sensors

MDPI

Article

Machinery Prognostics and High-Dimensional Data Feature Extraction Based on a Transformer Self-Attention Transfer Network

Shilong Sun [1,2,*], Tengyi Peng [1,2] and Haodong Huang [1,2]

[1] Guangdong Key Laboratory of Intelligent Morphing Mechanisms and Adaptive Robotics, Shenzhen 518055, China; 20s053007@stu.hit.edu.cn (T.P.); 21s053030@stu.hit.edu.cn (H.H.)
[2] School of Mechanical Engineering and Automation, Harbin Institute of Technology, Shenzhen 518055, China
* Correspondence: sunshilong@hit.edu.cn

Abstract: Machinery degradation assessment can offer meaningful prognosis and health management information. Although numerous machine prediction models based on artificial intelligence have emerged in recent years, they still face a series of challenges: (1) Many models continue to rely on manual feature extraction. (2) Deep learning models still struggle with long sequence prediction tasks. (3) Health indicators are inefficient for remaining useful life (RUL) prediction with cross-operational environments when dealing with high-dimensional datasets as inputs. This research proposes a health indicator construction methodology based on a transformer self-attention transfer network (TSTN). This methodology can directly deal with the high-dimensional raw dataset and keep all the information without missing when the signals are taken as the input of the diagnosis and prognosis model. First, we design an encoder with a long-term and short-term self-attention mechanism to capture crucial time-varying information from a high-dimensional dataset. Second, we propose an estimator that can map the embedding from the encoder output to the estimated degradation trends. Then, we present a domain discriminator to extract invariant features from different machine operating conditions. Case studies were carried out using the FEMTO-ST bearing dataset, and the Monte Carlo method was employed for RUL prediction during the degradation process. When compared to other established techniques such as the RNN-based RUL prediction method, convolutional LSTM network, Bi-directional LSTM network with attention mechanism, and the traditional RUL prediction method based on vibration frequency anomaly detection and survival time ratio, our proposed TSTN method demonstrates superior RUL prediction accuracy with a notable SCORE of 0.4017. These results underscore the significant advantages and potential of the TSTN approach over other state-of-the-art techniques.

Keywords: feature extraction; prognostics; self-attention transfer network; high-dimensional data; remaining useful life prediction

check for updates

Citation: Sun, S.; Peng, T.; Huang, H. Machinery Prognostics and High-Dimensional Data Feature Extraction Based on a Transformer Self-Attention Transfer Network. *Sensors* **2023**, *23*, 9190. https://doi.org/10.3390/s23229190

Academic Editor: Jongmyon Kim

Received: 8 October 2023
Revised: 31 October 2023
Accepted: 7 November 2023
Published: 15 November 2023

1. Introduction

Machine condition prognostics is the critical part of an intelligent health management (PHM) system, which aims to predict a machine's remaining useful life (RUL) based on condition monitoring information [1]. The general PHM procedures include the construction of health indicators (HIs) and RUL prediction. The HI is a crucial variable that indicates the current machine health condition, and also it represents the information extracted from sensor data and provides degradation trends for RUL prediction.

The HI construction process is called data fusion and has three categories: feature-level, decision-level, and data-level fusion [2]. Feature-level fusion methods rely on prior knowledge of degradation mechanisms and physical models. Ma [3] reported a multiple-view feature fusion method for predicting the RUL of lithium-ion batteries (LiBs). Decision-level techniques fuse high-level decisions based on individual sensor data and do not depend on raw-signal feature extraction. Lupea [4] developed a system utilizing features from vibration signals to detect mounting defects on a rotating test rig, with the quadratic SVM classifier emerging as the top performer Wei [5] proposed a decision-level data fusion method to map a unique sensor signal onto reliable data to improve the capability of the quality control system in additive manufacturing and RUL estimation for aircraft engines. Data-level fusion methods find the embedding feature suitable for a task from raw data. They can monitor the machine system state based on the requirements of an effective aero-engine prognostic and also the monitoring task has strong versatility. Chen [6] proposed an improved HI fusion method for generating a degradation tendency tracking strategy to predict the gear's RUL. Wang [7] extended the extreme learning machine to an interpretable neural network structure, which can automatically localize informative frequency bands and construct HI for machine condition monitoring. RUL prediction reveals the remaining operating time before equipment requires maintenance. They can be classified into four categories: physics model-based, statistical model-based, artificial intelligence-based, and hybrid methods [8]. Many recent studies have focused on artificial intelligence-based machine RUL prediction methods such as convolutional neural networks (CNNs) [9], long short-term memory (LSTM) recurrent networks [10], and gated recurrent (GRU) networks [11]. Recurrent neural networks (RNNs) have gradually become the most popular of these methods. Many scholars have focused on LSTM recurrent networks and GRU networks to address the vanishing gradient problem. Xiang [12] added an attention mechanism to the basis of an ordered, updated LSTM network, which further improved the robustness and accuracy of the LSTM network-based RUL prediction model.

Although these methods can achieve an effective machine prognostic, most artificial intelligent-based models rely on manual feature extraction (HI construction). Manual feature extraction inevitably leads to information loss, which has a negative influence on prognostics. Several studies have focused on allowing neural networks to extract features automatically from the original input, a procedure that can avoid input information loss from manual feature extraction. In the fault diagnosis field, artificial intelligence-based models exhibit excellent fault diagnosis performance with the original vibration signal input [13]. Ambrożkiewicz [14] presented an intelligent approach to detect the radial internal clearance values of rolling bearings by analyzing short-time intervals and calculating selected indicators, later enhancing classification accuracy using Variational Mode Decomposition (VMD). They can directly extract disguisable fault features from unlabeled vibration signals [15]. These methods mainly utilize CNNs to realize automatic feature extraction. Therefore, several researchers have attempted to utilize CNNs to extract degradation features for predictive purposes. Xu [16] applied a dilated CNN to the field of prognostics, used five convolutional layers to extract features from the original signal, and combined them with a fully connected network to realize effective prognostics. Li [17] proposed a multivariable machine predictive method based on a deep convolutional network. The proposed method uses the time-window method to construct 2D data as convolutional network input. Ren [18] built a spectrum principal energy vector from a raw vibration signal as a CNN input for bearing prognostics. CNNs demonstrate a strong capability in high-dimensional input situations but are not good at dealing with long-term series prognostics tasks. RNNs can easily construct long-term relationships but cannot directly utilize the abundant long-term information owing to their limited in-network processing capacity. Thus, this study proposes building a network that can directly deal with high-dimensional, long-term, time-series data for machine prognostics. The aim was to establish the long-term degradation relationship for prognostics from a large amount of raw data without relying on manual feature extraction and HI construction.

Another non-negligible defect of the existing prognostics methods is that all degradation datasets satisfy independent and identically distributed conditions. Due to the operating condition and fault type variation, a distribution discrepancy generally exists between degradation datasets (each degradation dataset is an independent domain), leading to performance fluctuation in prognostics methods. Hadi [19] introduced two automated machine-learning models aimed at precisely identifying various ball-bearing faults. Using the CWRU bearing faults dataset for evaluation, their study emphasized the potential of AutoML techniques in IIoT applications, especially valuable for industries where unscheduled downtimes can be costly. Transfer learning (TL) is introduced to help artificial intelligence-based prognostics methods extract domain-varied features and achieve effective outcomes under cross-operating conditions. TL can utilize the knowledge learned in previous tasks for new tasks by removing the domain invariance feature [20], which is widely used in fault-diagnosis tasks. In recent years, many researchers have focused on TL application in the prognostics field to achieve effective cross-operating condition prognostics. For example, Wen [21] utilized a domain adversarial neural network structure to solve the crossing domain prognostic problem. Roberto [22] proposed a domain adversarial LSTM neural network that achieved an effective aero-engine prognosis. Mao [23] performed a transfer component analysis that sequentially adjusts the features of current testing bearings from auxiliary bearings to enhance prognostics accuracy and numerical stability. This study introduces TL to extract the general representation of bearing degradation data from different operating conditions and the final fault types to achieve prognostics in cross-operating conditions. Figure 1 shows a general transfer learning algorithm for the cross-operating conditions' HIs.

