Module

The convolution operation at position t for a signal series $\boldsymbol{x}$ with length s is defined as follows:

$$y_t = \sum_{k=1}^{m} w_k x_{t-k+1} \tag{25}$$

where m is the length of convolution kernel and w_k is the kernel's value at position k. Additionally, the vector convolution can be defined as follows:

$$\boldsymbol{y} = \{y_i\}_{i=1}^{s-m+1} = \boldsymbol{w}^* \boldsymbol{x} \tag{26}$$

Several studies proved that the depth of the neural network (NN) directly determined the model's feature extraction capability [25]. However, as the model layer is increased, the vanishing gradient problem becomes inevitable. The residual network [34] (ResNet) greatly improved the efficiency of information transmission via adding a short connection to the nonlinear convolution layer. The process is described by Equation (25):

$$h(x) = x + (h(x, \theta) - x) \tag{27}$$

where the forward function is split into two parts: the identity function $\boldsymbol{x}$ and the residue function $h(\boldsymbol{x}) - \boldsymbol{x}$. A nonlinear element composed of a neural network has sufficient ability

to approximate the original objective function or residual function, but the latter is easier to learn in practice, according to the universal approximation theorem [35]. The data flow of the residual connection module is shown in Figure 5.

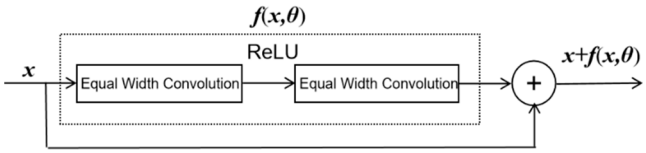


Figure 5. Residual connection module.

In Figure 5, θ is the learning parameter related to the convolution kernel, convolution channel and convolution times. The original characteristics of the signal are maintained in addition to creating constant functions to facilitate model training. Combining Equations (25)–(27), the forward function can be defined as follows:

$$F(\boldsymbol{x}) = \sum_{i=1}^{sc} Relu\left(\sum_{j=1}^{Ni} \boldsymbol{w^{ij}} * \boldsymbol{x} + b^{ij}\right) + \boldsymbol{x} \tag{28}$$

where $\boldsymbol{w^{ij}}$ is the j-th filter's weight in i-th layer b^{ij} represents the bias, and they are both learnable parameters. Ni is the number of kernels in i-th layer.

3.3. Multiscale Feature Extraction

As described in Section 3.2, the data are processed by the residual connection module. The specific scale convolution kernel is used to extract specific features in the convolution network. Three kinds of kernel size are used to enrich the features of the signal, and the number of filters in each layer is 32, 96 and 32. The multiscale convolution layer combined with the residual connection module is shown in Figure 6.

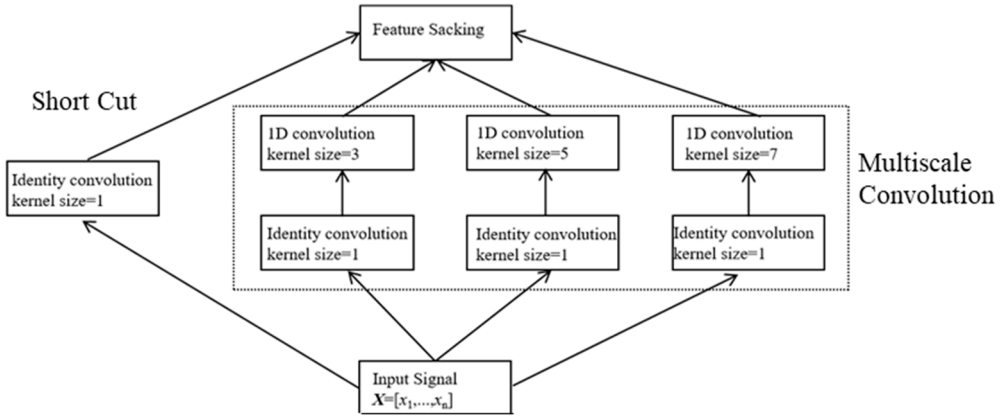


Figure 6. Multiscale convolution with residual connection.

In Figure 6, three convolution kernels with different scales are designed to extract data features. The corresponding features are fused by matrix stacking. It is inappropriate to assign the same weight to each feature during the entire WIM process. The strategy of weight distribution is crucial and directly influences the deep learning model's performance.

