

Article



Robust Interval Prediction of Intermittent Demand for Spare Parts Based on Tensor Optimization

Kairong Hong¹, Yingying Ren¹, Fengyuan Li¹, Wentao Mao^{2,*} and Xiang Gao²

- ¹ China Railway Tunnel Group, Zhengzhou 450001, China
- ² School of Computer and Information Engineering, Henan Normal University, Xinxiang 453007, China
- * Correspondence: maowt@htu.edu.cn; Tel.: +86-150-3730-1821

Abstract: Demand for spare parts, which is triggered by element failure, project schedule and reliability demand, etc., is a kind of sensing data to the aftermarket service of large manufacturing enterprises. Prediction of the demand for spare parts plays a crucial role in inventory management and lifecycle quality management for the aftermarket service of large-scale manufacturing enterprises. In real-life applications, however, demand for spare parts occurs randomly and fluctuates greatly, and the demand sequence shows obvious intermittent distribution characteristics. Additionally, due to factors such as reporting mistakes made by personnel or environmental changes, the actual data of the demand for spare parts are prone to abnormal variations. It is thus hard to capture the evolutional pattern of the demand for spare parts by traditional time series forecasting methods. The reliability of prediction results is also reduced. To address these concerns, this paper proposes a tensor optimization-based robust interval prediction method of intermittent time series for the aftersales demand for spare parts. First, using the advantages of tensor decomposition to effectively mine intrinsic information from raw data, a sequence-smoothing network based on tensor decomposition and a stacked autoencoder is proposed. Tucker decomposition is applied to the hidden features of the encoder, and the obtained core tensor is reconstructed through the decoder, thus allowing us to smooth outliers in the original demand sequence. An alternating optimization algorithm is further designed to find the optimal sequence feature representation and tensor decomposition factors for the extraction of the evolutionary trend of the intermittent series. Second, an adaptive interval prediction algorithm with a dynamic update mechanism is designed to obtain point prediction values and prediction intervals for the demand sequence, thereby improving the reliability of the forecast. The proposed method is validated using the actual aftersales data from a large engineering manufacturing enterprise in China. The experimental results demonstrate that, compared with typical time series prediction methods, the proposed method can effectively grab the evolutionary trend of various intermittent series and improve the accuracy of predictions made with small-sample intermittent series. Moreover, the proposed method provides a reliable elastic prediction interval when distortion occurs in the prediction results, offering a new solution for intelligent planning decisions related to spare parts in practical maintenance.

Keywords: demand prediction; intermittent time series; tensor decomposition; interval prediction; time series forecasting

1. Introduction

In complicated equipment manufacturing enterprises such as shield tunneling, rail transportation, and wind energy, the cost of maintaining an inventory of spare parts generally accounts for more than 60% of inventory costs [1]. Qualified inventory optimization [2] and flexible scheduling of parts [3] are critical to improve the efficiency of aftermarket service in lifecycle product management [4]. Due to various uncertain factors, such as element failure, project schedule, safe inventory level, etc., there will be an inventory shortage of spare parts in warehouses which, in turn, triggers the demand for spare parts. The



Citation: Hong, K.; Ren, Y.; Li, F.; Mao, W.; Gao, X. Robust Interval Prediction of Intermittent Demand for Spare Parts Based on Tensor Optimization. *Sensors* **2023**, *23*, 7182. https://doi.org/10.3390/s23167182

Academic Editors: Yongbo Li, Bing Li and Khandaker Noman

Received: 8 July 2023 Revised: 1 August 2023 Accepted: 4 August 2023 Published: 15 August 2023



Copyright: © 2023 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/). demand for spare parts can then serve as a kind of sensing data to monitor the maintenance efficiency and evaluate the aftermarket service quality. Accurate prediction of the demand for spare parts plays a supporting role in intelligent inventory optimization. However, in practical operations, parts planning is often linked to new online projects or associated with the unavailability of spare parts, resulting in sporadic demand for spare parts. The data distribution of the demand for spare parts therefore has intermittent characteristics. Precisely predicting the demand of intermittent time series is challenging in spare parts management for manufacturing enterprises.

Due to the intermittent characteristics of these demand data, predicting demand relies heavily on time series prediction. Currently, time series prediction methods can be divided into three categories [5]: (1) statistical methods (e.g., exponential smoothing [6] and moving average [7]); (2) machine learning methods (e.g., support vector regression (SVR) [8], random forests (RF) [9], and LightGBM (light gradient boosting machine) [10,11]); and (3) deep learning methods (e.g., recurrent neural networks (RNN) [12] and long short-term memory (LSTM) [13]). These methods are often applicable to time series with strong periodicity and apparent trends. However, there are some challenges involved in effectively extracting evolutionary patterns from time series with strong randomness, poor continuity, and, especially, small sample sizes, leading to low prediction accuracy for intermittent series. To solve this problem, Croston [14] improved the exponential smoothing algorithm by decomposing intermittent time series into zero-interval and demand sequences. The exponential smoothing algorithm was then applied to each sequence separately, and the results were weighted to improve prediction performance. Syntetos et al. [15] further improved the Croston algorithm and designed the Syntetos–Boylan approximation (SBA) method. The SBA method introduced a bias term $(1 - \alpha/2)$ to mitigate the uncertainty of intermittent distributions. In addition, some studies proposed different metrics, such as the average demand interval (ADI) and the square of the coefficient of variation (CV2) [16], to explore intermittent characteristics to better extract intrinsic information from demand sequences [17]. Another typical approach is to perform hierarchical clustering on demand sequences [18], which divides the original sequences with weak overall patterns into multiple clusters with more significant patterns. Then, different regression algorithms can be applied to the clusters for prediction. Shi et al. [19] proposed the block Hankel tensor-autoregressive integrated moving average (BHT-ARIMA) model, which uses tensor decomposition [20] to extract the intrinsic correlations among multidimensional small-sample sequences. Although the aforementioned methods have achieved certain results, they still have some limitations. First, they are mostly based on the assumption that all sequence demands have predictive value and disregard the interference of abnormal values. In fact, in actual business, due to special events such as natural disasters, emergencies, market fluctuations, and other factors, some spare parts have abnormal demand patterns, which require manual analysis for recognition. Second, demand prediction results are random and unreliable. Operations still want to obtain trustworthy results regarding demand prediction, but the existing methods fail to make valid decisions when the prediction results are distorted.

