



Article A Deep-LSTM-Based Fault Detection Method for Railway Vehicle Suspensions

Yuejian Chen ¹, Xuemei Liu ¹, Wenkun Fan ², Ningyuan Duan ² and Kai Zhou ^{1,*}

- ¹ Institute of Rail Transit, Tongji University, Shanghai 201804, China; yuejianchen@tongji.edu.cn (Y.C.); xmliu@tongji.edu.cn (X.L.)
- ² Shanghai Marine Diesel Engine Research Institute, Shanghai 201108, China; fwk711@126.com (W.F.); dny9427@126.com (N.D.)
- * Correspondence: zhoukai_mech@tongji.edu.cn

Abstract: The timely detection of faults that occur in industrial machines and components can avoid possible catastrophic machine failure, prevent large financial losses, and ensure the safety of machine operators. A solution to tackle the fault detection problem is to start with modeling the condition monitoring signals and then examine any deviation of real-time monitored data from the baseline model. The newly developed deep long short-term memory (LSTM) neural network has a high nonlinear flexibility and can simultaneously store long- and short-term memories. Thus, deep LSTM is a good option for representing underlying data-generating processes. This paper presents a deep-LSTM-based fault detection method. A goodness-of-fit criterion is innovatively used to quantify the deviation between the baseline model and the newly monitored vibrations as opposed to the mean squared value of the LSTM residual used in many reported works. A railway suspension fault detection method performs better than the vanilla-LSTM-based and linear-autoregression-model-based methods. Using the goodness-of-fit criterion, railway suspension faults can be better detected than when using the mean squared value of the LSTM residual.

Keywords: fault detection; deep long short-term memory; railway vehicle suspensions

1. Introduction

The timely detection of faults that have occurred in railway suspensions is of great significance in terms of health management and maintenance optimization [1]. A fault, if not addressed in time, will keep propagating and may lead to a catastrophic failure of the railway, causing a huge financial loss and threatening the safety of train crew and passengers.

A solution to tackle the fault detection problem is to model the vibration signals in a data-driven manner and then examine the deviation of real-time monitored vibration data from the baseline model. In this context, vibration signals refer to the vibration signals collected when the machine is healthy. The deviation is often measured by a score. The score is then typically compared to a decision limit. When the score exceeds the limit, the machine is categorized as faulty. Fault detection in the industrial machine health monitoring discipline has a close theoretical similarity with the concepts of anomaly detection, novelty detection and outlier detection [2] in application scenarios like medical diagnostics [3] and video surveillance [4].

Many vibration modeling approaches are available, such as the Gaussian mixture model [5], information metrics such as entropy [6], one-class support vector machine that takes features as inputs (also referred to as the safety domain method) [7], nearest neighbors, clustering analyses [8], autoencoders [9], and time-series models [10]. In this paper, time series models are focused on, given that modeling vibration as a time series avoids any additional steps such as domain transformation or feature extraction and thus reduces the



Citation: Chen, Y.; Liu, X.; Fan, W.; Duan, N.; Zhou, K. A Deep-LSTM-Based Fault Detection Method for Railway Vehicle Suspensions. *Machines* **2024**, *12*, 116. https://doi.org/10.3390/ machines12020116

Academic Editor: Domenico Mundo

Received: 18 December 2023 Revised: 2 February 2024 Accepted: 4 February 2024 Published: 7 February 2024



Copyright: © 2024 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 (https:// creativecommons.org/licenses/by/ 4.0/). complexity of fault detection. Another reason that they are studied is that by using time series models, the sequential (spatial) features of the vibration signal can be fully modeled.

Linear time series models are widely used for vibration modeling and thus for fault detection. Linear models include autoregression (AR), moving average (MA), autoregression and moving average (ARMA), etc. Assaad et al. [11] used the AR model to represent the vibrations of a multistage planetary gearbox under constant operating conditions. Gear wear was detected by deriving condition indicators from AR residuals. Yip [12] used the ARMA model to represent the vibration of a three-stage planetary gearbox under constant operating conditions. Health indices were calculated from residuals to determine gear wear trends. Chen et al. [13] proposed a sparse linear parameter varying AR model to represent the vibrations of a fixed-axis gearbox under variable speed conditions. However, data-generation processes (DGPs) may be nonlinear, especially in sophisticated systems like railway vehicles.

Many nonlinear time-series models have been developed in recent decades. One of the easiest ways to account for nonlinearity is to add interaction product terms into linear time-series models. The so-called nonlinear autoregressive and moving average (NARMA) model belongs to this category. The NARMA model was proposed by Kim et al. [14] to represent the structural vibrations of a three-story building structure. A more recent model called the generalized nonlinear autoregressive model (GNAR) was proposed by Ma et al. [15] to represent the vibrations of a rolling bearing. This model can be extended to the GNARMA model by adding the MA part and the corresponding interaction product terms. This GNAR model is more flexible than NARMA given the existence of higher-order interaction product terms. The interaction product terms, however, restrict the nonlinear model from representing the nonlinearity of real-world DGPs. Other types of nonlinear terms are available and can be used in a nonlinear time-series model to achieve a higher modeling accuracy with computational efficiency. In the fields of ecology, finance, and economics, a widely studied nonlinear time-series model is the so-called functionalcoefficient autoregressive (FAR) model (also known as state-dependent autoregressive model) proposed by Chen and Tsay [16] and Cai et al. [17]. The FAR model depends on the choice of a single lagged variable as a model-dependent variable. This limits its applications. A generalization of this FAR model allows for an adaptive search of the dependent variables. This generalized FAR model was proposed by Fan et al. [18]. Furthermore, Ma and Song [19] proposed a varying index coefficient model (VICM) that contains different indexes for each smooth function. When configuring the predictors as lagged responses, the VICM becomes the VICAR (varying index coefficient autoregression) model. The resulting VICAR model is even more flexible than the GFAR model.