3.4. Attention Mechanism Layer

For the linear model, the best weight distribution can be calculated by the least square method, maximum likelihood estimation and other probability estimation methods. The deep learning model is self-organizing and training is based on the error gradient; therefore, the traditional method is not applicable. The attention mechanism as a resource allocation strategy allocates limited computer resources on key features, which is the main method to improve model efficiency and solve information overload. A key-value pair-based attention

layer was designed to allocate suitable weight to each channel and restructure the signal (the input and output have the same shape) to adapt the signal's sequence feature. The channel-based attention module's processing flow can be described as follows:

For the input channel sequence $\boldsymbol{C}$, the output sequence $\boldsymbol{H}$ has different shapes. The corresponding three sequences can be defined as a linear transformation of the input sequence by Equations (29)–(31):

$$\boldsymbol{Q} = \boldsymbol{W_Q X} \in \mathbb{R}^{d_3 \times N} \tag{29}$$

$$\boldsymbol{K} = \boldsymbol{W_K X} \in \mathbb{R}^{d_3 \times N} \tag{30}$$

$$\boldsymbol{V} = \boldsymbol{W_V X} \in \mathbb{R}^{d_2 \times N} \tag{31}$$

where $\boldsymbol{Q}$, $\boldsymbol{K}$ and $\boldsymbol{V}$ denote the query vector sequence, the key vector sequence and the value vector sequence, respectively; d_3 denotes the channel adjustment factor, and the nonlinear fitting capability of the model is enhanced when the value increases.

The output sequence can be calculated by the following equation:

$$\boldsymbol{h}_i = att((\boldsymbol{K}, \boldsymbol{V}), q_i) = \sum_{j=1}^{N} \boldsymbol{\alpha}_{ij} \boldsymbol{v}_j = \sum_{j=1}^{N} softmax\left(s\left(\boldsymbol{k}_j, \boldsymbol{q}_i\right)\right) \boldsymbol{v}_j \tag{32}$$

In addition, the weight vector α can be directly defined and dynamically adjusted according to the error gradient for each iteration via a Fully Connected Layer. By combining the information presented in Sections 3.1–3.4, the signal feature handle path and the shape-change flow of the matrix can be described, as shown in Figure 7.

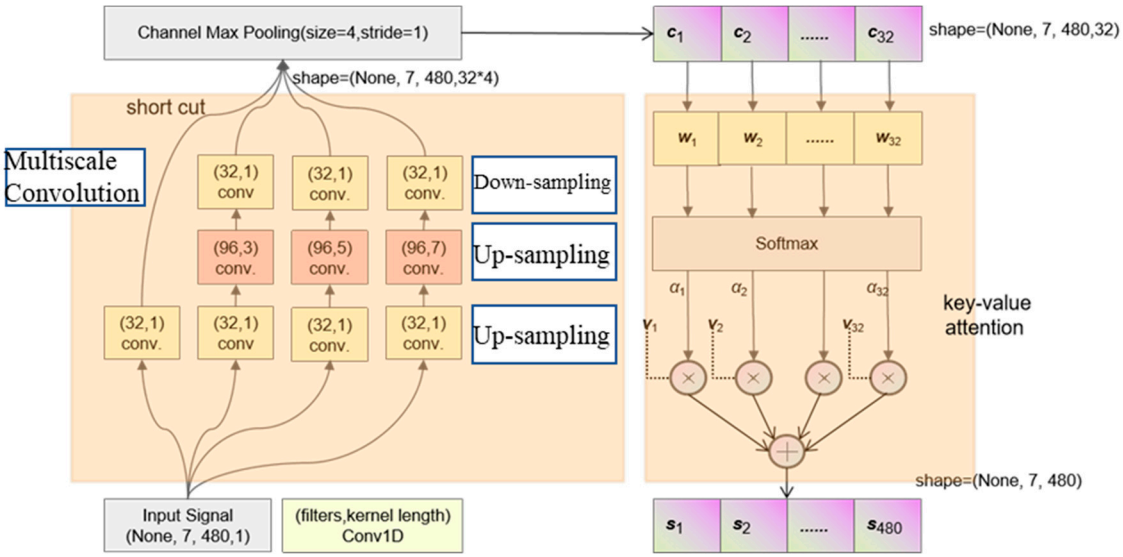


Figure 7. Flow of feature extraction.