There is also a similar concept, i.e., abnormal demand forecasting, which refers to the process of forecasting and analyzing abnormal demands that may occur in the future. Guo et al. [21] utilized the passenger flow characteristics extracted by SVR into LSTM to predict abnormal passenger flow. Liu et al. [22] combined statistical learning and linear regression to build a model of the relationship between price discounts and demand for medical consumables. Li et al. [23] proposed a multi-scale radial basis function (MSRBF) network to predict subway passenger flow under special events. Nguyen et al. [24] combined LSTM with SVM to detect outliers in a demand sequence, and then used an LSTM neural network to predict the occurrence of outliers. Li et al. [25] proposed a combination model of time series and regression analysis, plus dummy variables, to predict special factors related to passenger flow. These methods aim to predict the occurrence of abnormal but meaningful events in many fields. However, in the situation described by this paper, abnormal demands are commonly found due to reporting mistakes made by personnel or environ-

mental changes. Such abnormal demands are harmful to the prediction of the intermittent time series, since the evolutionary pattern of the demand for spare parts is disturbed. There is no value in predicting abnormal demands if such demands have no predictability. This paper is, therefore, solely devoted to predicting normal demands against the disturbances induced by abnormal demands, instead of predicting the abnormal demands.

To solve the problems mentioned above, this paper employs a tensor decomposition technique to smooth the abnormal demands in demand sequences. Tensor Tucker decomposition decomposes a tensor into a set of factor matrices and one core tensor that can describe the intrinsic information from raw data. With the decomposed forms, tensor Tucker decomposition brings the potential to extract the evolutionary trend from intermittent time series. The concept of a prediction interval is further introduced to tackle the problem of high uncertainty and less reliability in the prediction results. The methodology is as follows: First, a tensor autoencoder network is constructed, which performs tensor Tucker decomposition on the output features extracted from the hidden layers of the autoencoder. Second, the core tensor is decoded and reconstructed with an alternating optimization strategy to obtain the optimal feature representation of intermittent time series. Third, an adaptive prediction interval (API) algorithm is developed with a dynamical update mechanism to obtain point prediction values and prediction intervals, thus improving the reliability of the prediction results. Finally, the performance of the proposed method is validated using a set of real-life aftersales data from a large engineering manufacturing enterprise from China.

The contributions of this paper can be summarized as follows:

(1) An intermittent series smoothing algorithm is proposed. By integrating tensor Tucker decomposition into a stacked autoencoder network with an alternately optimizing scheme, the proposed algorithm extracts the evolutionary trend of the intermittent time series under irregular noise interference. Compared with existing time series anomaly detection methods, the proposed algorithm does not require any pre-fixed detection thresholds, and can adaptively identify outliers in the series. It is highly suitable for smoothing the outliers in the intermittent time series. Moreover, the proposed algorithm is universally applicable, e.g., the stacked autoencoder can be easily replaced by other deep models;

(2) An adaptive prediction interval algorithm is designed. Different from the existing point prediction methods using a deterministic prediction model, this algorithm incorporates the prediction intervals with the point prediction, which can match the intermittent characteristics of demand sequence for spare parts. This algorithm provides an effective solution to address the uncertainty in the demand for spare parts. According to the literature survey, there is no related research about interval prediction, specifically for demand prediction.

2. Background

2.1. Multi-Way Delay Embedding Transform

The multi-way delay embedding transform (MDT) [26] is capable of embedding loworder data into a high-dimensional space, which can be used to construct Hankel matrices or block Hankel tensors. The tensor obtained by MDT possesses the characteristics of low rank, which can smooth the original data to make them easier to train. $v = (v_1, ..., v_L)^T \in \mathbb{R}^L$ denotes a vector that is transformed into a Hankel matrix with a delay of τ through MDT. This process is referred to Hankelization of the vector. The transform process is shown in Equation (1).

$$\mathcal{H}_{\tau}(vs.) := \begin{pmatrix} v_1 & v_2 & \cdots & v_{L-\tau+1} \\ v_2 & v_3 & \cdots & v_{L-\tau+2} \\ \vdots & \vdots & \ddots & \vdots \\ v_{\tau} & v_{\tau+1} & \cdots & v_L \end{pmatrix} \in R^{\tau \times (L-\tau+1)} \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$
(1)

First, the duplication matrix $S \in \{0, 1\}^{\tau \times (L-\tau+1) \times L}$ is constructed with a delay of τ , as shown in Equation (2).

$$S^{T} = \begin{pmatrix} I_{\tau} & & \\ & I_{\tau} & \\ & & \ddots & \\ & & & I_{\tau} \end{pmatrix}_{\tau \times \tau}^{T}$$
(2)

The vector v is transformed into a Hankel matrix denoted by $\mathcal{H}_{\tau}(vs.)$. The duplication matrix S is essentially a linear transformation. The vectorization expansion is shown in Equation (3):

$$vec(\mathcal{H}_{\tau}(vs.)) = Sv, \quad Sv \in R^{\tau \times (L-\tau+1)}$$
(3)

where $vec(\cdot)$ expands the matrix along the column direction. The Hankel matrix through delay embedding can be shown in Equation (4).

$$\mathcal{H}_{\tau}(vs.) = fold_{(L,\tau)}(Sv) := v_{H},$$

$$fold_{(L,\tau)} : R^{\tau \times (L-\tau+1)} \to R^{\tau \times (L-\tau+1)}$$
(4)

where $fold_{(L,\tau)}$ folds a vector into a matrix.

The inverse transform of multi-way delay embedding for vectors can convert the data from a high-dimensional space to a lower-dimensional target space. This can be calculated using Equations (5) and (6).

$$\mathcal{H}_{\tau}^{-1}(V_H) = S^{\dagger} vec(V_H) \tag{5}$$

$$S^{\dagger} := \left(S^T S\right)^{-1} S^T \tag{6}$$

where *t* is the Moore–Penrose inverse matrix [27].