Neural networks, ranging from the AR neural network [20,21] to the recurrent neural network (RNN) [22], have recently been a popular choice for nonlinear time-series modeling. An AR neural network requires proper determination of the order of lagged inputs. An RNN, on the other hand, does not require determination of the order of lagged inputs. Conventional RNNs are only capable of looking back approximately ten time steps, due to the fed-back signal either vanishing or exploding [23]. The long short-term memory (LSTM) neural network was proposed to cope with the vanishing or exploding problem by introducing an intermedia cell state [24–26]. As a result, LSTM can simultaneously store long- and short-term information. Therefore, LSTM is a good option for vibration time-series modeling. LSTM, especially when multiple LSTM layers are stacked and thus form so-called deep LSTM, has proven to be successful in many disciplines [27-30]. Challenges associated with (deep) LSTM, however, are proper architecture selection and training hyperparameter determination. Improper architecture selection and training hyperparameter determination will lead to inaccurate modeling of the time series. Researchers have used random search [31] and grid search [32] to select the architecture and determine the training hyperparameters. The validation error was usually used as a criterion to guide the random/grid search.

Over the past decade, researchers have strived to improve standard LSTM, and several variants of LSTM have been reported, including gated recurrent units (GRUs) [33], LSTM with coupled forget and input gates [31], LSTM with peephole connections [31], and other mutations [34]. However, Greff et al. [31] observed that none of these variants could significantly improve standard LSTM.

LSTM and its variants have been used for fault detection in recent years. Yokouchi et al. [35] present a method for anomaly detection in railway vehicle air-conditioning units using LSTM networks. This approach involves learning from normal operational data to build a predictive model and then to predict normal data, establishing a distribution of prediction errors. Rustam et al. [36] propose a method for detecting railway track faults using machine learning and deep learning models, in which acoustic data are analyzed to identify different types of railway track faults. Eunus et al. [37] combine convolutional autoencoders, a ResNet-based RNN, and LSTM to analyze images of railway tracks for detection. Wang et al. [38] applied single-layer LSTM to model fixed-axis gearbox vibration signals and used the residual to detect a bore crack. They also compared the LSTM residual with spectral kurtosis demodulation, resonance demodulation, and unified change detection, and found that the LSTM residual performs better when detecting small cracks. Liu et al. [25] presented a nonlinear predictive (NP) GRU-based denoising autoencoder for bearing fault diagnoses. Its performance was compared with many other autoencoders. For acoustic anomaly detection, Marchi et al. [24] examined LSTM, BLSTM, and MLP when acting as a basic AE, a compressed AE, and a denoising AE, and when configured as NP or not, respectively. They found that the NP-BLSTM-DAE performs the best. However, LSTM has not been used to model railway vibration responses yet, and thus its performance has not been comprehensively assessed in this context.

In this paper, the novelty is, for the first time, the introduction of deep LSTM to model railway vehicle vibration responses. The grid search method is adopted for LSTM architecture selection and training hyperparameter determination based on minimizing the validation error. Once a deep LSTM model is identified to model the vibration response, it is used to detect railway suspension faults by examining any deviation from newly monitored vibration data in the baseline deep LSTM model. Furthermore, a goodness-of-fit criterion is innovatively utilized for quantifying the deviation between the baseline model and the newly monitored vibration data. Previous works used the mean squared error (MSE) [25,38], which essentially is a Euclidean distance [24] between the predicted vibration and newly monitored vibration data. It is found that better fault detection results can be obtained by using the goodness-of-fit criterion.

The structure of this paper is as follows: Section 2 details the fundamentals of deep LSTM; Section 3 presents the deep-LSTM-based fault detection method; Section 4 demonstrates the performance of the deep-LSTM-based fault detection method on a railway suspension; and the concluding remarks are drawn in Section 5.

2. Fundamentals

In this section, deep LSTM is first introduced, and then the designed approach for proper architecture selection and training hyperparameter determination based on minimization of the validation error is presented.