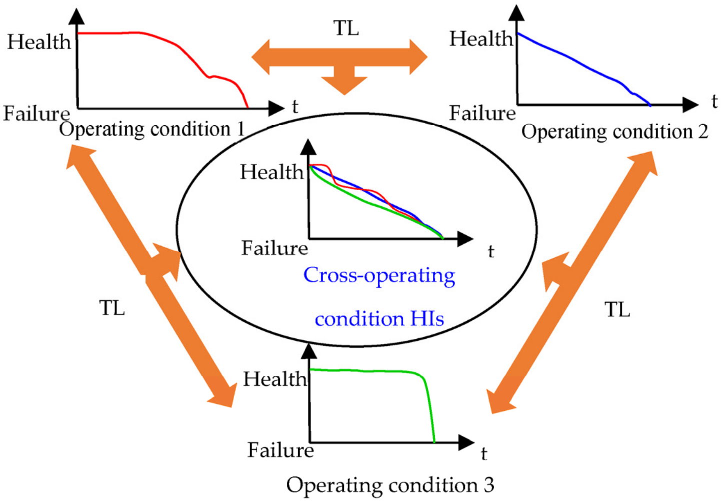

Figure 1. Transfer learning for cross-operating condition HIs construction.

Transformer [24] is a popular multi-modal universal architecture neural network architecture. The transformer utilizes a self-attention mechanism to capture the long-term dependence (spatial dependence) information between input elements in a sequence. It uses the full sequence input for each inference; therefore, it is less affected by the sequence length than traditional methods (RNN and LSTM). This feature of the transformer network is suitable for the prognostic task. Zhang [25] proposed a dual-aspect transformer network to fuse the time steps and sensor information for long-time machine prognostic. Su [26] proposed a bearing prognostic method consisting of a transformer and LSTM, achieving effective RUL prediction. Thanks to the advantages of the transformer architecture in processing long series and high-dimensional features, it has the potential to become a well-data-driven prognostic tool. Therefore, the cross-domain prognostic based on a transformer architecture is studied.

To address the limitations introduced by the above issues concerning feature extraction, cross-operating conditions, and different data distributions, this study takes the FEMTO-ST bearing dataset as an example to explore the degradation process based on a transformer-based self-attention transfer learning network (TSTN). The method can automatically construct an HI from high-dimensional feature inputs and realize long-term information association to monitor machine conditions. The innovations and contributions of this study are summarized as follows:

(1) Development of TSTN for Machine Prognostics:

We have introduced the Transformer-Based Self-Attention Transfer Learning Network (TSTN) as a dedicated solution for machine prognostics. TSTN leverages long-term, high-dimensional spectrum vectors as its input and directly produces a linear Health Index (HI) output, a numerical value ranging from 0 to 1. This HI value is straightforwardly compared to a failure threshold of 1. The core transformer architecture within TSTN plays a pivotal role in extracting critical features from extended time sequences.

(2) Incorporation of Long-term and Short-term Self-Attention Mechanisms:

TSTN incorporates both long-term and short-term self-attention mechanisms, empowering it to discern short-term and long-term fluctuations in machine conditions. By analyzing historical high-dimensional feature data in conjunction with current information, TSTN excels at identifying evolving machine states.

(3) Integration of Domain Adversarial Network (DAN) in TSTN:

To enhance TSTN's robustness and versatility, we have integrated a Domain Adversarial Network (DAN) within its architecture. DAN effectively minimizes data disparities across various operational conditions, thus enabling TSTN to monitor machine states consistently across different scenarios and environments. This integration significantly extends TSTN's applicability for cross-operation machine state monitoring.

The remainder of this paper is organized as follows. Section 2 introduces the preliminaries of the proposed method. The principle of the proposed algorithm is presented in Section 3. Section 4 describes the proposed model's experimental study, and Section 5 summarizes this work.

2. The Related Work

This section reviews the basic architecture of the transformer network structure and adversarial domain structure.

2.1. Transformer Network Structure

Vaswani proposed a transformer network structure [24]. This network is used to solve the shortcomings of the sequential computation network; that is, the number of operations required to relate signals from two arbitrary input positions increases with the distance between positions. The critical part of the transformer is the self-attention layer, which consists of two sub-parts: the multi-head attention layer and the feedforward network (FFN). The structure of the self-attention layer is illustrated in Figure 2.

The critical operation of the self-attention layer is scaled dot-product attention (right side of Figure 2).

Assuming that the input data $\mathbf{X}$ consists of n patches, the $i-th$ patch is denoted as $\mathbf{x}_i$, and the corresponding "query" ($\mathbf{q} \in \mathbb{R}^{1\times d_{\text{model}}}$), "keys" ($\mathbf{k} \in \mathbb{R}^{1\times d_{\text{model}}}$), and "values" ($\mathbf{v} \in \mathbb{R}^{1\times d_{\text{model}}}$) can be calculated through linear mapping ($\mathbf{q}_i = \mathbf{W}_Q \times \mathbf{x}_i{}^T$, $\mathbf{k}_i = \mathbf{W}_K \times \mathbf{x}_i{}^T$, $\mathbf{v}_i = \mathbf{W}_V \times \mathbf{x}_i{}^T$).

In addition, $\mathbf{W}_Q \in \mathbb{R}^{d_{\text{model}}\times d_{\text{patch}}}$, $\mathbf{W}_K \in \mathbb{R}^{d_{\text{model}}\times d_{\text{patch}}}$, and $\mathbf{W}_V \in \mathbb{R}^{d_{\text{model}}\times d_{\text{patch}}}$ were trainable variables.

To improve the learning capability of the self-attention layer, $\mathbf{k}$, $\mathbf{v}$, and $\mathbf{q}$ are linearly projected h times, which is called the multi-head attention layer. For example, $\mathbf{q}_i$ is decomposed into $\left[\mathbf{q}_{i,1}, \mathbf{q}_{i,2}, \cdots, \mathbf{q}_{i,h}\right]$, and the operations of $\mathbf{k}_i$ and $\mathbf{v}_i$ are similar to those of $\mathbf{q}_i$. $j-th$ sub-parts of $\mathbf{q}_i$, $\mathbf{k}_i$, and $\mathbf{v}_i$ are denoted as $\mathbf{q}_{i,j}$, $\mathbf{k}_{i,j}$, and $\mathbf{v}_{i,j}$, respectively. The scaled dot-product attention operation is

$$Head_{i,j} = att(\mathbf{q}_{i,j}, \mathbf{k}_{\text{dot},j}, \mathbf{v}_{i,j}) \triangleq \text{softmax}\left(\frac{\mathbf{q}_{i,j}\mathbf{k}_{\text{dot},j}{}^T}{\sqrt{d_k}}\right)\mathbf{v}_{i,j}, \quad (1)$$

where $\mathbf{k}_{\text{dot},j}$ refers to all $\mathbf{k}_{i,j}$ that must be calculated via the scaled dot-product attention operation. After the scaled dot-product attention operation, the output results of the multi-head attention layer are

$$MultiHead_i = \text{Concat}(Head_1, Head_2, \cdots, Head_h)\mathbf{W}^O, \tag{2}$$

where $\mathbf{W}^O \in \mathbb{R}^{d_{\text{patch}} \times d_{\text{model}}}$ represents the learnable linear projection. To facilitate expression, the operations (1) and (2) are summarized into one operation symbol $SM(\mathbf{q}_{i,j}, \mathbf{k}_{\text{dot},j}, \mathbf{v}_{i,j})$. FFN consists of one hidden layer, and the density of the hidden layer is denoted as d_{diff}; the density of the output layer is d_{model}.