In Figure 7, the input signal was processed in two stages:

(1) Four convolution paths were designed in the multiscale convolution module to extract the input signal's time-domain features except for the short-cut path. All paths are up-sampled twice to enrich the receptive field and down-sampled once to adjust the number of channels and deepen the layer depth. The short-cut path is aimed to build the identity mapping between input and the processed value, which is beneficial in the training stage and avoids the degeneration of the network. Additionally, channel max-pooling was used to reduce computational complexity and to smooth local noise.

(2) A key-value pair-based attention layer was designed to allocate suitable weight to each channel and restructure the signal (the input and output have the same shape).

3.5. Long Short-Term Memory (LSTM) Layers

In Sections 3.1–3.4, the attention-based preprocessing filtering module was constructed, and in this section the data processing issues are discussed. This problem is equal to using filtered signals to output an object's mass as accurately as possible. The back propagation (BP) algorithm, which can also be called as FCN, is widely used in weight-in-motion systems. However, as a supervised regression model, it can only build the nonlinear mapping connection with input and output, since time-series signals $\boldsymbol{x}$ with length t tend to exhibit time dependence and time-domain coupling. In this process, features are lost if autoregressive theory is not applied. If the look-back coefficient (LBC) is set to k, the autoregressive model can be described as:

$$y_t = c + \varnothing_1 y_{t-1} + \varnothing_2 y_{t-2} + \cdots + \varnothing_k y_{t-k} + e_t \tag{33}$$

where c is a constant term, $\varnothing_1 \ldots \varnothing_k$ are the parameters of the model and e_t is the vector of white noise.

Similarly, conclusions can be achieved in LSTM; each output y_i ($k \leq I \leq t - k$) is influenced by former output y_{i-k} to y_{i-1} and input x_i to extract the time sequence feature. These features were dropped in FCN, LR and common ML models limited to model structure. LSTM controls cell state using the forgetting gate, input gate and output gate. Sigmoid layer is used to dot the data to complete the addition and deletion of data features in these gate structures. It can effectively solve the problem of long-term dependence of RNN under bulk data: set the current time index as t; the data input is x; the predicted output is $\boldsymbol{H}$; and $\boldsymbol{C}$ is the memory unit. The LSTM structure is shown in Figure 8.

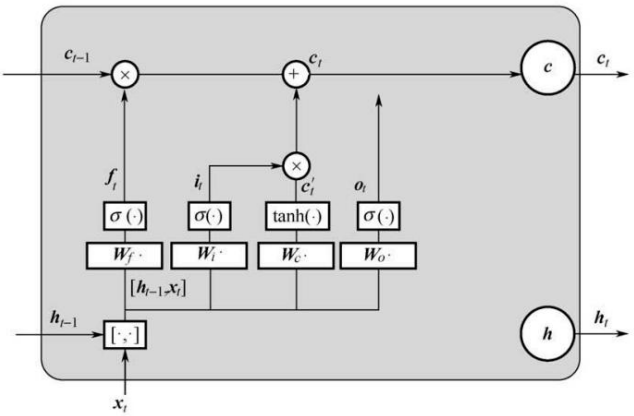


Figure 8. Structure of LSTM.

In Figure 8, σ is the sigmoid activation function; $[\boldsymbol{h}_{t-1}, \boldsymbol{x}_t]$ is the composite matrix of the network's state parameters at $t - 1$ and the input $\boldsymbol{x}_t$ at this round; $\boldsymbol{W}_f$, $\boldsymbol{W}_t$, $\boldsymbol{W}_c$, and $\boldsymbol{W}_o$ are the weight matrices of the forgetting gate, input gate, output gate and state control gate. If the learnable bias is defined as $\boldsymbol{b}$, then these gates can be described by Equation (34):

$$\begin{cases} f_t = \sigma\left(W_f[h_{t-1}, x_t] + \boldsymbol{b}_f\right) \\ i_t = \sigma(W_i[h_{t-1}, x_t] + \boldsymbol{b}_i) \\ c_t = \sigma(W_c[h_{t-1}, x_t] + \boldsymbol{b}_c) \\ o_t = \sigma(W_o[h_{t-1}, x_t] + \boldsymbol{b}_o) \end{cases} \tag{34}$$

By setting the weight matrix $\boldsymbol{W}$ for each gate in Equation (34) to 0 or 1, features of the input parameters can be effectively added and deleted through a dot product operation, and the output range of the sigmoid function is (0, 1). The final output of LSTM at time t is determined by the present input x and the output gate, which can be described as follows:

$$h_t = o_t tanh(c_t) \tag{35}$$

After the extraction of the time-series feature, a classic FCN is linked to the output weight. The entire model is shown in Figure 9.