2.1. Deep LSTM

LSTM is a special type of RNN. A unique innovation of LSTM is the introduction of an internal cell state to avoid the gradient vanishing or exploding problem [23], as shown in Figure 1. The inputs of an LSTM block are input features \mathbf{x}_t (dimensions $N_x \times 1$), the cell state \mathbf{C}_{t-1} (dimension $N_h \times 1$), and the hidden state \mathbf{h}_{t-1} (dimensions $N_h \times 1$). The outputs of an LSTM block are the cell state \mathbf{C}_t (dimensions $N_h \times 1$) and the hidden state \mathbf{h}_t (dimensions $N_h \times 1$) at time *t*. This input information will first pass through the forget gate as follows, which determines the information to throw away from the cell state C_{t-1} .

$$\mathbf{f}_t = \sigma \Big(\mathbf{W}_f \mathbf{x}_t + \mathbf{R}_f \mathbf{h}_{t-1} + \mathbf{b}_f \Big), \tag{1}$$

where \mathbf{f}_t is the output of forget gate, which is of size $N_h \times 1$; $\mathbf{W}_f (N_h \times N_x)$ is the input weight matrix; $\mathbf{R}_f (N_h \times N_h)$ is recurrent weight matrix; $\mathbf{b}_f (N_h \times 1)$ is the bias vector for the forget gate; and $\sigma(\cdot)$ denotes sigmoid activation.



Figure 1. An LSTM block.

The next step is the input gate output \mathbf{i}_t (of size $N_h \times 1$), which decides what new information is to be stored in the cell state.

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \mathbf{x}_t + \mathbf{R}_i \mathbf{h}_{t-1} + \mathbf{b}_i), \tag{2}$$

where \mathbf{W}_i ($N_h \times N_x$) is the input weight matrix; \mathbf{R}_i ($N_h \times N_h$) is the recurrent weight matrix; and \mathbf{b}_i ($N_h \times 1$) is the bias vector.

Then, the cell state is updated:

$$\tilde{\mathbf{C}}_t = \tanh(\mathbf{W}_g \mathbf{x}_t + \mathbf{R}_g \mathbf{h}_{t-1} + \mathbf{b}_g), \tag{3}$$

$$\mathbf{C}_t = \mathbf{f}_t \times \mathbf{C}_{t-1} + \mathbf{i}_t \times \tilde{\mathbf{C}}_t,\tag{4}$$

where $\tilde{\mathbf{C}}_t$ denotes the candidate cell state which is of size $N_h \times 1$; tanh(·) is the hyperbolic tangent activation function; \mathbf{W}_g and \mathbf{R}_g are the input and recurrent weights, respectively, where \mathbf{W}_g is of size $N_h \times N_x$ and \mathbf{R}_g is of size $N_h \times N_h$; and $\mathbf{b}_g (N_h \times 1)$ is the bias vector. The earlier state \mathbf{C}_{t-1} is multiplied element-wise by \mathbf{f}_t , thereby discarding the information that was decided to be forgotten. Then, the candidate value $\tilde{\mathbf{C}}_t$ is added and scaled by \mathbf{i}_t , which denotes how much information we decided to update each state value with.

The last step is updating the output gate and hidden state:

$$\mathbf{o}_t = \sigma(\mathbf{W}_o \mathbf{x}_t + \mathbf{R}_o \mathbf{h}_{t-1} + \mathbf{b}_o), \tag{5}$$

$$\mathbf{h}_t = \mathbf{o}_t \times \tanh(\mathbf{C}_t),\tag{6}$$

where \mathbf{o}_t denotes the output vector, which is of size $N_h \times 1$; \mathbf{W}_o and \mathbf{R}_o are the input and recurrent weights; \mathbf{W}_o is of size $N_h \times N_x$, where whereas \mathbf{R}_o is of size $N_h \times N_h$; and $\mathbf{b}_o (N_h \times 1)$ is the bias vector.

When applied to time series representation, x_t may be input. \hat{x}_{t+1} at time t + 1 is obtained from aggregation of the hidden state \mathbf{h}_t .

$$\hat{x}_{t+1} = \mathbf{W}_c^T \mathbf{h}_t,\tag{7}$$

where \mathbf{W}_c is the weight matrix of size $N_h \times 1$ and T denotes the transpose operation.

Figure 2 shows deep LSTM for time series representation. In this case, multiple LSTM layers are stacked. The hidden state output from the lower LSTM layers is treated as an input in higher layers. Note that vanilla LSTM refers to the basic LSTM neural network structure with only one LSTM layer ($N_l = 1$). Compared to deep LSTM, vanilla LSTM is much simpler and less complex.



Figure 2. Deep LSTM for time-series modeling.

Let $\mathbf{W} = [\mathbf{W}_{f} \ ^{T} \ \mathbf{W}_{i} \ ^{T} \ \mathbf{W}_{g} \ ^{T} \ \mathbf{W}_{o} \ ^{T} \ \mathbf{W}_{c} \ ^{T}]^{T}$, $\mathbf{R} = [\mathbf{R}_{f} \ ^{T} \ \mathbf{R}_{i} \ ^{T} \ \mathbf{R}_{g} \ ^{T} \ \mathbf{R}_{o} \ ^{T}]^{T}$, and $\mathbf{b} = [\mathbf{b}_{f} \ ^{T} \ \mathbf{b}_{g} \ ^{T} \ \mathbf{b}_{o} \ ^{T}]^{T}$, where N_{l} denotes the number of LSTM layers, the superscript l in \mathbf{W}^{l} , \mathbf{R}^{l} , \mathbf{b}^{l} denotes the associated weight matrices for the *l*th layer, and the loss function is described as

$$loss = \frac{1}{N} \sum_{t=2}^{N} (\hat{x}_t - x_t)^2 + \lambda \sum_{l=1}^{N_l} \left(\left\| \mathbf{W}^l \right\| + \left\| \mathbf{R}^l \right\| + \left\| \mathbf{b}^l \right\| \right),$$
(8)

where *N* is the length of the data sequence, λ is an L2 regulator, and $\|\cdot\|$ denotes the L2 norm. The training problem becomes:

$$\begin{pmatrix} \mathbf{W}^{l}, \mathbf{R}^{l}, \mathbf{b}^{l} \end{pmatrix} = \underset{\mathbf{W}^{l}, \mathbf{R}^{l}, \mathbf{b}^{l}}{augmin \ loss \ for \ l = 1, 2, \dots, N_{l}$$
(9)

The Adam [39] is usually employed to solve this training problem.