2.2. Tensor Decomposition

Tensor decomposition aims to decompose high-order tensor data into low-rank matrices or vectors [28]. It is commonly applied in data compression, dimensionality reduction, feature extraction, etc. Tensor Tucker decomposition decomposes an *N*th-order tensor $\chi \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ into the product of a core tensor $\zeta_t \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_N}$ and *N* factor matrices $U^{(n)} \in \mathbb{R}^{I_n \times J_n}$, as shown in Equation (7). The factor matrices obtained from Tucker decomposition represent the principal components of the tensor's modular expansion, while the core tensor captures the correlations between these components.

$$\chi = \varsigma \times_1 U^{(1)} \times_2 U^{(2)} \cdots \times_N U^{(N)}$$
⁽⁷⁾

where $\varsigma \times {}_1 U^{(n)}$ is the *n*-mode product of the modular (*n*) expansion of tensor *S* and matrix $U^{(n)} \in \mathbb{R}^{I_n \times J_n}$:

$$\left[\varsigma \times U^{(n)}\right]_{j_1 \cdots j_{n-1} i_n j_{n+1} \cdots j_N} = \sum_{j_n=1}^{J_n} g_{j_1 \cdots j_{n-1} i_n j_{n+1} \cdots j_N} u_{i_n j_n}$$

$$\varsigma \times U^{(n)} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_N}$$
(8)

With the above equation, one data point in the tensor can be expanded by Equation (9)

as:

$$x_{i_1 i_2 \cdots i_N} = \sum_{j_1, j_2, \cdots, j_N} g_{j_1 \cdots j_N} u_{i_1 j_1}^{(1)} u_{i_2 j_2}^{(2)} \cdots u_{i_3 j_3}^{(3)}$$
(9)

Figure 1 illustrates a three-order tensor decomposed using Tucker decomposition, resulting in the product of a smaller core tensor and three factor matrices.



Figure 1. Illustration of three-order tensor Tucker decomposition.

2.3. LightGBM

LightGBM, developed by Microsoft engineers [10], is a gradient-boosting framework based on decision trees. By utilizing feature parallelism during its training process, Light-GBM assigns discrete features to multiple bins and constructs decision trees using histogram algorithms, which provides a quick and efficient training mechanism for LightGBM. Additionally, LightGBM employs a sparse feature algorithm to significantly reduce memory consumption, which is suitable for training with extremely large-scale data. Moreover, LightGBM utilizes a leaf-wise tree growth strategy, which leads to faster convergence and higher accuracy compared to traditional level-wise strategies.

3. Proposed Method

In this section, a tensor optimization-based robust interval prediction method of intermittent time series is presented. This method consists of two parts: (1) a sequence smoothing network based on tensor decomposition and a stacked autoencoder, which aims to smooth anomalous demand values in the original sequence and to extract the evolutionary trend of demand as well; and (2) an adaptive prediction interval algorithm, which aims to construct a reliable prediction interval to avoid the oversupply risk or inventory shortage caused by inaccurate predictions. In this method, the role of tensor Tucker decomposition is: (1) extracting core tensors from the demand sequence for representing the evolutionary trend; and (2) smoothing the outliers in the sequence. The negative interference of anomalous demands can then be effectively reduced under an unsupervised mode. Moreover, training a LightGBM model with core tensors can better mine the trend information from the demand sequence. Tensor Tucker decomposition is believed suitable for intermittent time series forecasting with small samples.

The flowchart of the proposed method is shown in Figure 2. First, the hidden features are extracted from the original data using a stacked autoencoder, then tensor Tucker decomposition is performed on the hidden features to obtain the core tensors. An alternating optimization scheme is designed to obtain the optimal core tensors by alternately updating the autoencoder parameters and tensor factor matrices. Second, an adaptive interval prediction algorithm is constructed. The interval is calculated using the predicted values and prediction residuals from LightGBM estimators. Finally, a dynamic update mechanism is used to adjust the width of prediction interval. The detailed implementation will be presented as follows.



Figure 2. Flowchart of the proposed method.

3.1. Sequence Smoothing Network

Denote the demand sequence by $X = \{x_1, x_2, ..., x_m\} \in \mathbb{R}^{m \times n}$, where *m* indicates the sample number and *n* represents the time dimension. The sequence can be encoded and mapped into a hidden layer, as shown in Equation (10).

$$h = f(Wx + b) \tag{10}$$

where *h* represents the hidden layer features, $f(\cdot)$ is the activation function of the encoding layer, and $W \in \mathbb{R}^r$ and $b \in \mathbb{R}^r$ are the weight matrix and bias vector of the encoding layer, respectively. The obtained feature set is denoted by $H = \{h_1, h_2, \dots, h_m\} \in \mathbb{R}^{m \times l}$, where *l* represents the dimension of the deep features.

In order to adequately represent the temporal information between samples in the feature set *H*, an MDT with the operations of multi-linear duplication and multi-way folding is employed to transform the original sequence to a three-order tensor. This process can also reduce noise disturbance. Denote by $H = \{h_1, h_2, \dots, h_m\} \in \mathbb{R}^{m \times l}$ the tensor of $\mathbb{X} = \{\mathcal{X}_1, \dots, \mathcal{X}_{m-\tau+1}\} \subseteq \mathbb{R}^{l \times \tau \times (m-\tau+1)}$, then the MDT for \mathbb{X} can be defined as:

$$\mathcal{X}_{i} = \mathcal{H}_{\tau}(H) = Fold_{(m,\tau)}(H \times_{1} S_{1} \times \dots \times_{m-\tau+1} S_{m-\tau+1})$$
(11)

where τ and m represent the time window size and sample length, respectively, and *S* is a duplication matrix. In this paper, τ is set to 6.

With the tensors constructed by MDT, the Tucker decomposition technique is introduced to obtain the three-order core tensor $\mathcal{G} = \{g_1^1, ..., g_{m-\tau+1}^l\} \in \mathbb{R}^{l \times \tau \times (m-\tau+1)}$ that represents the essential information of the sequence:

$$\mathcal{G} = \mathbb{X} \times {}_{1} U^{(1)^{\mathrm{T}}} \times {}_{2} U^{(2)^{\mathrm{T}}} \times \cdots \times {}_{V} U^{(v)^{\mathrm{T}}}$$

s.t. $U^{(v)^{\mathrm{T}}} U^{(v)} = I, v = 1, \cdots, V$ (12)

where $\{U^{(v)}\}_{v=1}^{V}$ is the projection matrix and can maximally preserve the temporal continuity between core tensors, where the superscript v represents the tensor dimension, and $m - \tau + 1$ represents the sample length after reconstruction. Equation (12) means that the

decomposition result consists of a core tensor and a series of factor matrices. Obviously, the core tensor \mathcal{G} contains the intrinsic information of an evolutionary trend. By optimizing $\{U^{(v)}\}_{v=1}^{V}$, the inherent correlation between feature sequences can be sufficiently captured, while noise interference can also be reduced. The loss of tensor reconstruction can be calculated as:

$$\mathcal{L}_{Tensor} = \min \left\| \mathbb{X} - \hat{\mathbb{X}} \right\|_{F}^{2}$$
(13)

where \mathbb{X} is the original tensor, and $\mathbb{X} = \mathcal{G} \times_1 U^{(1)} \times_2 U^{(2)} \times \cdots \times_H U^{(H)}$ is the tensor reconstructed from the factor matrix $\{U^{(v)}\}_{v=1}^V$.