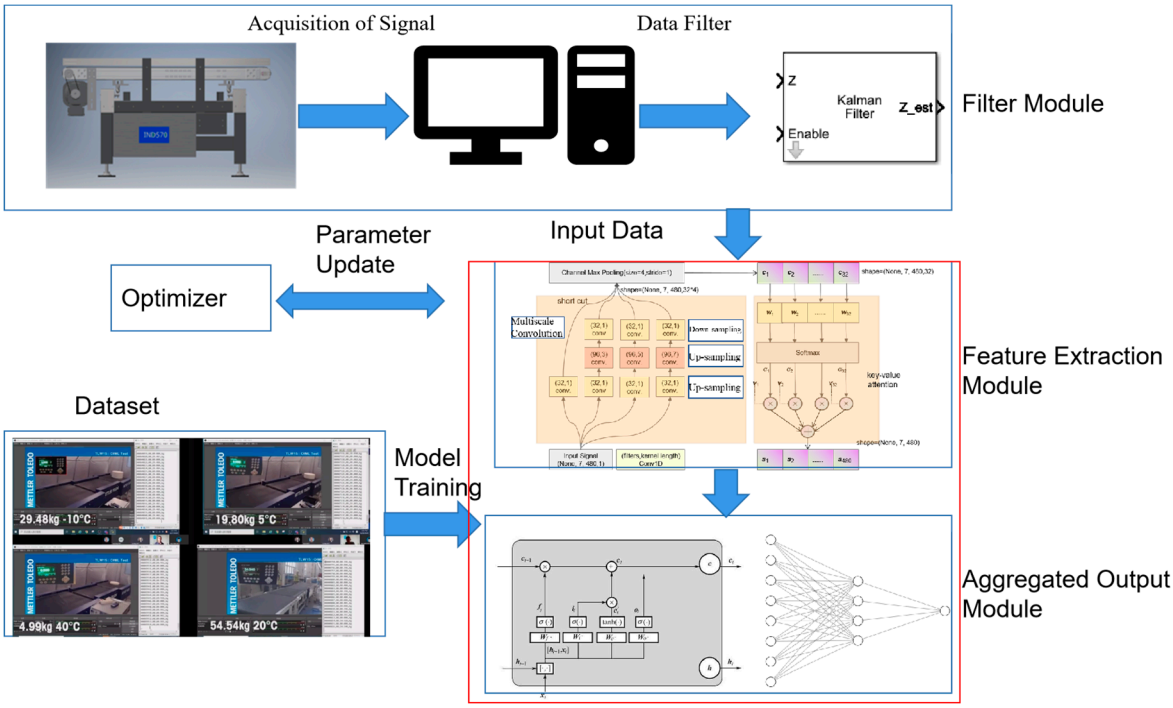


Figure 9. Structure of the entire model.

In Figure 9, the training dataset is shown in Table 1; 80% of the dataset is used for model training and the remaining for model validation. The deep learning model consists of the feature extraction and aggregated output modules; the model's training is described in the next section.

3.6. Model Training

The MultiConv1d-attention-LSTM model was built under Python 3.7.3 and the Pytorch 1.8.0. Toolkit. The training process is mainly about matrix calculation and gradient spread; the graphics processing unit (GPU)'s performance is remarkable faster than the center processing unit (CPU). By using CUDA11.0 and CUDNN 8.0.1, the model can be deployed in GPU. The computer configuration follows: CPU (i9-10900k), GPU (RTX3080) and RAM (32G 3200). Figure 6 shows the training loss and validation loss under different optimizations.

In Figure 10, diagrams show the training and validation loss curve with different optimizers under 100 iteration epochs. The optimizer determined the strategies of the learning rate. The SGD and SGD with Momentum update the learning parameters according to the loss gradient by randomly selected samples. This strategy is valid in the early period. However, the loss diverges in the end as the corresponding training process shown in Figure 10b,d. SGD optimizer amends the model according to the gradient. However, the loss showed the tendency of divergence due to the learning rate is fixed at each stage of the iteration. The models cannot attain an ideal state at the end of iteration. Adam is an optimizer that combines the advantages of Adadelta and RMSprop. Adam adaptively computes the learning rate for the training parameters via computation and stores each parameter's exponentially decaying average of previous gradients and squared gradients. However, as shown in Figure 10a, the later period also shows the tendency of divergence. The loss of training and validation shown in Figure 10c indicates decreasing continuity when using NADAM. This method combined Adam and Nesterov accelerated gradient (NAG), which is superior to other optimizers; the final validation loss is convergent at 0.02.