2.2. Architecture Selection and Training Hyperparameter Determination

Proper architecture selection and training hyperparameter determination are important. The grid search method is adopted for architecture selection and training hyperparameter determination based on minimization of the validation error.

The hyperparameters include the number of layers N_l and the number of hidden states in each layer, the minimal batch size, the maximum epochs, the L2 regulator λ , and the learning rate. For efficiency reasons, the number of hidden states is assumed to drop by half from a low layer to its succinct higher layer.

The procedure to determine these architecture and training hyperparameters is:

- Configure search candidate sets for each hyperparameter. For instance, the set for Nh1 is = [10:10:100 150:50:1200], where 10:10:100 denotes a sequence from 10 to 100 with an increment of 10.
- (2) Use training data to train deep LSTM given specific hyper-parameter values, and then obtain the validation error.
- (3) Repeat step (2) until all possible combinations of hyperparameters are evaluated.
- (4) Find the combination of hyperparameters that gives the minimal validation error.

Note that when the number of combinations of hyperparameters is large, a genetic algorithm may be employed to find the optimal hyperparameters.

3. Deep-LSTM-Based Fault Detection Method

Figure 3 illustrates the procedure of deep LSTM for fault detection. First, the sensors and the data acquisition systems are installed, which are essential prerequisites. Next, the collected data are utilized to train the deep LSTM model. Following training, additional monitoring data are gathered and fed into the previously trained deep LSTM, resulting in an estimated series. The detection criterion is then determined by comparing the similarity between the estimated time series and the actual collected data. Finally, with the established criterion and a decision limit in place, fault detection is feasible.



Figure 3. Deep-LSTM-based fault detection method.

In this paper, the goodness-of-fit, *fit*, criterion is used between the estimated time-series data and true collected data for fault detection:

$$fit = 1 - \sum_{t=2}^{N} (x_t - \hat{x}_t)^2 / \sum_{t=2}^{N} (x_t - \overline{x}_t)^2, \qquad (10)$$

where \overline{x}_t is the mean value of the collected data. This *fit* criterion measures the discrepancy between the expected time series generated by deep LSTM and the true collected series. As deep LSTM was trained using data collected in a healthy state, the expected time series would have a high fit with data collected under a healthy state. Once the health state changes, deep LSTM is no longer a good representation of the underlying DGP, and thus the modeled time series would have low agreement with the collected data.

As for the detection threshold, different threshold values will lead to the true positive rate (TPR) and the false positive rate (FPR). In this paper, we plot the receiver operating curve with the TPR against the FPR across all possible threshold values. The specific detection threshold is determined by the users based on their tolerance to the TPR and the FPR.

4. Detection of Railway Vehicle Suspension Faults

In this section, the deep-LSTM-based method is applied to railway suspension fault detection. A comparison with the AR-based fault detection method is also made.

4.1. Railway Vehicle Dynamics Model

A railway vehicle suspension model [40–42], as depicted in Figure 4, is used for generating vertical railway suspension dynamic responses. The model has six degrees of freedom: the bounce *Z* and pitch β of the car body and the two bogies (front and back). The equations of motion are given in [42], as well as the parameters.



Figure 4. Vehicle dynamics model.

The assumptions made in the model are:

- The car body is symmetric and rigid.
- The bogies have a symmetric and rigid body.
- The wheel is modeled as a massless point that follows the rail surface.
- The damping and stiffness are fixed constants.

4.2. Simuluation Configuration

The suspension fault detection relies on analyzing car body vibrations. The dynamic model is solved by an improved Newmark method [41]. The track irregularity is modeled as Gaussian white noise (mean = 0 and standard deviation = 0.002 m), following ref. [40]. The train speed v is 72 km/h, and the sampling frequency f_s is 10 Hz. Car body vibrations are generated under four health states: healthy state and states with a 10% reduction, 20% reduction, and 30% reduction in the stiffness of the secondary suspension K_{sz1} from its nominal value due to faults like material deterioration. Under each health state, 200 segments were generated as testing data. For training and validation purposes, 50 segments were generated under the healthy state. Each segment lasts 50 s. Table 1 summarizes the data for training, validation, and testing.

Table 1. Simulated data.