To implement the decoding operation, it is necessary to transform the core tensor \mathcal{G} back to the original input space. Here, the inverse MDT transform [6] is applied to \mathcal{G} by reversing the transformation along the time dimension to obtain a second-order core tensor $\mathcal{G}' = \{g'_1, ..., g'_m\} \in \mathbb{R}^{m \times l}$, as shown in Equation (14). The core tensor \mathcal{G}' is then used as the input of decoding to update the network.

$$\mathcal{G}' = \mathcal{H}(\mathcal{G}) = Unfold_{(m,\tau)}(\mathcal{G}) \times {}_{1}S_{1}^{\dagger} \times \cdots \times {}_{m-\tau+1}S_{m-\tau+1}^{\dagger}$$
(14)

where t is the Moore-Penrose pseudo-inverse.

Through decoding \mathcal{G}' , the reconstructed data \hat{x} can be obtained as follows:

$$\hat{x} = f^* (W^* g' + b^*) \tag{15}$$

where $f^*(\cdot)$ is the activation function of the decoding layer, $W^* \in \mathbb{R}^{n \times r}$ is the weight matrix of the decoding layer, and $b^* \in \mathbb{R}^n$ is the bias vector of the decoding layer. Consequently, the reconstruction loss of the autoencoder can be calculated as:

$$\mathcal{L}_{AE} = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{2} \|\hat{x} - x\|^2 + \frac{\lambda}{2} (\|W\|_F^2 + \|W^*\|_F^2)$$
(16)

where λ is the weight decay parameter and $\|\cdot\|$ is the Frobenius norm. Based on the aforementioned analysis, the whole loss function is:

$$\mathcal{L}_{loss} = \sum_{i=1}^{M} \mathcal{L}_{AE} + \eta \mathcal{L}_{Tensor}$$
(17)

Minimizing Equation (17) can smooth anomalous demands in the demand sequence and extract the evolutionary trend of the demand for spare parts. The key idea of this process is to reduce the significant deviations of anomalous demands, making them suitable for intermittent sequence anomaly detection. The crucial aspect of this process lies in utilizing the core tensor to represent the evolutionary trend. As Equation (12) indicates, $\{U^{(v)}\}_{v=1}^{V}$ is randomly initialized. Therefore, the optimization of tensor decomposition in minimizing Equation (17) is required to obtain the optimal representation of the core tensors.

3.2. Alternating Optimization Scheme

Minimization of Equation (17) includes the optimization of \mathcal{L}_{AE} and \mathcal{L}_{Tensor} . The \mathcal{L}_{AE} can be solved using a stochastic gradient descent (SGD) strategy [29]. However, SGD cannot be directly applied to tensor decomposition. Therefore, this paper adopts an alternating optimization strategy: fix $\{U^{(v)}\}_{v=1}^{V}$, and update W, W^* ; then fix the updated W, W^* , and update $\{U^{(v)}\}_{v=1}^{V}$. These two steps are performed alternately until convergence. It should be noted that, since the number of tensor optimization iterations is typically smaller than the number of W, W^* updates, updating $\{U^{(v)}\}_{v=1}^{V}$ is set to stop when the difference between two consecutive tensor optimizations of $\{U^{(v)}\}_{v=1}^{V}$ is less than a specific threshold. W, W^* is updated until the model converges. With the initialized W and W^* , the specific optimization process is as follows:

(1) Fix $\{U^{(v)}\}_{v=1}^{V}$, and update *W*, *W*^{*};

1. Encoding stage: the partial derivatives of \mathcal{L}_{AE} with respect to the parameters are:

$$\begin{cases} \frac{\partial \mathcal{L}_{AE}}{\partial W} = \frac{2}{M} \sum_{m=1}^{M} \left(\hat{x}^{(m)} - x^{(m)} \right) \cdot \frac{\partial \left(\hat{x}^{(m)} - x^{(m)} \right)^{T}}{\partial W} + 2\lambda \frac{\partial \left(\|W\|_{F}^{2} + \|W^{*}\|_{F}^{2} \right)}{\partial W} \\ \frac{\partial \mathcal{L}_{AE}}{\partial b} = \frac{2}{M} \sum_{m=1}^{M} \left(\hat{x}^{(m)} - x^{(m)} \right) \cdot \frac{\partial \left(\hat{x}^{(m)} - x^{(m)} \right)^{T}}{\partial b} \end{cases}$$
(18)

The partial derivatives corresponding to the error term on each sample can be obtained

by:

$$\begin{cases} \frac{\partial (\hat{x}^{(m)} - x^{(m)})}{\partial W} = \sum_{m=1}^{M} \frac{\partial (\hat{x}^{(m)})^{T}}{\partial W} = f' \cdot diag(X^{(m)}) \in \mathbb{R}^{1 \times r} \\ \frac{\partial (\hat{x}^{(m)} - x^{(m)})}{\partial b} = \sum_{m=1}^{M} \frac{\partial (\hat{x}^{(m)})^{T}}{\partial b} = f' \cdot 1_{r} \in \mathbb{R}^{1 \times 1} \end{cases}$$
(19)

where " \cdot " represents the dot product operator, $diag(\cdot)$ is the diagonal matrix, 1_r is the unit column vector of size r, and f' is the derivative of the activation function;

2. Decoding stage: the partial derivatives of \mathcal{L}_{AE} with respect to the parameters are:

$$\begin{cases} \frac{\partial \mathcal{L}_{AE}}{\partial W^{*}} = \frac{2}{M} \sum_{m=1}^{M} \left(\hat{x}^{(m)} - x^{(m)} \right) \cdot \frac{\partial \left(\hat{x}^{(m)} - x^{(m)} \right)^{T}}{\partial W^{*}} + 2\lambda \frac{\partial \left(\|W\|_{F}^{2} + \|W^{*}\|_{F}^{2} \right)}{\partial W^{*}} \\ \frac{\partial \mathcal{L}_{AE}}{\partial b^{*}} = \frac{2}{M} \sum_{m=1}^{M} \left(\hat{x}^{(m)} - x^{(m)} \right) \cdot \frac{\partial \left(\hat{x}^{(m)} - x^{(m)} \right)^{T}}{\partial b^{*}} \end{cases}$$
(20)