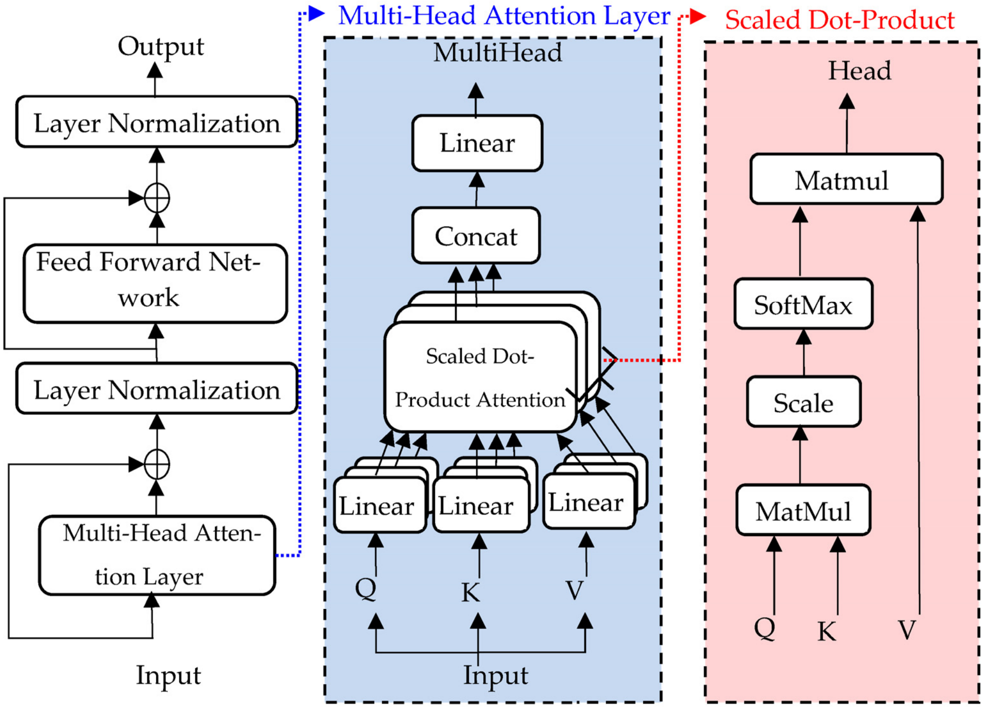

Figure 2. Details of the self-attention layer network structure.

2.2. Domain Adversarial Network

An adversarial domain network (DAN) is an effective TL method that can extract domain-invariant features [27], and its architecture is shown in Figure 3. The DAN introduces adversarial learning to achieve domain adaptation. In addition to the standard feed-forward feature extractor and label predictor, the DAN contains a domain classifier that connects to the feature extractor via a gradient reversal layer. During backpropagation-based training, the gradient reversal layer multiplies the gradient by a certain negative constant. The training process must minimize label prediction and domain classification losses. The feature distributions of all domains were similar to those of the domain classifier and the gradient reversal layer.

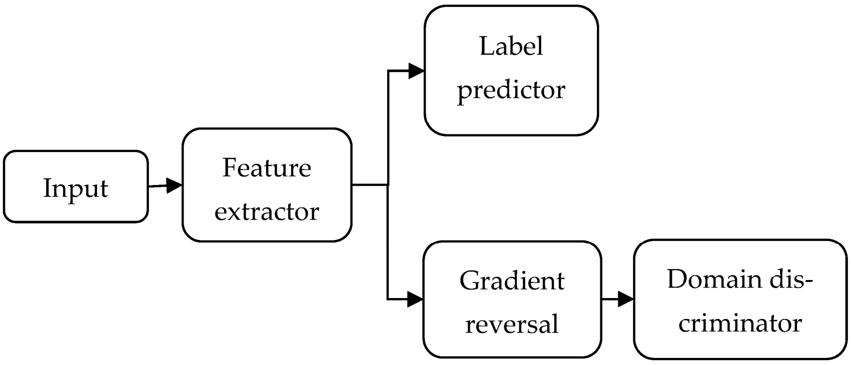

Figure 3. The architecture of an adversarial domain network.

3. The Proposed TSTN

3.1. TSTN Structure

The proposed network structure for machine RUL prediction based on the transformer and multiple-source domain adaptation is shown in Figure 4. The proposed network consists of three subparts: an encoder, HI estimator, and a domain discriminator.

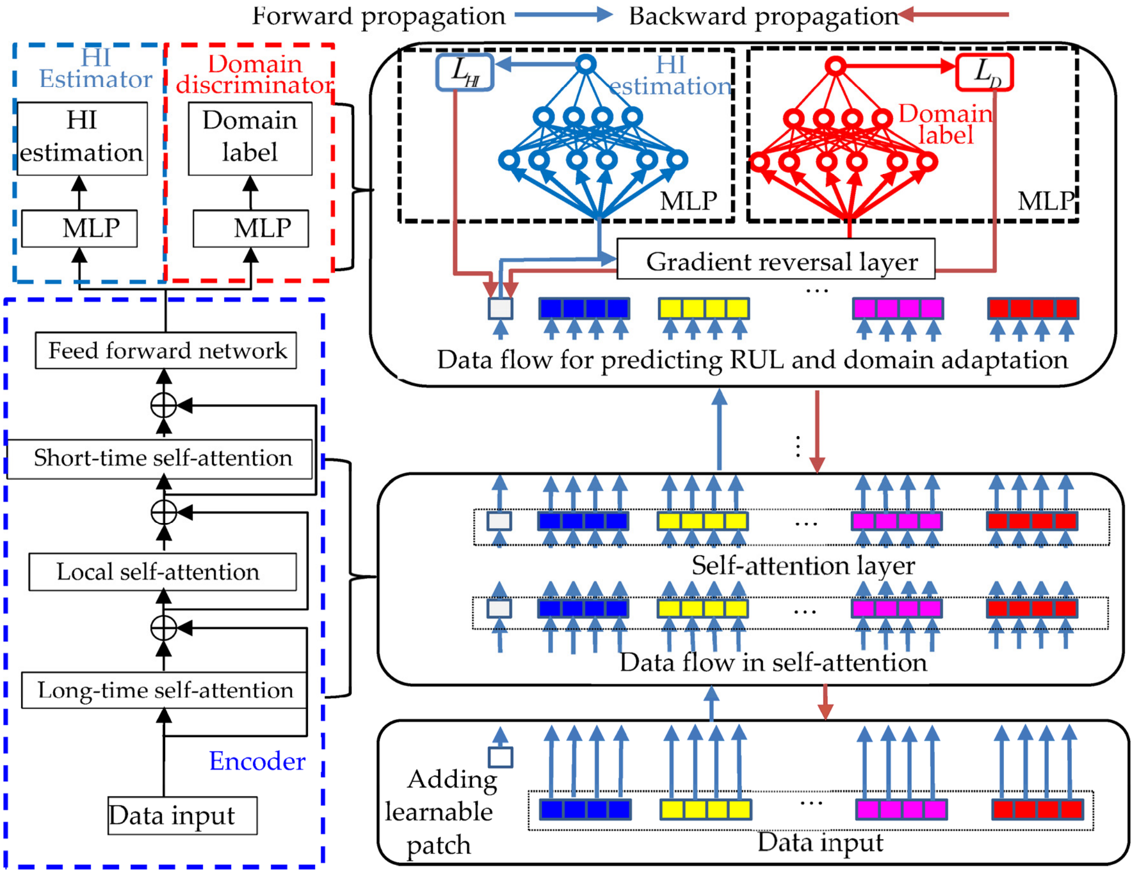

Figure 4. The whole flowchart of the proposed TSTN methodology.