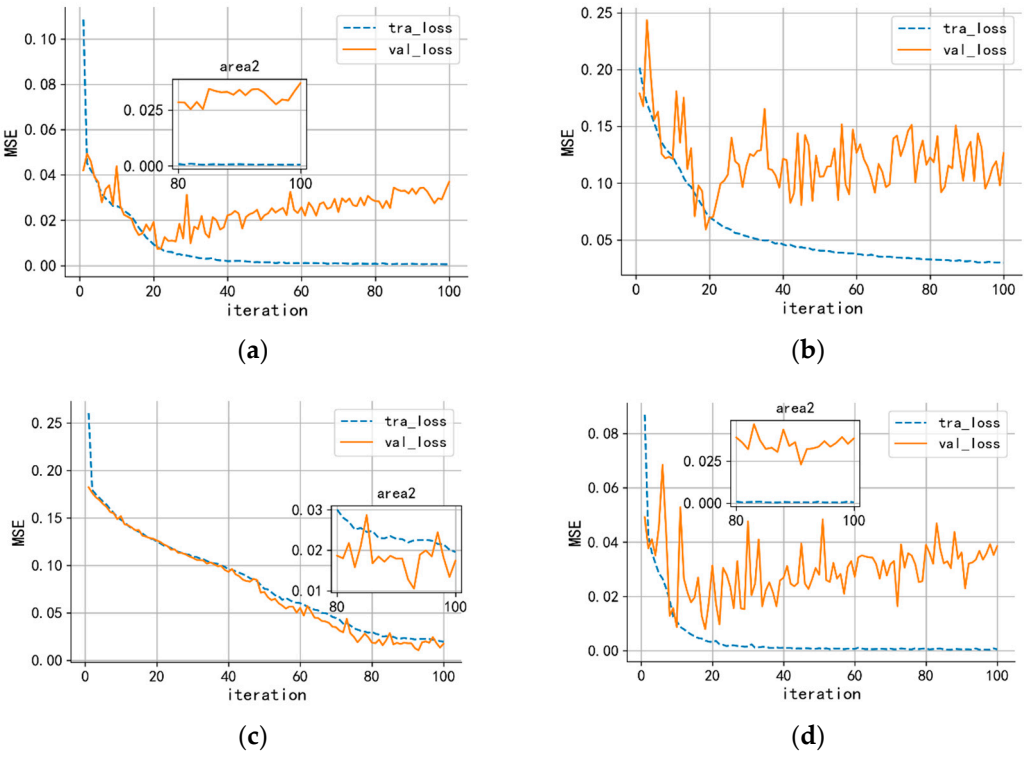


Figure 10. The loss of validation and training under different algorithm optimizers: (**a**) adaptive moment estimation (ADAM); (**b**) stochastic gradient descent (SGD); (**c**) Nesterov accelerated adaptive moment estimation (NADAM); (**d**) SGD with momentum.

4. Performance under a Practical Engineering Situation

Building the Testing Environment

A TW155 dynamic scale in the factory was used to set-up a three-stage drive system to test each algorithm's robustness and portability. The weights and the three-stage drive system (front and rear drive and measurement system) are shown in Figure 11. The weights of the measurement objects were measured using a high precision dynamic balance (static measurement error less than 0.001 g).

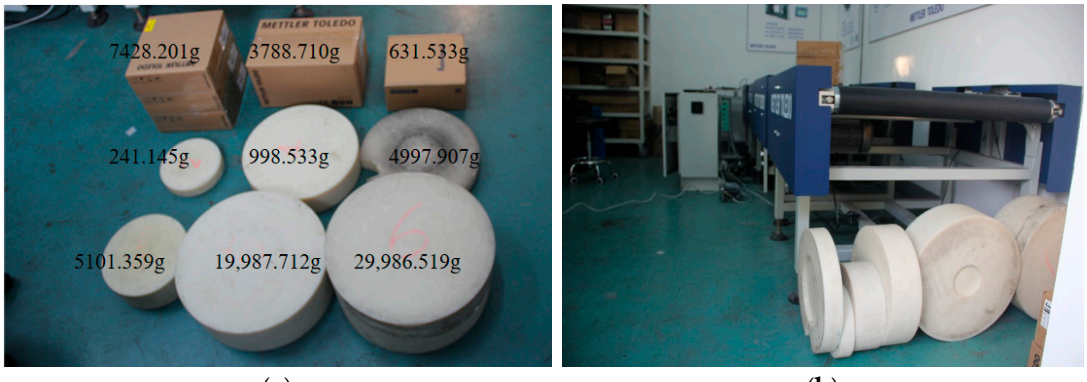


Figure 11. Testing environment: (**a**) measurement object and (**b**) three-stage drive system.