	Training	Validation		Testing		
Health state	Healthy	Healthy	Healthy	$K_{sz1} \ 10\% \ 200$	K _{sz1} 20%	K _{sz1} 30%
Number of segments	50	50	200		200	200

4.3. Performance of the Deep-LSTM–Based Fault Detection Method

The deep LSTM model was trained and its hyperparameters were optimized while minimizing the validation error. The search space was set as follows: number of hidden layers $N_l = [1 \ 2 \ 3 \ 4 \ 5]$, number of hidden state $N_{h1} = [10:10:100 \ 150:50:1200]$, L2 regulation $\lambda = [1 \times 10^{-1} \ 1 \times 10^{-2} \ 1 \times 10^{-3} \ 1 \times 10^{-4}]$, and learning rate $[0.001 \ 0.005]$. The optimal deep LSTM model is $N_l = 5$, $N_h = [800 \ 400 \ 200 \ 100 \ 50]$, $\lambda = 0.01$, and learning rate = 0.005. The minimal batch size was fixed at 128 and the maximum epochs were 500. The algorithm was executed using a Tesla T4 Graphic Processing Unit. The computational time was 25 h, 26 min, and 27 s.

The raw car body vibration signals were first standardized. The modeling performance is quantified using the root mean squared error (RMSE) and mean absolute error (MAE).

RMSE =
$$\sqrt{\frac{1}{N-1} \sum_{t=2}^{N} (x_t - \hat{x}_t)^2}$$
, (11)

$$MAE = \frac{1}{N-1} \sum_{t=2}^{N} |x_t - \hat{x}_t|.$$
(12)

The RMSE and MAE of the trained deep LSTM model are listed in Table 2, which were measured using healthy testing data.

Table 2. Modeling accuracy measured using healthy test data. The value in parentheses is the relative ratio to the smallest criterion.

Models	Testing RMSE (m/s ²)	Testing MAE (m/s ²) ²
AR	0.3814 (108.1)	0.3049 (108.3)
Vanilla LSTM	0.3866 (109.6)	0.3086 (109.6)
Deep LSTM	0.3529 (100.0)	0.2815 (100.0)

For comparison, an AR model with $n_a = 65$ is used with the same training data. n_a was chosen by minimizing the Bayesian information criterion [43]. The modeling accuracy was measured using the same testing data. Evidently, deep LSTM is more accurate than the AR model. Both the RMSE and MAE values are about 8% higher, thanks to deep LSTM's highly nonlinear model flexibility. One advantage of the AR model, on the other hand, is its high efficiency. The consumed central processing unit time was only 3 min and 7 s, significantly lower than the time of deep LSTM.

The modeling performance of vanilla LSTM ($N_l = 1$) is also reported in Table 2 for comparison. It is evident that using deep LSTM is necessary to accurately model the baseline car body vibrations. In fact, vanilla LSTM performs even slightly worse than the AR model.

The deep-LSTM-based method was used for fault detection. The goodness-of-fit values were obtained and are shown in Figure 5. For comparison, Figure 5 also shows the goodness-of-fit values obtained using the AR model and the vanilla LSTM model. With the increase in fault severity, the goodness-of-fit value decreases. This implies that the more severe the fault is, the less similar the car body responses will be to the responses in a healthy state. Therefore, this goodness-of-fit value can be used for fault detection.

To show the motivation behind selecting the goodness-of-fit value as a fault detection criterion, instead of the MSE, the MSE value of the deep LSTM residual is presented in Figure 5. The MSE of the residual, essentially a Euclidean distance [24], has been widely used by researchers [25,38] as a fault detection criterion. As shown in Figure 5, the MSE value increases as the fault severity increases. This implies that the more severe the fault is, the larger the prediction error will be.

However, it is hard to objectively compare the fault detection performance of each method by observing the goodness-of-fit values and the fault detection performance when using either the goodness-of-fit value or the MSE. Meanwhile, a suitable fault detection threshold is also not clear. To this end, the receiver operating curve (ROC) was plotted and the area under the receiver operating curve (AUC) was calculated to provide a quantitative comparison.





 $K_{sz1}\,10\%$

0

Healthy

0.9

Figure 5. Detection criterion under each health state: (top row) goodness-of-fit values from deep LSTM, (second row) goodness-of-fit values from vanilla LSTM, (third row) goodness-of-fit values from AR, (bottom row) MSE values from deep LSTM.

The ROC was constructed by plotting the true positive rate (TPR) against the false positive rate (FPR) across all possible threshold values. Consider a test set comprising Nfaulty (negative) samples and P healthy (positive) samples. At a specific threshold, TP samples out of the *P* healthy samples are correctly identified as healthy, while *NP* samples out of N faulty samples fail to be identified. To determine the TPR and FPR at this threshold, the following equations were used:

$$\Gamma PR = \frac{TP}{P},$$
(13)

$$FPR = \frac{NP}{N}.$$
 (14)

By calculating the true positive and false positive rates across various thresholds, the receiver operating curve can be plotted, with the false positive rate on the x-axis and the true positive rate on the y-axis. The AUC of this ROC, obtained through numerical integration, offers a comprehensive evaluation of the model's fault detection performance by integrating the true and false positive rates. The AUC, ranging from 0 to 1, serves as an indicator of model accuracy-the higher, the better. The ROC was obtained using the MATLAB function roc() and is shown in Figure 6.



Figure 6. Fault detection performance: (a) K_{sz1} 10%, (b) K_{sz1} 20%, (c) K_{sz1} 30%. False positive rate denotes a type I error. True positive rate denotes a type II error.