For easy analysis, the error propagation term (i.e., the derivative of the error term with respect to the hidden layer output) is briefly denoted by:

$$\delta_{H^{(m)}} = \left(\frac{\partial(\hat{x}^{(m)} - x^{(m)})}{\partial H^{(m)}}\right) \tag{21}$$

Furthermore, following the chain rule, we have:

$$\begin{cases} \frac{\partial (\hat{x}^{(m)} - x^{(m)})}{\partial W^*} = \delta_{H^{(m)}} \cdot \frac{\partial H^{(m)}}{\partial W^*} = \delta_{H^{(m)}} \cdot (f' \odot diag(H^{(m)}))^T \\ \frac{\partial (\hat{x}^{(m)} - x^{(m)})}{\partial b^*} = \delta_{H^{(m)}} \cdot \frac{\partial H^{(m)}}{\partial b^*} = \delta_{H^{(m)}} \cdot (f' \odot 1_r)^T \end{cases}$$
(22)

where " \cdot " and *diag*(\cdot) have the same meaning as in the encoding stage;

3. Substitute Equations (19) and (22) into Equations (18) and (20), respectively, and the model parameters can be updated as:

$$\begin{cases} W^{(l+1)} = W^{(l)} - \xi \cdot \frac{\partial \mathcal{L}_{AE}}{\partial W} \big|_{W=W^{(l)}} \\ b^{(l+1)} = b^{(l)} - \xi \cdot \frac{\partial \mathcal{L}_{AE}}{\partial b} \big|_{b=b^{(l)}} \\ W^{*(l+1)} = W^{*(l)} - \xi \cdot \frac{\partial \mathcal{L}_{AE}}{\partial W^{*}} \big|_{W^{*}=W^{*(l)}} \\ b^{*(l+1)} = b^{*(l)} - \xi \cdot \frac{\partial \mathcal{L}_{AE}}{\partial b^{*}} \big|_{b^{*}=b^{*(l)}} \end{cases}$$
(23)

where ξ is the learning rate.

(2) Fix W, W^* and update $\{U^{(v)}\}_{v=1}^V$.

Updating $\{U^{(v)}\}_{v=1}^{V}$ can be achieved by minimizing the tensor reconstruction error

as shown in Equation (13). Since $\{U^{(v)}\}_{v=1}^{V}$ is an orthogonal matrix, Equation (13) can be rewritten as follows:

$$\begin{aligned} \left\| \mathbb{X} - \hat{\mathbb{X}} \right\|_{F}^{2} &= \left\| \operatorname{vec}(\mathbb{X}) - \left(U^{(V)} \otimes U^{(V-1)} \otimes \cdots \otimes U^{(1)} \right) \cdot \operatorname{vec}(\mathcal{G}) \right\|_{F}^{2} \\ &= \left\| \operatorname{vec}(\mathbb{X}) - \left(U^{(V)} \otimes U^{(V-1)} \otimes \cdots \otimes U^{(1)} \right) \cdot \left(U^{(V)} \otimes U^{(V-1)} \otimes \cdots \otimes U^{(1)} \right)^{T} \cdot \operatorname{vec}(\mathbb{X}) \right\|_{F}^{2} \\ &= \left\| \operatorname{vec}(\mathbb{X}) \right\|_{F}^{2} - \left\| U^{T} \cdot \operatorname{vec}(\mathbb{X}) \right\|_{F}^{2} \end{aligned}$$
(24)

where $U^T = U^{(H)} \otimes U^{(H-1)} \otimes \cdots \otimes U^{(1)}$. To minimize Equation (24), we should maximize the following equation:

$$\left\| U^{T} \cdot vec(\mathbb{X}) \right\|_{F}^{2} = \begin{cases} \left\| U^{(1)^{T}} \cdot A^{(1)} \cdot (U^{(H)} \otimes U^{(H-1)} \otimes \dots \otimes U^{(2)}) \right\|_{F}^{2} \\ \left\| U^{(2)^{T}} \cdot A^{(2)} \cdot (U^{(H)} \otimes U^{(H-1)} \otimes \dots \otimes U^{(1)}) \right\|_{F}^{2} \\ \dots \\ \left\| U^{(H)^{T}} \cdot A^{(H)} \cdot (U^{(H-1)} \otimes U^{(H-2)} \otimes \dots \otimes U^{(1)}) \right\|_{F}^{2} \end{cases}$$
(25)

where $A^{(i)}$ is the unfolding matrix of the tensor X along the *i*-th dimension. The alternating least squares method [30] can be used to solve it, where each factor matrix in different directions is fixed sequentially to find the least squares solution. This process is shown in Equation (26):

$$U_{k+1}^{(v)} = \arg\min_{\{U^{(v)}\}_{n=1}^{V}} \left\| \hat{\mathbb{X}} - \mathcal{G} \times_{M} U_{k}^{(V)} \times_{M-1} \cdots \times_{1} U_{k}^{(1)} \right\|_{F}^{2}$$
(26)

3.3. Adaptive Prediction Intervals (API)

After obtaining the optimal feature representation, this paper designs the API algorithm to generate the point prediction values and reliable prediction intervals. The details of the API algorithm are presented as follows. The API algorithm consists of two stages: training and prediction. At the training stage, a fixed number of LightGBM estimators are fitted from a subset of training data. Then, the predicted value from all the Light-GBM estimators is aggregated using the leave-one-out (LOO) strategy to generate LOO prediction factors and residuals. At the prediction stage, the API algorithm calculates the LOO predicted value by averaging the prediction factors from each test sample. With the predicted values, the center of the prediction interval is determined. The prediction intervals are then established using the LOO residuals. The width of prediction interval is updated using a dynamic updating strategy. The specific implementation steps are as follows:

First, a LightGBM model *f* is trained using the training samples $\{(x_t, y_t)\}_{t=1}^T$, and the prediction interval at the *t*-th time step is calculated as:

$$\hat{C}_{t}^{\alpha} = [\hat{f}_{-t}(x_{t}) + q_{\hat{\beta}}, \hat{f}_{-t}(x_{t}) + q_{(1-\alpha+\hat{\beta})}]$$
(27)

where α is the significance factor, \hat{f}_{-t} represents the *t*-th estimator of f, $q_{\hat{\beta}}$ is the $\hat{\beta}$ -quantile on $\{\hat{\varsigma}_i\}_{i=t-1}^{t-T}$, and $q_{(1-\alpha+\hat{\beta})}$ is the $(1-\alpha+\hat{\beta})$ -quantile on $\{\hat{\varsigma}_i\}_{i=t-1}^{t-T}$. The LOO prediction residual $\hat{\varsigma}_i$ and $\hat{\beta}$ are defined as follows:

$$\hat{\varsigma}_i = y_i - \hat{f}_{-t}(x_t) \tag{28}$$

$$\hat{\beta} = \arg\min_{\beta \in [0,\alpha]} (\hat{f}_{-t}(x_t) + q_{(1-\alpha+\hat{\beta})} - \hat{f}_{-t}(x_t) + q_{\hat{\beta}})$$
(29)

Second, the interval center is $\hat{f}_{-t}(x_t)$, and the interval width is the difference between the $(1 - \alpha + \hat{\beta})$ and $\hat{\beta}$ -quantiles on the past *T* residuals.