The input data of this network is $\mathbf{x}_t$. When data $\mathbf{x}_t \in \mathbb{R}^{(m\times n)\times p}$ are fed into the network p, a learnable patch $\mathbf{x}_0$ is added in front of vector $\mathbf{x}_t$ and multiplies this vector $\sqrt{p}$. The input sequence is $\mathbf{X} \in \mathbb{R}^{(1+m\times n)\times p}$. The learnable patch on the encoder output serves as the HI representation, connecting the HI estimator and domain discriminator. Learnable patches calculate self-attention with others to capture the long-term collected signal sequence's high-dimensional feature (spectrum) change. The encoder of the proposed TSTN consists of a local, long-term, and short-term self-attention layer and a feed-forward network. For the ease of expression, $\mathbf{H}_{\text{input}} \in \mathbb{R}^{(1+m\times n)\times p}$ and $\mathbf{H}_{\text{output}} \in \mathbb{R}^{(1+m\times n)\times p}$ are denoted as encoder input and output, respectively.

It is well known that the datasets collected from different operating conditions and fault types are challenging in terms of satisfying the independent identically distribution (IID) property. Hence, this proposed method introduces a domain discriminator with a gradient reversal layer to make the HI representation distribution of different degradation datasets similar. This method can realize prognostics under cross-operating conditions. The encoder, HI estimator, and domain discriminator are introduced as follows. The detailed network settings are listed in Figure 4. In the training process, the forward data flow is plotted using blue arrows, and the backward gradient flow is plotted using orange arrows. The functions L_{HI} and L_d were added directly as $L = L_{HI} + L_d$ in the training process. Figure 4 displays the parameter setting of the proposed TSTN methodology.

- **Query–key–value computation.** The encoder input $\mathbf{H}_{\text{input}}$ consists of $1 + m \times n$ patches. The $l - th$ patch collected in the $s - th$ frame is $\mathbf{H}_{\text{input}}$ denoted as $\mathbf{h}_{s,l}$, and the query, key, and value vectors are t_{index} computed by $\mathbf{q}_{s,l} = \mathbf{W}_Q \times \mathbf{h}_{s,l}{}^T$, $\mathbf{k}_{s,l} = \mathbf{W}_K \times \mathbf{h}_{s,l}{}^T$, and $\mathbf{v}_{s,l} = \mathbf{W}_V \times \mathbf{h}_{s,l}{}^T$, respectively. Following the extended derivation in [28], denoting the $s - th$ frame corresponding time is t_{index}, and the rotary position embedding in the proposed method as follows:

$$\mathbf{q}^R_{s,l} = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ \vdots \\ q_{d_{\text{model}-1}} \\ q_{d_{\text{model}}} \end{pmatrix} \otimes \begin{pmatrix} \cos t_{index}\theta_1 \\ \cos t_{index}\theta_1 \\ \cos l\theta_1 \\ \cos l\theta_1 \\ \vdots \\ \cos l\theta_{p/4} \\ \cos l\theta_{p/4} \end{pmatrix} + \begin{pmatrix} q_2 \\ q_1 \\ q_4 \\ q_3 \\ \vdots \\ q_{d_{\text{model}}} \\ q_{d_{\text{model}-1}} \end{pmatrix} \otimes \begin{pmatrix} -\sin t_{index}\theta_1 \\ \sin t_{index}\theta_1 \\ -\sin l\theta_1 \\ \sin l\theta_1 \\ \vdots \\ -\sin l\theta_{p/4} \\ \sin l\theta_{p/2} \end{pmatrix}, \tag{3}$$

The predefined parameter is $\Theta = \left\{\theta_j = 10000^{-4(j-1)/d_{\text{model}}}, j \in [1, 2, \cdots, d_{\text{model}}/4]\right\}$, and the calculation operation of $\mathbf{k}^R_{s,l}$ is similar to that in (3). Using this position embedding method, the signal collected time information t_{index} and the spectrum location information l of patch $\mathbf{h}_{s,l}$ can be recognized using self-attention. The first learnable patch $\mathbf{h}_{0,0}$ needs the use of the same method to generate $\mathbf{q}^R_{0,0}$, $\mathbf{k}^R_{0,0}$, and $\mathbf{v}_{0,0}$. Since the time-embedding information offers the time auxiliary information, private over-fitting t_{index} will time a random value governed by $N(1, 0.003)$.

- **Long-term, local, and short-term self-attention.** The dimensions of the input data $\mathbf{x}_t$ are enormous. The number of calculations is large when self-attention is calculated for each patch, thereby confusing the network. We propose three sub-self-attention parts to allow the network to capture the degradation trend from the high-dimensional spectrum: local, long-term, and short-term self-attention.

To trace the long-term trend of machine conditions, we compute it by comparing each patch with all patches at the same spectrum location.

$$\mathbf{a}^{R\,(\text{Long-term})}_{s,l} = SM(\mathbf{q}^R_{s,l}, \left[\mathbf{k}^R_{0,0}, \left\{\mathbf{k}^R_{i,l}\right\}_{i=1,\cdots,m}\right], \mathbf{v}_{s,l}). \tag{4}$$

To learn the spectrum information from each collected signal, local self-attention operation only computes patches with the others collected simultaneously. The local self-attention operation is

$$\mathbf{a}^{R\,(\text{Local})}_{s,l} = SM(\mathbf{q}^R_{s,l}, \left[\mathbf{k}^R_{0,0}, \left\{\mathbf{k}^R_{s,i}\right\}_{i=1,\cdots,n}\right], \mathbf{v}_{s,l}). \tag{5}$$

The rapid, short-term changes in the machine conditions can be computed as follows:

$$\mathbf{a}_{s,l}^{R\,(\text{Short}-\text{term})} = SM(\mathbf{q}_{s,l}^{R}, \left[\mathbf{k}_{0,0}^{R}, \left\{\mathbf{k}_{i,j}^{R}\right\}_{i=1,\cdots,s;j=1,\cdots,n}\right], \mathbf{v}_{s,l}), \tag{6}$$

where s denotes the first s frame on which we wish to focus. After calculating all patches $\mathbf{H}_{\text{input}}$ via a self-attention operation, the output is represented as $\mathbf{A}$.

- **Residual connection and layer normalization.** After the self-attention computation, the output of the attention layer is calculated via the $\mathbf{B} = LayerNorm\left(\mathbf{A} + \mathbf{H}_{\text{input}}\right)$ residual connection [29] and layer normalization [30].
- **FFN and layer normalization.** The final layer of the encoder is the FFN and layer normalization; that is, $\mathbf{H}_{\text{output}} = LayerNorm(\mathbf{B} + FFN(\mathbf{B}))$.

The feed-forward layer consists of an MLP with one hidden layer. The density of the hidden layer is denoted by $d_{\text{diff}} = 8\ p$, and the density of the output layer is denoted by p. Notably, the activation function of the hidden layer is GeGLU [31], and the output layer has no activation function. GeGLU introduced gates to modulate the linear projection, which can control the information that is not conducive to HI estimation passed on to the encoder.

Subsequently, all operations in $\mathbf{H}_{\text{input}}$ are encoder outputs. To combine the long-term, local, and short-term self-attention into one encoder, $\mathbf{B}^{(\text{Long}-\text{term})}$ is fed back to calculate the local self-attention instead of being passed to the FFN. Hence, the new $\mathbf{Q}^{R}, \mathbf{K}^{R}$ and $\mathbf{V}$ are generated from $\mathbf{B}^{(\text{Long}-\text{term})}$ and fed into Equation (5) to calculate local self-attention. The operation of short-term self-attention was similar to that of local self-attention.

- **HI estimator.** An MLP with one hidden layer was connected to the learnable patch of the encoder output, and the MLP output was the HI estimated result e_{HI}.