The measure errors for each model were measured under the data sets indicated above. Each target was measured 100 times under different speeds (30, 60, 90 and 120 m/min). The mean absolute error (MAE, $\bar{e}$(kg)), mean relative error (MRE, $\overline{\delta_e}$ (%)) and mean maximum error (MME, $\overline{max}$ (kg)) were used to estimate each algorithm's accuracy and stability. The performance of each algorithm is shown in Tables 2–4.

Table 2. SVM.

SVM	$\bar{e}$ (kg)	$\overline{max}$ (kg)	$\overline{\delta_e}$ (%)
v = 30 (m/min)	0.054	0.071	0.0700
v = 60 (m/min)	0.077	0.115	0.0997
v = 90 (m/min)	0.121	0.157	0.1568
v = 120 (m/min)	0.238	0.329	0.3084

Table 3. FCN.

FCN	$\bar{e}$ (kg)	$\overline{max}$ (kg)	$\overline{\delta_e}$ (%)
v = 30 (m/min)	0.074	0.091	0.0959
v = 60 (m/min)	0.107	0.183	0.1386
v = 90 (m/min)	0.143	0.294	0.1853
v = 120 (m/min)	0.278	0.410	0.3603

Table 4. XGBoost.

FCN	$\bar{e}$ (kg)	$\overline{max}$ (kg)	$\overline{\delta_e}$ (%)
v = 30 (m/min)	0.046	0.0053	0.0596
v = 60 (m/min)	0.097	0.0063	0.1256
v = 90 (m/min)	0.115	0.0074	0.1490
v = 120 (m/min)	0.218	0.0137	0.2824

Each algorithm's performance under different speeds is shown in Tables 2–5. As the speed increases, each model's MAE, MME and MRE shows nonlinear growth. The speed directly determined the valid sampling points and the stability of test objects in motion. The improved Kalman filter combined with the LSTM-attention algorithm shows the best performance under different speeds and loads.

Table 5. Our Model.

Our Model	$\bar{e}$ (kg)	$\overline{max}$ (kg)	$\overline{\delta_e}$ (%)
v = 30 (m/min)	0.034	0.041	0.0441
v = 60 (m/min)	0.057	0.091	0.0739
v = 90 (m/min)	0.084	0.132	0.1089
v = 120 (m/min)	0.108	0.194	0.1401

5. Conclusions

An improved Kalman filter together with a dynamic equation of mechanical structure was used in this paper to estimate the system's state matrix under different speeds. A deep learning base was built to process bulk data and output the weight. The results showed that:

(1) The pressure signal's noise indicates increasing nonlinearity, greatly affecting the accuracy and stability of the weight check in motion as the speed increases.
(2) The improved Kalman filter can efficiently use the WIM system's state matrix to estimate the system's actual situation and filters the noise under different speeds.
(3) Compared with the traditional models, a deep learning-based model decreases error and can greatly improve the system's measurement accuracy.

Although the measurement accuracy of the WIM is improved with our Kalman and LSTM-attention algorithm, the personnel and resources necessary to construct the training sample set are costly. The calculated performance is also required in framework deployment

and is not applicable for common embedded devices. Additional work needs to be carried out in the future:

(1) In logistics weighing, the sampling process can be approximately identified as a uniform velocity compared with vehicle scales, bridge vehicle weighing and other WIM fields. The applicability of the model to other WIM domains needs to be investigated.

(2) A method to simplify the neural network models is required so that the model can be deployed in low-cost embedded devices.

Author Contributions: Conceptualization, B.S. and Y.J.; methodology, B.C. and B.S.; software, K.Y. and X.C.; validation., Y.B.; formal analysis, B.S. and Y.J.; investigation, X.C.; resources, Y.J.; writing—original draft preparation, Y.J. and B.S.; writing-review and editing, B.S., X.C. and Y.J.; visualization, B.S.; supervision, Y.J.; project administration, Y.J.; funding acquisition, Y.J. All authors have read and agreed to the published version of the manuscript.