However, the obtained interval values cannot adequately address the problem of large fluctuations of demands for different spare parts. This section proposes an adaptive update mechanism to improve the reliability of the prediction interval. The specific step is as follows: First, each demand sequence is divided into a zero-interval sequence (i.e., the gap between two subsequent demands occur) and a non-zero sequence (i.e., the real demand values). The squared coefficient of variation is then calculated for the two sequences , which are denoted by cv_1 and cv_2 , respectively. A larger coefficient indicates greater sequence fluctuation, thus requiring a larger interval width for prediction, and vice versa. Second, initialize the interval width parameter $\alpha = 0.1$ and update α by:

$$\alpha = \alpha + cv_1 + cv_2 \tag{30}$$

The interval update mechanism, as shown in Equation (30), can improve the rationality and reliability of the prediction interval for demand sequences with different volatility and intermittency characteristics. Obviously, different values of interval widths will directly affect the decision of inventory management, e.g., spare part coverage rate. The spare part coverage rate, reflected by the interval coverage rate in this paper, is defined as the ratio of the number of demands covered by the interval to the total number of demands in the sequence. The mechanism begins by setting an initial interval with a larger width, and then reduces the width to meet the fluctuations in the demand sequence. The update process is stopped when the interval width reaches a certain threshold. In this study, the threshold of the interval width is determined just by the interval coverage rate. The setting of the coverage rate relies heavily on business logic. Different kinds of maintenance tasks or enterprises have different requirement for the setting. In this experiment, we received help from the maintenance engineers from our cooperative enterprise and set the lower limit of coverage rate to be 60%. We also observe that, with this threshold, the prediction results become stable, which indicates that the threshold runs well.

4. Validation Results and Analysis

4.1. Experimental Settings

The experimental data for validation are the real-life demand data of spare parts from a large engineering manufacturing enterprise from China. The data set contains 75 sequences of different spare parts, in which each sequence covers 34 months from November 2018 to August 2021. The training data are the demand values in the first 33 months, while the data of the last month are for the test. The enterprise usually has one month in advance to make inventory plans and implement the allocation. Therefore, in this experiment we mainly focus on the prediction value of the last month in the whole sequence. To visualize the intermittent characteristic of the demand data, we randomly select two spare parts and illustrate their demand sequences in Figure 3.

For a fair comparison, we introduce six representative methods of time series forecasting in this experiment. These six methods are applicable for different kinds of time series. It is worth noting that BHT-ARIMA, which also adopts tensor decomposition and joint optimization strategy, can be viewed as the SOTA method for intermittent time series forecasting. See Table 1 for detailed information.



Figure 3. Example of demands for different spare parts used in this experiment. Due to commercial confidentiality requirements, only the part index, instead of its specific name, is given here.

Table 1. Six prediction methods for comparison.

Method Name	Туре	Implementation	
Croston [14] BHT-ARIMA [19]	Intermittent time series forecasting	Exponential smoothing Joint optimization with ARIMA and tensor decomposition	
ARIMA [31]	One-dimensional time series forecasting	Autoregressive modeling with moving average	
Random Forest [9] LightGBM [10]	Multidimensional time series forecasting	Ensemble prediction Decision trees using histogram algorithm	
LSTM [13]	Temporal deep learning method	Modeling long-term dependencies with memory units	

In the experiment, the parameters α and β in the Croston method are set in the range of [0.13, 0.26], and the step size is set to 0.07. The parameters p and q of BHT-ARIMA are set to 1 and 3, respectively. For ARIMA, the parameters p and q are set to 2 and 3, respectively. For Random Forest, a grid search strategy is adopted to select the optimal parameters, with the parameters determined as max_depth = 5, n_estimators = 30, learning = 0.05, and num_leaves = 20. The LSTM hidden layer is set to 20, and the learning rate is 0.001.

4.2. Result Analysis

In this experiment, the demand data of the first 33 months are used as training and the demand data of the 34th month is for the test. The predicted values and prediction intervals obtained by the proposed method are shown in Figure 4.

As suggested by the enterprise's engineers, a prediction result within a range of $\pm 30\%$ around the real values can be regarded correct. From Figure 4, our method works well when dealing with high-demand data. The reason is that, besides the unsupervised feature extraction capability of the autoencoder network, the core tensors can represent the evolutionary trend of the raw demand data well. More importantly, when the prediction results are heavily biased, the prediction intervals obtained by the proposed method can effectively cover the true value. It is clear that the prediction results by the proposed method can improve the decision-making ability of enterprises in inventory management.

Figure 5 shows the comparison of different prediction methods on the demand sequences with different degrees of intermittent distribution.



Figure 4. Prediction results of the proposed method on a total of 75 demand sequences.



Figure 5. Comparative results of different prediction methods on the demand sequences for the spare parts, respectively, indexed by (**a**) No. 0052, (**b**) No. 0392, and (**c**) No. 1437. The demands for the three spare parts are randomly selected for validation and appear different levels of intermittent distribution characteristic. The left column is the prediction results, while the right column is the raw sequence for reference.

From Figure 5, the zero-interval of the No.1437 sequence is relatively stable, so the Croston method can achieve good results. However, the Croston method performs worse than the other methods on the sequences No.0392 and No.0052 with large demand fluctuations. It indicates that Croston is only suitable for the intermittent series with a stationary zero-interval. The other conventional methods also have similar effects. These methods all have certain limitations and are only applicable to certain types of intermittent time series. Our method outperforms the other methods. The key part of our method is the introduction of tensor decomposition for smoothing anomalous demands, which is beneficial for the effective extraction of evolutionary trends from intermittent time series. With no surprise, our method shows superiority in the forecasting of intermittent time series with different distribution characteristics.