From Figure 6, it can be observed that (1) When the goodness-of-fit value is used as a fault detection criterion, deep LSTM performs better than the AR and vanilla LSTM methods in detecting three different railway suspension faults, which is in good agreement with the modeling accuracy observation. To quantitatively compare the fault detection performance, the AUC was calculated using the MATLAB function *trapz()*, and it is listed in Table 3. Evidently, deep LSTM performs better than the AR and vanilla LSTM methods in detecting three different railway suspension faults. (2) Using the goodness-of-fit value for fault detection is much better than using the MSE. When goodness-of-fit values are used, the deep=LSTM-based method returns AUC values of 0.8128, 0.9899, and 1.000 for the three different railway suspension faults, respectively. But when using the MSE, the deep-LSTM-based method only returns AUC values of 0.6462, 0.9450, and 0.9945 for three different railway suspension faults, respectively. This result confirms the superiority of using the goodness-of-fit criterion for fault detection.

Table 3. Area under the ROC.

Models	Detection Criterion	<i>K</i> _{<i>sz</i>1} 10%	<i>K</i> _{<i>sz</i>1} 20%	<i>K</i> _{<i>sz</i>1} 30%
AR	fit	0.7284	0.9430	0.9979
Vanilla LSTM	fit	0.7273	0.9504	0.9976
Deep LSTM	fit	0.8128	0.9899	1.0000
Deep LSTM	MSE	0.6462	0.9450	0.9945

5. Discussion

In this paper, deep LSTM is, for the first time, used to model the railway vehicle vibration response. Furthermore, a goodness-of-fit criterion is innovatively utilized to quantify the deviation between the baseline model and the monitored vibrations.

11 of 13

It is worth mentioning that a transformer and LSTM have different architectures. Transformers use an attention mechanism that processes data points in parallel, allowing them to capture long-range dependencies efficiently. LSTM, a member of the RNN family, processes data sequentially and uses a gating mechanism to handle long-term and short-term dependencies. Transformers are well suited for tasks where the context of the entire sequence is important, like language translation or text generation. LSTM excels in tasks where sequential data with time dependencies, such as time-series analysis or speech recognition, are crucial. Transformers often outperform LSTM in handling large datasets and in complex tasks, especially in NLP. However, LSTM can be more efficient with smaller datasets and simpler tasks. Moreover, training transformers generally requires more computational power and memory, especially for longer sequences, due to their parallel processing nature. Therefore, LSTM is used in our work, and a transformer is not used for comparison. In the future, we will test the performance of transformers using experimental data.

Despite the advancements this paper achieved, there are some limitations to consider. The assumption that the number of hidden states will drop by half from a low layer to its smaller high layer in deep LSTM may limit the network's performance. This represents a potential area for future investigation, where removing this assumption could possibly enhance the deep LSTM neural network's capabilities but would entail higher computational costs. Moreover, the current study is based on simulated signals. For practical applications, further validation with experimental signals is necessary. Thus, another direction for future research involves validating the effectiveness of the presented method on experimental data, ensuring its reliability and robustness in real-world scenarios.

The temporal evolution of the fault (fault trends) is meaningful and prediction of the fault ahead of time is essential for decision making in railway vehicle maintenance. The focus of this paper is primarily on fault detection. We will study the temporal evolution of faults and try to anticipate the fault ahead of its occurrence in the future.

6. Conclusions

In this paper, a deep-LSTM-based fault detection method is presented for railway suspensions. The grid search method is adopted for architecture selection and training hyperparameter determination based on the minimization of the validation error. The goodness-of-fit criterion is proposed for use in fault detection as opposed the MSE of LSTM residuals used in many reported works. A benchmark study using railway vehicle dynamic simulation data has shown that the deep-LSTM-based fault detection method performs better than the vanilla-LSTM-based and linear-time-series-model-based methods. Additionally, using the goodness-of-fit criterion can lead to better fault detection results compared to using the mean squared value of the LSTM residual.

The data and codes associated with this paper are available at: https://github.com/ yuejianchen/LSTM-for-suspension-detection (accessed on 17 December 2023).

Author Contributions: Conceptualization, Y.C.; methodology, Y.C. and X.L.; software, Y.C.; validation, X.L.; formal analysis, W.F. and N.D.; investigation, N.D.; resources, N.D.; data curation, Y.C.; writing—original draft preparation, Y.C.; writing—review and editing, K.Z.; visualization, W.F. and N.D.; supervision, K.Z.; project administration, K.Z.; funding acquisition, W.F., N.D. and K.Z. All authors have read and agreed to the published version of the manuscript.

Funding: The Shanghai Raising-star Program (Grant Number: 22YF1450500) and the Fundamental Research Funds for the Central Universities.

Data Availability Statement: Data are contained within the article.

Acknowledgments: The editors and anonymous reviewers are acknowledged.