To indicate HI easily and intuitively, the training label is defined by the index results from the normalized operating time t divided by the machine system operating time T, $label_{HI,t} = t/T$. Assuming that G datasets are required in the training process, the loss function $L^{g}{}_{HI}$ from the $g-th$ training dataset is the mean squared error of $e_{HI,t}$ and $label_{HI,t}$. The naive average induces label imbalance because the length of the dataset varies. An adaptive weighting scheme [32] is introduced to avoid the label imbalance problem, and the formula is

$$L_{HI} = \sum_{g=1}^{G} \exp(L^{g}{}_{HI})L^{g}{}_{HI} \Big/ \sum_{g=1}^{G} \exp(L^{g}{}_{HI}). \tag{7}$$

- **Domain discriminator.** The domain discriminator consisted of an MLP with one hidden layer connected to the learnable patch of the encoder output. The number of domain discriminators is equal to the number of degradation-process datasets. The output of each domain discriminator was a 2D vector. The second and first elements represent the current inputs sampled during the degradation process. The network learns a domain-invariant HI representation if the domain discriminator cannot differentiate the current input from the dataset.

Assuming that this network has G domain discriminators, the loss function $L^{g}{}_{D}$ of a single-domain discriminator g is based on cross-entropy loss. The same adaptive weighting scheme was applied to make domain discriminators available. A gradient reversal layer is inserted between the domain discriminator and the learnable patch of the encoder output. In the forward process, the gradient reversal layer performs nothing; however, in the backward process, the gradient is multiplied by a pre-specified negative constant $-\lambda$. The pre-specified negative constant $-\lambda$ is followed by $-\lambda = -(2/(1+\exp(-10 training\_process)) - 1)$ in training, where $training\_process$ denotes the training progress linearly changing from zero to one.

Table 1 shows the network structure parameter setting of TSTN.

Table 1. Parameter setting of TSTN.

Encoder	Multi-Head	d_{model}	d_{diff}	Dropout rate
	16	64	512	0.2
HI estimator (MLP)	Layer	Dense	Activation function	number
	Fully connected	32	GeGLU	1
	Fully connected	1	GeGLU	1
Domain discriminator (MLP)	Layer	Dense	Activation function	number
	Fully connected	32	GeGLU	Equal to dataset number
	Fully connected	2	Softmax	

3.2. Data Pre-Processing

For the data pre-processing part, there are two sub-parts: signal collection and the decomposition of patches. Figure 5 displays the data pre-processing input network.

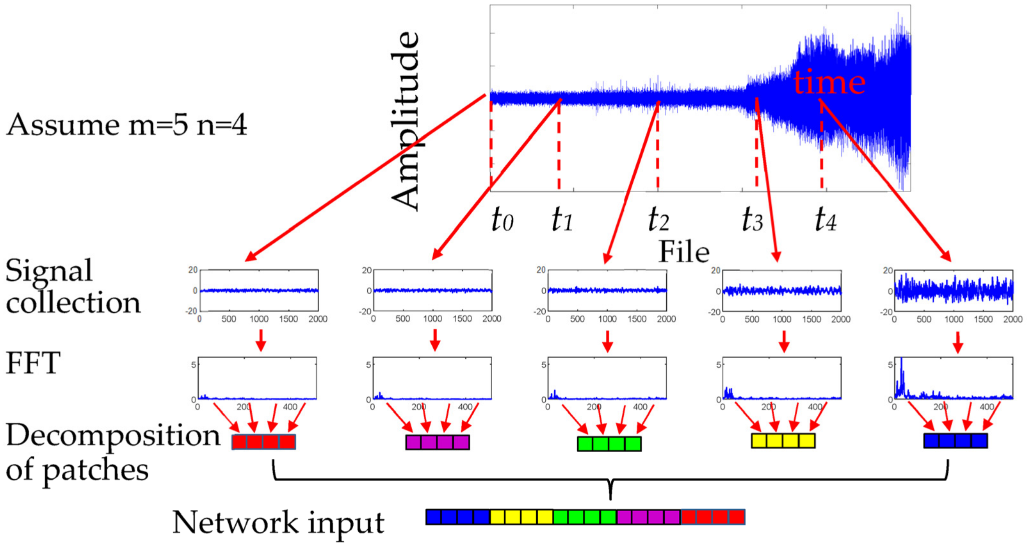

Figure 5. Data pre-processing input network.

- **Signal collection.** The input of the proposed TSTN was a clip $\mathbf{X}_t \in \mathbb{R}^{m\times 512}$ consisting of m frames with 512 spectrum features extracted from the measured vibration signal. The frames were divided according to the time to obtain abundant temporal information. The time-divided relationship follows $t_{index} = \tau \times \left[\left(\sin\left(\frac{m-1-index}{m-1}\times\pi-\frac{\pi}{2}\right)+1\right)/2\right]$, $index = (0,\ 1,\ 2,\ \cdots,\ m-2,\ m-1)$, which τ denotes the time required to collect data.
- **Decomposition of patches.** Each spectrum feature is decomposed into non-overlapping patches with a size of p; that is, $n = 512/p$. These patches are then flattened into a vector $\mathbf{X} \in \mathbb{R}^{(m\times n)\times p}$ as the network input.

In summary, the data preprocessing process can be divided into the following seven steps:

(1) Index collection: Assuming that the total length of the time series is 20 s, set parameter m = 5. The indexes for collecting data are 0, 5, 10, 15, and 20;
(2) Calculation of times: From the indexes, we can calculate the t_{index} using the index.
(3) Sampling data: Based on the calculated t_{index}, the data are sampled at these times;
(4) Fourier transform: Perform Fourier transform on the sampled time;
(5) Select data points: From the Fourier transformed data, select the first 512 points for each sampling time;

(6) Divide into blocks: Divide the selected 512 data points into 4 blocks, each with a length of 1278;
(7) Reverse concatenation: concatenate these 4 blocks in reverse order.

3.3. TSTN Training

This section mainly introduced the proposed diagnosis framework. First, the problem description is illustrated. The proposed machine monitoring methodology is based on historical data, fitting the normalized RUL label y_i (1-0) via the input features x_i. Then, the transfer task is utilized to extract the domain invariant part for cross-operation condition monitoring. The prognostics process consists of two steps: first, constructing the TSTN network based on the input spectrum feature combined with the health indicator; second, using the Monte Carlo method to predict RUL via the TSTN output HI. In this section, a TSTN is developed to predict the machine HI. Details of the proposed TSTN network are presented and shown in Figure 6. The domain discriminator of the developed TSTN was utilized only in the TL training process.

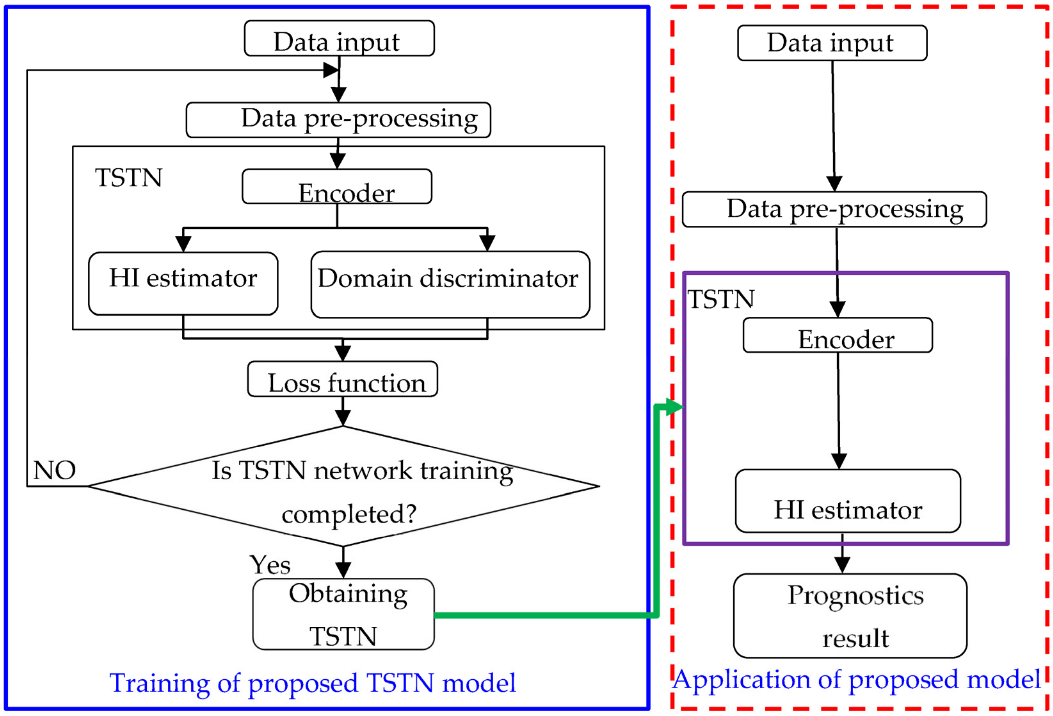

Figure 6. Flowchart of the proposed TSTN model and its application.