We further evaluate the effectiveness of the designed prediction interval. As stated above, a reliable prediction interval is able to provide a reference for inventory management, even if the point prediction is heavily biased or distorted. Figure 6 shows the prediction interval and predicted value of the demand sequences No. 0392, 0602, 0348, and 3082.



Figure 6. Prediction results of four parts sequences (**a**–**d**) with different intermittent distribution characteristics.

From Figure 6, the point prediction values of the sequences No. 0392 and No. 0602 are more accurate than the others. The prediction intervals also cover the real values, while the interval range remains small. On the contrary, the demand number in the sequences No. 0348 and No. 3082 is smaller, while the deviation of point prediction results is rather large. The prediction interval can cover the real value, so the enterprise can obtain precise information for the spare parts in this interval and make a more reliable plan on inventory management.

For numerical comparison, we introduce three commonly-used error metrics: MAE, RMSE, and RMSSE. The prediction errors obtained by different methods are listed in Table 2. Our method obtains the lowest prediction error in terms of all three metrics. With these results, our method is believed to provide a new reliable solution for enterprises to realize inventory management and parts scheduling.

The mainstream method in this field is the demand prediction method based on deep learning models. One advantage of the proposed method, compared to the current mainstream deep learning model, is introduction of tensor decomposition. By applying tensor decomposition, the core tensor can be extracted from the demand sequences. This effectively mitigates the impact of outliers in the demand sequences and diminishes their interference on the prediction results. Additionally, by utilizing LightGBM prediction with the core tensor, the proposed method can better capture the demand trend information in the demand sequences. The proposed method is then more suitable than deep learning methods for predicting intermittent time series with small samples.

We must point out the potential disadvantages of tensor Tucker decomposition. Actually, the process of tensor decomposition is rather computationally expensive. In the experiment, each round of tensor decomposition needs about 0.6 s, while the alternating optimization needs 3.07 s on average. The total cost of the proposed method is 30 s per round. In contrast, LSTM takes an average of 2.15 s per round, and the other shallow models like Croston, Random Forest, and ARIMA needs much less time. The high cost raises the potential risks of the proposed method for applications that require high real-time performance.

Table 2. Numerical comparison of different methods in terms of three error metrics on all 75 demand sequences.

Algorithm	MAE	RMSE	RMSSE
Random Forest	1.85	2.77	0.79
ARIMA	1.99	4.40	0.73
Croston	1.77	2.79	0.84
LightGBM	1.85	2.80	0.78
LSTM	1.74	3.20	0.91
BHT-ARIMA	1.67	2.73	0.71
Our method	1.64	2.57	0.58

5. Conclusions

In this paper, a new tensor optimization-based robust interval prediction method of intermittent time series is proposed to forecast the demand for spare parts. With the introduction of tensor decomposition, the proposed method can smooth the anomalous demand while preserving the intrinsic information of evolutionary trend from the demand sequences. Moreover, to tackle the distorted or biased prediction results of intermittent time series, the API algorithm is designed to transform traditional point prediction to adaptive interval prediction. The proposed method is able to provide a trustworthy interval for the prediction results, and can accurately reflect the uncertainty of the prediction results.

This study is fundamental to develop the effect of inventory management. As a typical extended application, the proposed method is critical to update the current safety stock model to a dynamic version by integrating the evolutionary trend of demand for spare parts. For instance, the prediction results from this paper can adjust the upper bound of safety stock to match the fluctuations in the actual demands. We believe this operation can decrease the cost level while keeping maintenance.

Another interesting issue is the correlation among the demand sequences for different spare parts. The demand for two spare parts is probably correlated due to many factors such as project cycle, climatic influence, failure probability, etc. As proven by many prior works, the correlation information between two time series is valuable. Especially for intermittent time series prediction, analyzing the correlation among some sequences is believed to improve the predictability, which is critical for intermittent time series. The joint prediction of two or more sequences is able to enhance the prediction performance in terms of accuracy and numerical stability. We plan to explore the utilization of correlation information in the future work. A feasible implementation is running adaptive clustering algorithms with profile coefficients to determine the proper sequence clusters before the prediction, which is expected to improve the predictability. We can further adopt multioutput regression or multi-task learning algorithms to achieve joint prediction on the sequences in the same cluster. We plan to explore the utilization of correlation on the

in the future work.

In our future work, transfer learning will be studied for the prediction of demand for spare parts. In actual enterprises, the demand data are usually insufficient, especially for newly deployed equipment. Considering the distinction between different types of equipment, we plan to build a transfer learning model to incorporate the evolutionary information of the demand from available equipment. The reliability of the prediction interval is also an interesting work. It is necessary to find a reliable method to converge the prediction interval into a prediction point with a high confidence level.

Author Contributions: Conceptualization, K.H. and F.L.; Methodology, W.M.; Software, X.G.; Validation, Y.R. and W.M.; Formal analysis, W.M.; Investigation, W.M.; Resources, K.H.; Data curation, W.M.; Writing—original draft preparation, X.G.; Writing—review and editing, W.M.; Visualization, Y.R.; Supervision, K.H.; Funding acquisition, K.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Key R&D Program of China under Grant 2020YFB1712105, in part by the National Natural Science Foundation of China under Grant U1704158, and in part by the Key Technology Research Development Joint Foundation of Henan Province under Grant 225101610001.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