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

References

- 1. Pascal, V.; Toufik, A.; Manuel, A.; Florent, D.; Frédéric, K. Improvement indicators for Total Productive Maintenance policy. *Control. Eng. Pract.* **2018**, *82*, 86–96. [CrossRef]
- Pimentel, M.A.; Clifton, D.A.; Clifton, L.; Tarassenko, L. A review of novelty detection. *Signal Process.* 2014, 99, 215–249. [CrossRef]
 Clifton, L.; Clifton, D.A.; Watkinson, P.J.; Tarassenko, L. Identification of patient deterioration in vital-sign data using one-class
- Chiton, E., Chiton, D.A., Watkinson, F.J., Tarassenko, E. Identification of patient deterioration in vital-sign data using one-class support vector machines. In Proceedings of the 2011 Federated Conference on Computer Science and Information Systems (FedCSIS), Szczecin, Poland, 18–21 September 2011; pp. 125–131.
- Diehl, C.; Hampshire, J. Real-time object classification and novelty detection for collaborative video surveillance. In Proceedings
 of the 2002 International Joint Conference on Neural Networks (IJCNN), Honolulu, HI, USA, 12–17 May 2002.
- Schmidt, S.; Heyns, P.S.; Gryllias, K.C. A discrepancy analysis methodology for rolling element bearing diagnostics under variable speed conditions. *Mech. Syst. Signal Process.* 2018, 116, 40–61. [CrossRef]
- 6. Li, Y.; Yang, Y.; Wang, X.; Liu, B.; Liang, X. Early fault diagnosis of rolling bearings based on hierarchical symbol dynamic entropy and binary tree support vector machine. *J. Sound Vib.* **2018**, *428*, 72–86. [CrossRef]
- Liu, Z.; Kang, J.; Zhao, X.; Zuo, M.J.; Qin, Y.; Jia, L. Modeling of the safe region based on support vector data description for health assessment of wheelset bearings. *Appl. Math. Model.* 2019, 73, 19–39. [CrossRef]
- Kim, D.; Kang, P.; Cho, S.; Lee, H.-J.; Doh, S. Machine learning-based novelty detection for faulty wafer detection in semiconductor manufacturing. *Expert Syst. Appl.* 2012, 39, 4075–4083. [CrossRef]
- Sakurada, M.; Yairi, T. Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction. In Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis—MLSDA'14, Gold Coast, Australia, 2 December 2014; Association for Computing Machinery (ACM): New York, NY, USA; p. 4.
- 10. Wang, W.; Wong, A.K. Autoregressive Model-Based Gear Fault Diagnosis. J. Vib. Acoust. 2002, 124, 172–179. [CrossRef]
- 11. Assaad, B.; Eltabach, M.; Antoni, J. Vibration based condition monitoring of a multistage epicyclic gearbox in lifting cranes. *Mech. Syst. Signal Process.* **2013**, *42*, 351–367. [CrossRef]
- Yip, L. Analysis and Modeling of Planetary Gearbox Vibration Data for Early Fault Detection. Master's Thesis, University of Toronto (Canada), Toronto, ON, Canada, 2011. Available online: https://search.proquest.com/pqdtglobal/docview/926973307/ abstract/DD1FB488429849ACPQ/1 (accessed on 15 December 2017).
- 13. Chen, Y.; Liang, X.; Zuo, M.J. Sparse time series modeling of the baseline vibration from a gearbox under time-varying speed condition. *Mech. Syst. Signal Process.* **2019**, *134*, 106342. [CrossRef]
- 14. Kim, Y.; Kim, J.; Kim, Y.H.; Chong, J.; Park, H.S. System identification of smart buildings under ambient excitations. *Measurement* **2016**, *87*, 294–302. [CrossRef]
- 15. Ma, J.; Xu, F.; Huang, K.; Huang, R. GNAR-GARCH model and its application in feature extraction for rolling bearing fault diagnosis. *Mech. Syst. Signal Process.* **2017**, *93*, 175–203. [CrossRef]
- 16. Chen, R.; Tsay, R.S. Functional-Coefficient Autoregressive Models. J. Am. Stat. Assoc. 1993, 88, 298-308. [CrossRef]
- 17. Cai, Z.; Fan, J.; Yao, Q. Functional-Coefficient Regression Models for Nonlinear Time Series. J. Am. Stat. Assoc. 2000, 95, 941–956. [CrossRef]
- 18. Fan, J.; Yao, Q.; Cai, Z. Adaptive Varying-Coefficient Linear Models. J. R. Stat. Soc. Ser. B Stat. Methodol. 2003, 65, 57-80. [CrossRef]
- 19. Ma, S.; Song, P.X.-K. Varying Index Coefficient Models. J. Am. Stat. Assoc. 2015, 110, 341–356. [CrossRef]
- Gan, M.; Peng, H.; Peng, X.; Chen, X.; Inoussa, G. A locally linear RBF network-based state-dependent AR model for nonlinear time series modeling. *Inf. Sci.* 2010, 180, 4370–4383. [CrossRef]
- Gan, M.; Chen, C.L.P.; Li, H.-X.; Chen, L. Gradient Radial Basis Function Based Varying-Coefficient Autoregressive Model for Nonlinear and Nonstationary Time Series. *IEEE Signal Process. Lett.* 2014, 22, 809–812. [CrossRef]
- Yu, Y.; Si, X.; Hu, C.; Zhang, J. A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *Neural Comput.* 2019, *31*, 1235–1270. [CrossRef] [PubMed]
- 23. Staudemeyer, R.C.; Morris, E.R. Understanding LSTM—A tutorial into Long Short-Term Memory Recurrent Neural Networks. *arXiv* 2020, arXiv:1909.09586. Available online: http://arxiv.org/abs/1909.09586 (accessed on 15 December 2020).
- 24. Marchi, E.; Vesperini, F.; Squartini, S.; Schuller, B. Deep Recurrent Neural Network-Based Autoencoders for Acoustic Novelty Detection. *Comput. Intell. Neurosci.* 2017, 2017, e4694860. [CrossRef]
- 25. Liu, H.; Zhou, J.; Zheng, Y.; Jiang, W.; Zhang, Y. Fault diagnosis of rolling bearings with recurrent neural network-based autoencoders. *ISA Trans.* 2018, 77, 167–178. [CrossRef]
- 26. Hochreiter, S.; Schmidhuber, J. Long short-term memory. Neural Comput. 1997, 9, 1735–1780. [CrossRef]
- Siami-Namini, S.; Tavakoli, N.; Namin, A.S. A Comparison of ARIMA and LSTM in Forecasting Time Series. In Proceedings of the 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, FL, USA, 17–20 December 2018; pp. 1394–1401.
- 28. Ayvaz, E.; Kaplan, K.; Kuncan, M. An Integrated LSTM Neural Networks Approach to Sustainable Balanced Scorecard-Based Early Warning System. *IEEE Access* 2020, *8*, 37958–37966. [CrossRef]
- 29. Mao, W.; He, J.; Tang, J.; Li, Y. Predicting remaining useful life of rolling bearings based on deep feature representation and long short-term memory neural network. *Adv. Mech. Eng.* **2018**, *10*, 168781401881718. [CrossRef]
- 30. Yu, W.; Kim, I.Y.; Mechefske, C. Remaining useful life estimation using a bidirectional recurrent neural network based autoencoder scheme. *Mech. Syst. Signal Process.* **2019**, 129, 764–780. [CrossRef]