In actual applications, the output of the HI estimator is the machine-condition monitoring HI of the proposed framework. This study utilized the Monte Carlo method based on a linear model with exponential smoothing with parameter 0.9 to generate the downstream prognostics result.

4. Experiment Details

4.1. Training and Testing Regimes

Training regime. Stochastic gradient descent (SGD) with 0.9 momenta is the optimizer in this work. For practical training, the learning rate throughout the training varied according to the following equation:

$$\mu = \min\left(S^{-0.5},\ S \times {W_S}^{-1.5}\right) \Big/ \left(1 + 10T_p\right)^{0.75}, \tag{8}$$

where S is the number of current training steps, and $W_S = 1000$. T_p is a training process that linearly changes from 0 to 1. The batch size is set to 32, the network weights are updated with gradient accumulation during training, and the random seed is 66.

Testing regime. Once the network finishes training, the testing data are fed into the grid for testing. Apart from performing data pre-processing, other operations are not required for testing. The HI estimator output was the bearing health condition of the input data. The HI-estimated output of the proposed method is $e_{HI,t}$.

4.2. Prognostics Result

The validation dataset was obtained from the PRONOSTIA [33] experimentation platform to test and validate bearing fault detection, diagnostic, and prognostic approaches. The rig bench is presented in Figure 7. When the test rig was initialized, a file that contained a 0.1 s vibration signal with a sampling frequency of 25.6 kHz was generated and recorded every 10 s. Three operating conditions were considered; each had two training sets and several testing sets. Information on the training and testing sets is presented in Table 2. The dataset provides 6 sets of data that ran to failure for the establishment of the prediction model, which are 1-1, 1-2, 2-1, 2-2, 3-1, and 3-2. In addition, 11 datasets are provided for RUL, which are 1-3, 1-4, 1-5, 1-6, 1-7, 2-3, 2-4, 2-5, 2-6, 2-7, and 3-3.

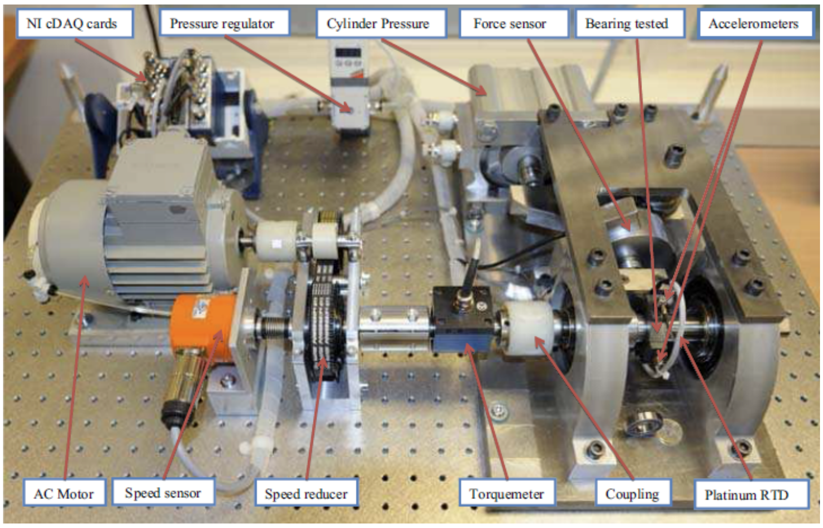

Figure 7. Overview of PRONOSTIA.

Table 2. Information on the FEMTO-ST dataset.

Operating Condition	1	2	3
Speed (rpm)	1800	1650	1500
Loading (N)	4000	4200	5000
Training dataset	1-1, 1-2	2-1, 2-2	3-1, 3-2
Testing dataset	1-3, 1-4, 1-5, 1-6, 1-7	2-3, 2-4, 2-5, 2-6, 2-7	3-3

The scoring benchmark was set according to [30], and only the vertical vibration signal (2560 points per file) was used to generate the network output. The size of the spectrum generated via fast Fourier transform was 512. The pre-processing operation entailed 21 spectrum frames, and each structure was decomposed into eight non-overlapping patches. The training epoch was set to 60. To achieve cross-domain condition monitoring in the bearing, we use six training datasets in the same training process.

After finishing the training process of the proposed network, the network can be utilized to monitor the health condition of the bearing in the testing data. The proposed method's expected output $e_{HI,t}$ is a direct HI of 0 to 1. To demonstrate the capability of the direct HI in RUL prediction, we use the Monte Carlo method based on the linear model for curve fitting and RUL prediction ($pre_{RUL,t}$).

Figure 8 shows the estimated HI results and RUL predictions from the test data of the proposed method. The blue line represents the HI output of the proposed method. The green line refers to the RUL prediction and 95% confidence interval, and the yellow area represents the probability distribution function of the RUL prediction result $pre_{RUL,t}$. As shown in Figure 8, HI estimation using the proposed method can effectively capture the bearing degradation trend. The proposed method can provide nearly linear HI estimation.

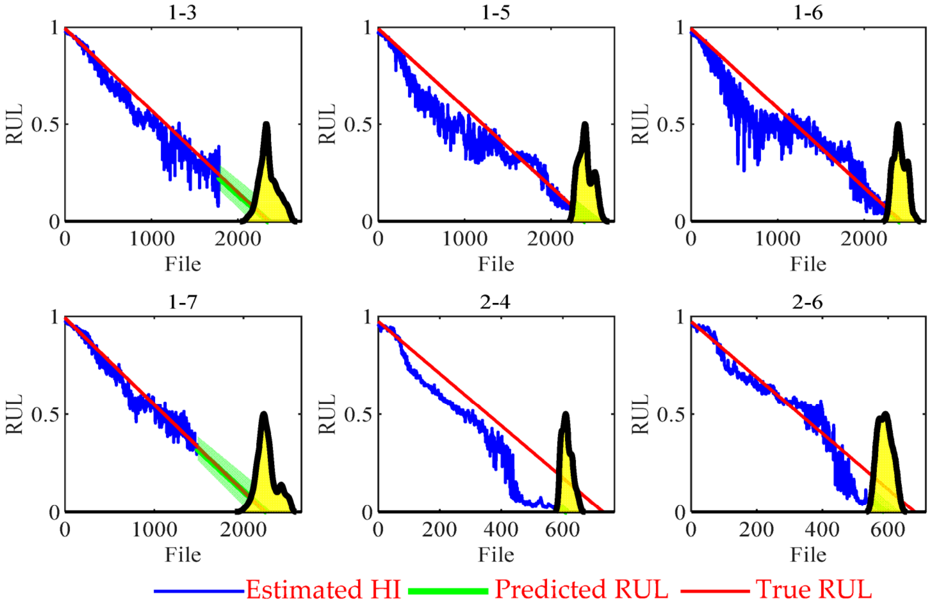

Figure 8. Estimated HIs of the proposed method. (Yellow areas represents the RUL of probability density distribution).