- Bao, Y.; Wang, W.; Zou, H. SVR-based method forecasting intermittent demand for service parts inventories. In *Rough Sets, Fuzzy* Sets, Data Mining, and Granular Computing: Proceedings of the 10th International Conference, RSFDGrC 2005, Regina, SK, Canada, 31 August–3 September 2005; Proceedings, Part II 10; Springer: Berlin/Heidelberg, Germany 2005; pp. 604–613.
- Van Horenbeek, A.; Buré, J.; Cattrysse, D.; Pintelon, L.; Vansteenwegen, P. Joint maintenance and inventory optimization systems: A review. Int. J. Prod. Econ. 2013, 143, 499–508.
- 3. Moore, J.R., Jr. Forecasting and scheduling for past-model replacement parts. Manag. Sci. 1971, 18, B-200–B-213. [CrossRef]
- 4. Saaksvuori, A.; Immonen, A. Product Lifecycle Management Systems; Springer: Cham, Switerland, 2008.
- Karthikeswaren, R.; Kayathwal, K.; Dhama, G.; Arora, A. A survey on classical and deep learning based intermittent time series forecasting methods. In Proceedings of the 2021 International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, 18–22 July 2021; pp. 1–7.
- Fu, G.; Zheng, Y.; Zhou, L.; Lu, C.; Zhang, L.; Wang, X.; Wang, T. Look-ahead prediction of spindle thermal errors with on-machine measurement and the cubic exponential smoothing-unscented Kalman filtering-based temperature prediction model of the machine tools. *Measurement* 2023, 210, 112536. [CrossRef]
- Chen, Y.; Zhao, H.; Yu, L. Demand forecasting in automotive aftermarket based on ARMA model. In Proceedings of the 2010 International Conference on Management and Service Science, Wuhan, China, 24–26 August 2010; pp. 1–4.
- Karmy, J.P.; Maldonado, S. Hierarchical time series forecasting via support vector regression in the European travel retail industry. Expert Syst. Appl. 2019, 137, 59–73. [CrossRef]
- 9. Van Steenbergen, R.; Mes, M.R. Forecasting demand profiles of new products. Decis. Support Syst. 2020, 139, 113401. [CrossRef]
- Suenaga, D.; Takase, Y.; Abe, T.; Orita, G.; Ando, S. Prediction accuracy of Random Forest, XGBoost, LightGBM, and artificial neural network for shear resistance of post-installed anchors. In *Structures*; Elsevier: Amsterdam, The Netherlands, 2023; Volume 50; pp. 1252–1263.
- Cao, Y.; Gui, L. Multi-step wind power forecasting model using LSTM networks, similar time series and LightGBM. In Proceedings of the 2018 5th International Conference on Systems and Informatics (ICSAI), Nanjing, China, 10–12 November 2018; pp. 192–197.
- 12. Cao, D.; Chan, M.; Ng, S. Modeling and Forecasting of nanoFeCu Treated Sewage Quality Using Recurrent Neural Network (RNN). *Computation* **2023**, *11*, 39. [CrossRef]
- Abbasimehr, H.; Shabani, M.; Yousefi, M. An optimized model using LSTM network for demand forecasting. *Comput. Ind. Eng.* 2020, 143, 106435. [CrossRef]
- 14. Croston, J.D. Forecasting and stock control for intermittent demands. J. Oper. Res. Soc. 1972, 23, 289–303. [CrossRef]
- 15. Syntetos, A.A.; Boylan, J.E. The accuracy of intermittent demand estimates. *Int. J. Forecast.* 2005, 21, 303–314. [CrossRef]
- 16. Mor, R.S.; Nagar, J.; Bhardwaj, A. A comparative study of forecasting methods for sporadic demand in an auto service station. *Int. J. Bus. Forecast. Mark. Intell.* **2019**, *5*, 56–70. [CrossRef]

- Fu, W.; Chien, C.F.; Lin, Z.H. A hybrid forecasting framework with neural network and time-series method for intermittent demand in semiconductor supply chain. In *Advances in Production Management Systems. Smart Manufacturing for Industry 4.0, Proceedings of the IFIP WG 5.7 International Conference, APMS 2018, Seoul, Republic of Korea, 26–30 August 2018;* Proceedings, Part II; Springer: Cham, Switerland 2018; pp. 65–72.
- Xu, S.; Chan, H.K.; Ch' ng, E.; Tan, K.H. A comparison of forecasting methods for medical device demand using trend-based clustering scheme. J. Data Inf. Manag. 2020, 2, 85–94. [CrossRef]
- Shi, Q.; Yin, J.; Cai, J.; Cichocki, A.; Yokota, T.; Chen, L.; Yuan, M.; Zeng, J. Block Hankel tensor ARIMA for multiple short time series forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34; pp. 5758–5766.
- 20. Zhou, X.Y.; Tang, T.; Zhang, S.Q.; Cui, Y.T. Missing Information Reconstruction for Multi-aspect SAR Image Occlusion. J. Signal Process. 2021, 37, 1569–1580.
- 21. Guo, J.; Xie, Z.; Qin, Y.; Jia, L.; Wang, Y. Short-Term Abnormal Passenger Flow Prediction Based on the Fusion of SVR and LSTM. *IEEE Access* 2019, *7*, 42946–42955. [CrossRef]
- 22. Liu, P.; Ming, W.; Huang, C. Intelligent modeling of abnormal demand forecasting for medical consumables in smart city. *Environ. Technol. Innov.* **2020**, 20, 101069. [CrossRef]
- Li, Y.; Wang, X.; Sun, S.; Ma, X.; Lu, G. Forecasting short-term subway passenger flow under special events scenarios using multiscale radial basis function networks. *Transp. Res. Part C Emerg. Technol.* 2017, 77, 306–328. [CrossRef]
- Nguyen, H.D.; Tran, K.P.; Thomassey, S.; Hamad, M. Forecasting and Anomaly Detection approaches using LSTM and LSTM Autoencoder techniques with the applications in supply chain management. *Int. J. Inf. Manag.* 2021, 57, 102282. [CrossRef]
- 25. Li, B. Urban rail transit normal and abnormal short-term passenger flow forecasting method. *J. Transp. Syst. Eng. Inf. Technol.* **2017**, *17*, 127.
- Yokota, T.; Hontani, H. Tensor completion with shift-invariant cosine bases. In Proceedings of the 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Honolulu, HI, USA, 12–15 November 2018; pp. 1325–1333.
- 27. Courrieu, P. Fast computation of Moore-Penrose inverse matrices. arXiv 2008, arXiv:0804.4809.
- Yokota, T.; Erem, B.; Guler, S.; Warfield, S.K.; Hontani, H. Missing slice recovery for tensors using a low-rank model in embedded space. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8251–8259.
- Mao, W.; Liu, J.; Chen, J.; Liang, X. An Interpretable Deep Transfer Learning-based Remaining Useful Life Prediction Approach for Bearings with Selective Degradation Knowledge Fusion. *IEEE Trans. Instrum. Meas.* 2022, *71*, 3508616. [CrossRef]
- 30. Comon, P.; Luciani, X.; De Almeida, A.L. Tensor decompositions, alternating least squares and other tales. *J. Chemom. J. Chemom. Soc.* **2009**, *23*, 393–405. [CrossRef]
- Box, G.E.; Jenkins, G.M.; Reinsel, G.C.; Ljung, G.M. *Time Series Analysis: Forecasting and Control*; John Wiley & Sons: Hoboken, NJ, USA, 2015.

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.