- Greff, K.; Srivastava, R.K.; Koutník, J.; Steunebrink, B.R.; Schmidhuber, J. LSTM: A Search Space Odyssey. *IEEE Trans. Neural* Netw. Learn. Syst. 2017, 28, 2222–2232. [CrossRef] [PubMed]
- 32. Reimers, N.; Gurevych, I. Optimal Hyperparameters for Deep LSTM-Networks for Sequence Labeling Tasks. *arXiv* 2021, arXiv:1707.06799. Available online: http://arxiv.org/abs/1707.06799 (accessed on 6 January 2021).
- Cho, K.; Van Merrienboer, B.; Bahdanau, D.; Bengio, Y. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv* 2014, arXiv:1409.1259. [CrossRef]
- Jozefowicz, R.; Zaremba, W.; Sutskever, I. An Empirical Exploration of Recurrent Network Architectures. In Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 2342–2350.
- Yokouchi, T.; Kondo, M. LSTM-based Anomaly Detection for Railway Vehicle Air-conditioning Unit using Monitoring Data. In Proceedings of the IECON 2021—47th Annual Conference of the IEEE Industrial Electronics Society, Toronto, ON, Canada, 13–16 October 2021; pp. 1–6.
- Rustam, F.; Ishaq, A.; Alam Hashmi, M.S.; Siddiqui, H.U.R.; López, L.A.D.; Galán, J.C.; Ashraf, I. Railway Track Fault Detection Using Selective MFCC Features from Acoustic Data. Sensors 2023, 23, 7018. [CrossRef]
- Eunus, S.I.; Hossain, S.; Ridwan, A.E.M.; Adnan, A. ECARRNet: An Efficient LSTM Based Ensembled Deep Neural Network Architecture for Railway Fault Detection. *Rochester* 2023. [CrossRef]
- Wang, W.; Galati, F.A.; Szibbo, D. LSTM Residual Signal for Gear Tooth Crack Diagnosis. In Advances in Asset Management and Condition Monitoring; Ball, A., Gelman, L., Rao, B.K.N., Eds.; Smart Innovation, Systems and Technologies Book Series; Springer International Publishing: Cham, Switzerland, 2020; pp. 1075–1090. [CrossRef]
- 39. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization, CoRR. *arXiv* **2014**, arXiv:1412.6980. Available online: https://arxiv.org/abs/1412.6980 (accessed on 15 December 2020).
- Aravanis, T.-C.; Sakellariou, J.; Fassois, S. A stochastic Functional Model based method for random vibration based robust fault detection under variable non-measurable operating conditions with application to railway vehicle suspensions. *J. Sound Vib.* 2019, 466, 115006. [CrossRef]
- 41. Zhai, W.; Wang, K.; Cai, C. Fundamentals of vehicle-track coupled dynamics. Veh. Syst. Dyn. 2009, 47, 1349–1376. [CrossRef]
- Chen, Y.; Niu, G.; Li, Y.; Li, Y. A modified bidirectional long short-term memory neural network for rail vehicle suspension fault detection. *Veh. Syst. Dyn.* 2022, *61*, 3136–3160. [CrossRef]
- 43. Chen, Y.; Liang, X.; Zuo, M.J. An improved singular value decomposition-based method for gear tooth crack detection and severity assessment. *J. Sound Vib.* **2019**, *468*, 115068. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.