5. Comparisons and Analysis

Then, the normalized prediction error Er_i and benchmark scores were calculated [33]. The results of all the testing sets are listed in Table 3. The specific calculation formula is as follows:

$$\%Er_i = 100 \times \frac{ActRUL_i - \widehat{RUL_i}}{ActRUL_i} \tag{9}$$

$$A_i = \begin{cases} \exp^{-\ln(0.5)\cdot(Er_i/5)} & if\ Er_i \le 0 \\ \exp^{+\ln(0.5)\cdot(Er_i/20)} & if\ Er_i > 0 \end{cases} \tag{10}$$

$$Score = \frac{1}{11}\sum_{i=1}^{11}(A_i) \tag{11}$$

As presented in Table 3, except for testing sets 2-7 and 3-3, the RUL prediction results of the proposed method are reasonable. The errors in the prediction results of datasets 1-5 to 2-6 were shallow, and the proposed method could effectively perform bearing condition monitoring with testing sets 1-5, 1-7, 2-4, and 2-6. Compared to the RNN-based RUL prediction method [34], convolutional LSTM network [35], Bi-directional LSTM network with attention mechanism [36], and the traditional RUL prediction method based on vibration frequency anomaly detection and survival time ratio [37], the proposed TSTN method has higher RUL prediction accuracy. These results confirm that the proposed method is applicable to the prognostics of mechanical rotating components. For the last two datasets, the RUL predictions exhibit large deviations. The reason for these large deviations is that the vibration signal changes slightly only in the early degradation process, which displays a linear degradation trend. However, as time goes on, the linear trend becomes nonlinear. The HI $e_{HI,t}$ does not have a linear change rate in the latter stage. Hence, the proposed HI is unsuitable for predicting the RUL in latter-stage degradation. However, compared with other methods, the computational complexity is higher, and the training time is 3 h.

Table 3. RUL Prediction results of the proposed method.

Dataset	Er_i% (Our)	Er_i% [34]	Er_i% [37]	Er_i% [36]	Er_i% [35]
1-3	0.5	43	37	−5	55
1-4	23	67	80	−9	39
1-5	25	−22	9	22	−99
1-6	9	21	−5	18	−121
1-7	−2	17	−2	43	71
2-3	82	37	64	45	76
2-4	85	−19	10	33	20
2-5	2	54	−440	50	8
2-6	70	−13	49	26	18
2-7	−1122	−55	−317	−41	2
3-3	−1633	3	90	20	3
Score	0.4017	0.2631	0.3066	0.3198	0.3828

Discussions of the Proposed Methodology

Influence of multi-head number. To improve the learning capability of the self-attention layer of the encoder, linearly project keys, values, and query h times, which is called the multi-head attention operation. In this section, the influence of multi-head numbers is discussed. The predicted RUL benchmark scores of different multi-head numbers indicate that 16 (score is 0.4017) is the most suitable for the prognostics task, and it is higher than the results of four multi-head (score is 0.0607) and eight multi-head (score is 0.1124) numbers. Theoretically, the larger the multi-head number, the stronger the fitting capability. However, the rotary position embedding method requires almost four numbers to indicate location information. When the multi-head operation breaks up the rotary position embedding, the self-attention calculation cannot capture the time information. Therefore, the score of the 32 multi-head numbers was 0.2631, and that of the 64 multi-head numbers was 0.0689. In summary, the multi-head number needs to be set to $d_{\text{model}}/4$ in the prognostics task.

Discussions with/without transfer learning. The proposed method uses the domain discriminator with the gradient reversal layer to extract the domain-invariant RUL representation. We expect to use the TL method to improve the linearity of the estimated HI under different operating conditions. An experiment was conducted on a TSTN without a TL, reflecting the domain discriminator's effectiveness in cross-operating condition monitoring. Aside from removing the domain discriminator, the other network framework settings were similar to those in Figure 9. The RUL prediction score decreased from 0.4017 to 0.0515. The prognostic results of TSTN and TSTN without a domain discriminator for test datasets 1-6, 1-7, 2-4, and 2-6 indicate TL's effectiveness. Figure 9 shows the comparison of TSTN and TSTN without transfer learning. The blue lines represent the classical TSTN HI results, and the greenish-blue lines denote the HI-estimated effects of TSTN without TL. TL improves the TSTN prognostics capability in cross-operating condition situations.

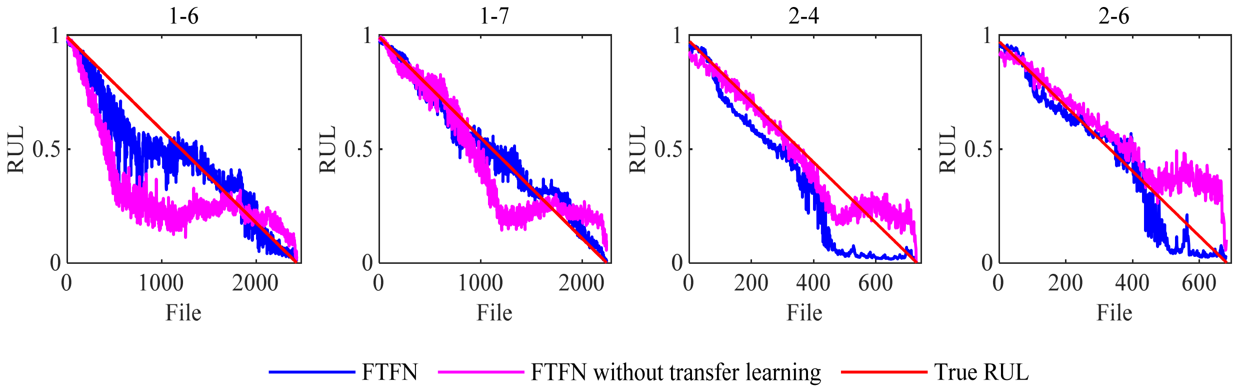

Figure 9. Comparison of TSTN and TSTN without transfer learning.

Effectiveness of the self-attention mechanism. This study utilized test sets 1-6 to generate a self-attention heatmap (shown in Figure 10) to indicate the effectiveness of the self-attention mechanism. The longitudinal of the self-attention heatmap refers to the m time frames, and the transverse of the self-attention heatmap pertains to the 16 multi-heads with eight patches. In this study, 1/3, 2/3, and 1 of the normalized operating time were selected. When a patch has a high self-attention value, the network focuses on that patch. Figure 10 shows that only a few heads undertake the HI estimation task, but our previous study indicated that a sizeable multi-head number equates to strong learning capability. A possible reason is that a large multi-head results in a flexible feature association capability, which means that features can be selected precisely.

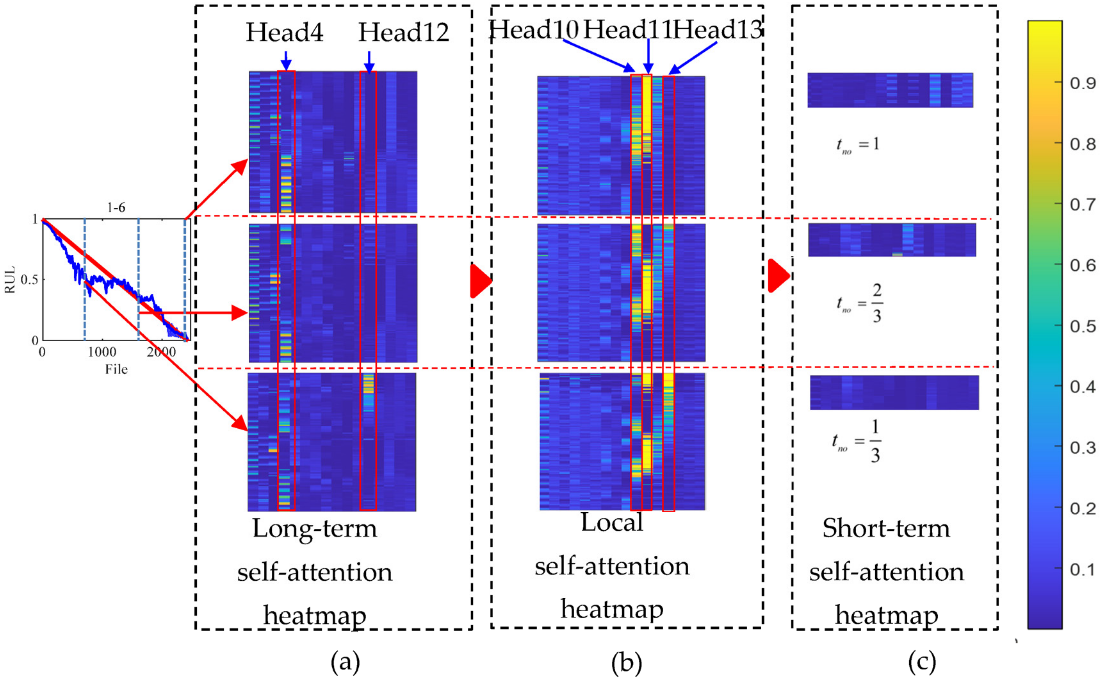

Figure 10. The long-term, local, and short-term self-attention heatmap from testing sets 1-6. (**a**) Long-term self-attention heatmap; (**b**) Local self-attention heatmap; (**c**) Short-term self-attention heatmap.