Funding: The authors express their appreciation for financial support provided by the National Natural Science Foundation of China (No. 51879089) and the Cooperative Innovational Center for Coastal Development and Protection (for the first group, 2011 Plan of China's Jiangsu Province, grant no. (2013) 56).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Khalili, M.; Vishwakarma, G.; Ahmed, S.; Papagiannakis, A.T. Development of a low-power weigh-in-motion system using cylindrical piezoelectric elements. *Int. J. Transp. Sci. Technol.* **2022**, *11*, 496–508. [CrossRef]
2. Wu, J. Sustainable development of green reverse logistics based on blockchain. *Energy Rep.* **2022**, *8*, 11547–11553. [CrossRef]
3. Liang, Z.; Chiu, Y.-H.; Guo, Q.; Liang, Z. Low-carbon logistics efficiency: Analysis on the statistical data of the logistics industry of 13 cities in Jiangsu Province, China. *Res. Transp. Bus. Manag.* **2022**, *43*, 100740. [CrossRef]
4. Burnos, P.; Gajda, J.; Sroka, R.; Wasilewska, M.; Dolega, C. High Accuracy Weigh-In-Motion Systems for Direct Enforcement. *Sensors* **2021**, *21*, 8046. [CrossRef] [PubMed]
5. Xu, S.; Chen, X.; Fu, Y.; Xu, H.; Hong, K. Research on Weigh-in-Motion Algorithm of Vehicles Based on BSO-BP. *Sensors* **2022**, *22*, 2109. [CrossRef]
6. Yang, H.; Yang, Y.; Hou, Y.; Liu, Y.; Liu, P.; Wang, L.; Ma, Y. Investigation of the Temperature Compensation of Piezoelectric Weigh-In-Motion Sensors Using a Machine Learning Approach. *Sensors* **2022**, *22*, 2396. [CrossRef]
7. Pintão, B.; Mosleh, A.; Vale, C.; Montenegro, P.; Costa, P. Development and Validation of a Weigh-in-Motion Methodology for Railway Tracks. *Sensors* **2022**, *22*, 1976. [CrossRef]
8. Shokravi, H.; Shokravi, H.; Bakhary, N.; Heidarrezaei, M.; Koloor, S.S.R.; Petrů, M. Vehicle-Assisted Techniques for Health Monitoring of Bridges. *Sensors* **2020**, *20*, 3460. [CrossRef]
9. Niedźwiecki, M.; Meller, M.; Pietrzak, P. System identification based approach to dynamic weighing revisited. *Mech. Syst. Signal Process.* **2016**, *80*, 582–599. [CrossRef]
10. Gajda, J.; Sroka, R.; Burnos, P. Sensor Data Fusion in Multi-Sensor Weigh-In-Motion Systems. *Sensors* **2020**, *20*, 3357. [CrossRef]
11. Yee, E.; Stewart, J.P.; Schoenberg, F.P. Characterization and utilization of noisy displacement signals from simple shear device using linear and kernel regression methods. *Soil Dyn. Earthq. Eng.* **2011**, *31*, 25–32. [CrossRef]
12. Manarikkal, I.; Elasha, F.; Mba, D. Diagnostics and prognostics of planetary gearbox using CWT, auto regression (AR) and K-means algorithm. *Appl. Acoust.* **2021**, *184*, 108314. [CrossRef]
13. Botros, J.; Mourad-Chehade, F.; Laplanche, D. CNN and SVM-Based Models for the Detection of Heart Failure Using Electrocardiogram Signals. *Sensors* **2022**, *22*, 9190. [CrossRef]
14. Zhang, Z.; Yin, G.; Wu, Z. Joint Estimation of Mass and Center of Gravity Position for Distributed Drive Electric Vehicles Using Dual Robust Embedded Cubature Kalman Filter. *Sensors* **2022**, *22*, 10018. [CrossRef]
15. Lu, H.; Ge, Z.; Song, Y.; Jiang, D.; Zhou, T.; Qin, J. A temporal-aware LSTM enhanced by loss-switch mechanism for traffic flow forecasting. *Neurocomputing* **2020**, *427*, 169–178. [CrossRef]
16. Masud, M.M.; Haider, S.W. Effect of static weight errors on Weigh-in-Motion (WIM) system accuracy. *Measurement* **2023**, *206*, 112301. [CrossRef]
17. Elbeltagi, R. *High Speed Weighing System Analysis via Mathematical Modelling*; Massey University: Auckland, New Zealand, 2012.
18. Halimic, M.; Balachandran, W.; Enab, Y. Fuzzy logic estimator for dynamic weighing system. In Proceedings of the 1996 5th IEEE International Conference on Fuzzy Systems, Part 3 (of 3), New Orleans, LA, USA, 8–11 September 1996; pp. 2123–2129.

19. Luyao, W. Study on the Linkage Development of Logistics Industry and Agriculture: A Case Study of Kaifeng. In Proceedings of the 2022 8th International Conference on Information Management (ICIM), Cambridge, UK, 25–27 March 2022; pp. 218–222. [CrossRef]
20. Alonge, F.; Cusumano, P.; D'Ippolito, F.; Garraffa, G.; Livreri, P.; Sferlazza, A. Localization in Structured Environments with UWB Devices without Acceleration Measurements, and Velocity Estimation Using a Kalman–Bucy Filter. *Sensors* **2022**, *22*, 6308. [CrossRef]
21. Huo, Z.; Wang, F.; Shen, H.; Sun, X.; Zhang, J.; Li, Y.; Chu, H. Optimal Compensation of MEMS Gyroscope Noise Kalman Filter Based on Conv-DAE and MultiTCN-Attention Model in Static Base Environment. *Sensors* **2022**, *22*, 7249. [CrossRef]
22. Xiong, K.; Zhou, P.; Wei, C. Autonomous Navigation of Unmanned Aircraft Using Space Target LOS Measurements and QLEKF. *Sensors* **2022**, *22*, 6992. [CrossRef]
23. Okoniewski, P.; Piskorowski, J. A concept of IIR filters with time-varying coefficients and equalised group delay response. *Meas. J. Int. Meas. Confed.* **2015**, *60*, 13–24. [CrossRef]
24. Zhang, J.Y.; Ying, Y.B.; Jiang, H.Y. Application of optimized digital filters and asymmetrically trimmed mean to improve the accuracy of dynamic egg weighing. *Trans. ASABE* **2017**, *60*, 1099–1111. [CrossRef]
25. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
26. Burnos, P.; Rys, D. The Effect of Flexible Pavement Mechanics on the Accuracy of Axle Load Sensors in Vehicle Weigh-In-Motion Systems. *Sensors* **2017**, *17*, 2053. [CrossRef] [PubMed]
27. Feng, N.; Kang, X.; Han, H.; Liu, G.; Zhang, Y.; Mei, S. Research on a Dynamic Algorithm for Cow Weighing Based on an SVM and Empirical Wavelet Transform. *Sensors* **2020**, *20*, 5363. [CrossRef] [PubMed]
28. Cai, S.; Chen, D.; Fan, B.; Du, M.; Bao, G.; Li, G. Gait phases recognition based on lower limb sEMG signals using LDA-PSO-LSTM algorithm. *Biomed. Signal Process. Control* **2023**, *80*, 104272. [CrossRef]
29. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the International Conference on Computer Vision, Las Condes, Chile, 11–18 December 2015; pp. 1026–1034.
30. Dai, Y.; Zhou, Q.; Leng, M.; Yang, X.; Wang, Y. Improving the Bi-LSTM model with XGBoost and attention mechanism: A combined approach for short-term power load prediction. *Appl. Soft Comput.* **2022**, *130*, 109632. [CrossRef]
31. Fu, R.; Huang, T.; Li, M.; Sun, Q.; Chen, Y. A multimodal deep neural network for prediction of the driver's focus of attention based on anthropomorphic attention mechanism and prior knowledge. *Expert Syst. Appl.* **2023**, *214*, 119157. [CrossRef]
32. Lv, H.; Chen, J.; Pan, T.; Zhang, T.; Feng, Y.; Liu, S. Attention mechanism in intelligent fault diagnosis of machinery: A review of technique and application. *Measurement* **2022**, *199*, 111594. [CrossRef]
33. Wang, W.; Li, Q.; Xie, J.; Hu, N.; Wang, Z.; Zhang, N. Research on emotional semantic retrieval of attention mechanism oriented to audio-visual synesthesia. *Neurocomputing* **2023**, *519*, 194–204. [CrossRef]
34. Li, W.; Chakraborty, M.; Sha, Y.; Zhou, K.; Faber, J.; Rümpker, G.; Stöcker, H.; Srivastava, N. A study on small magnitude seismic phase identification using 1D deep residual neural network. *Artif. Intell. Geosci.* **2022**, *3*, 115–122. [CrossRef]
35. Meade, B.J. Reply to: One neuron versus deep learning in aftershock prediction. *Nature* **2019**, *574*, E1–E4. [CrossRef]