The first self-attention layer was a long-term self-attention layer. In Figure 10, head 12 of long-term self-attention captures the severe degradation at the end of the operating time, and head 4 focuses on the weak degradation at the early and middle operating stages. After the long-term self-attention layer, the spectrum long-term change relationship was obtained, and the local self-attention layer was used to capture abundant information in one frame. In Figure 10, a clear degradation relationship was captured. Head 11 of the local self-attention layer captured the weak degradation in the early operating stage. Head 10 focuses on degradation in the middle operating phase, and head 13 focuses on rapid degradation at the late operational stage. Figure 10 shows that local self-attention plays a greater role than the long-term self-attention layer. However, the learning capability sharply declined when the two layers' order was changed. This result indicates that the long-term self-attention layer generates the long-term relationship and is strengthened by the local self-attention layer.

In summary, the multi-heads in the short-term self-attention layer focus on the spectrum value, thereby making the proposed TSTN sensitive to spectrum value changes.

6. Conclusions

Machine prognostics play a crucial role in the automaticity and intelligence of industrial plants, especially in intelligent plant manufacturing and asset health management. This study proposed a TSTN-based machine prognostic method to solve the HI automatic construction with a high-dimensional feature input in a cross-operating condition. The proposed method is integrated with a novel transformer network structure with a domain adversarial TL consisting of an encoder, an HI estimator, and a domain discriminator. First, the proposed TSTN automatically extracts features (HI) from a long-term high-dimensional feature input, avoiding information loss caused by manual feature extraction. Second, we have devised a self-attention mechanism that encompasses long-term, short-term, and local perspectives, enabling it to discern the dynamic interplay between long-term and short-term machine health conditions. Third, when incorporating the DAN TL method, it addresses issues of cross-operating conditions and different data distributions. The domain discriminator with a gradient reversal layer can generate an accurate and robust HI. Compared to the RUL prediction methods based on RNN, the convolutional LSTM network, the bi-directional LSTM network with an attention mechanism, and traditional strategies rooted in vibration frequency anomaly detection and survival time ratios, our proposed TSTN approach achieves a superior score of 0.417, indicating its enhanced accuracy in RUL prediction. In the future, we plan to collect more datasets to verify the effectiveness of the proposed method. In addition, we will conduct further research on improving the generalization ability of the method for dealing with extremely cross-operating conditions, such as predicting the RUL for an unseen operating condition. The proposed method is a promising methodology for coping with HI estimator construction with a high-dimensional feature input, monitoring machine health conditions, and predicting machines' RUL in cross-operating working conditions.

Author Contributions: Conceptualization, methodology, T.P.; software, T.P.; validation, S.S.; formal analysis, S.S.; writing—original draft preparation, T.P.; writing—review and editing, S.S.; funding acquisition, S.S.; H.H. revised and review. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by the Natural Science Foundation of Guangdong Province Under Grant No. 2021A1515110615.

Data Availability Statement: The data presented in this study are openly available in Ref. [33].

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

References

1. Zheng, K.; Li, T.; Su, Z.; Zhang, B. Sparse Elitist Group Lasso Denoising in Frequency Domain for Bearing Fault Diagnosis. *IEEE Trans. Ind. Inform.* **2021**, *17*, 4681–4691. [CrossRef]
2. Hall, D.L.; Llinas, J. An introduction to multisensor data fusion. *Proc. IEEE* **1997**, *85*, 6–23. [CrossRef]
3. Ma, C.; Zhai, X.; Wang, Z.; Tian, M.; Yu, Q.; Liu, L.; Liu, H.; Wang, H.; Yang, X. State of health prediction for lithium-ion batteries using multiple-view feature fusion and support vector regression ensemble. *Int. J. Mach. Learn. Cybern.* **2019**, *10*, 2269–2282. [CrossRef]

4. Lupea, I.; Lupea, M. Machine Learning Techniques for Multi-Fault Analysis and Detection on a Rotating Test Rig Using Vibration Signal. *Symmetry* **2022**, *15*, 86. [CrossRef]
5. Wei, Y.; Wu, D.; Terpenny, J. Decision-Level Data Fusion in Quality Control and Predictive Maintenance. *IEEE Trans. Autom. Sci. Eng.* **2021**, *18*, 184–194. [CrossRef]
6. Chen, X.; Wang, Y.; Sun, H.; Ruan, H.; Qin, Y.; Tang, B. A generalized degradation tendency tracking strategy for gearbox remaining useful life prediction. *Measurement* **2023**, *206*, 112313. [CrossRef]
7. Wang, D.; Chen, Y.; Shen, C.; Zhong, J.; Peng, Z.; Li, C. Fully interpretable neural network for locating resonance frequency bands for machine condition monitoring. *Mech. Syst. Signal Process.* **2022**, *168*, 108673. [CrossRef]
8. Zhu, J.; Chen, N.; Shen, C. A new data-driven transferable remaining useful life prediction approach for bearing under different working conditions. *Mech. Syst. Signal Process.* **2020**, *139*, 106602. [CrossRef]
9. Li, X.; Zhang, W.; Ding, Q. Deep learning-based remaining useful life estimation of bearings using multi-scale feature extraction. *Reliab. Eng. Syst. Saf.* **2019**, *182*, 208–218. [CrossRef]
10. An, Q.; Tao, Z.; Xu, X.; El Mansori, M.; Chen, M. A data-driven model for milling tool remaining useful life prediction with convolutional and stacked LSTM network. *Measurement* **2020**, *154*, 107461. [CrossRef]
11. Yudong, C.; Minping, J.; Peng, D.; Yifei, D. Transfer learning for remaining useful life prediction of multi-conditions bearings based on bidirectional-GRU network. *Measurement* **2021**, *178*, 109287.
12. Xiang, S.; Qin, Y.; Zhu, C.; Wang, Y.; Chen, H. LSTM networks based on attention ordered neurons for gear remaining life prediction. *ISA Trans.* **2020**, *106*, 343–354. [CrossRef] [PubMed]
13. Jia, F.; Lei, Y.; Lu, N.; Xing, S. Deep normalized convolutional neural network for imbalanced fault classification of machinery and its understanding via visualization. *Mech. Syst. Signal Process.* **2018**, *110*, 349–367. [CrossRef]
14. Ambrożkiewicz, B.; Syta, A.; Georgiadis, A.; Gassner, A.; Litak, G.; Meier, N. Intelligent Diagnostics of Radial Internal Clearance in Ball Bearings with Machine Learning Methods. *Sensors* **2023**, *23*, 5875. [CrossRef] [PubMed]
15. Peng, T.; Shen, C.; Sun, S.; Wang, D. Fault Feature Extractor Based on Bootstrap Your Own Latent and Data Augmentation Algorithm for Unlabeled Vibration Signals. *IEEE Trans. Ind. Electron.* **2022**, *69*, 9547–9555. [CrossRef]
16. Xu, X.; Wu, Q.; Li, X.; Huang, B. Dilated convolution neural network for remaining useful life prediction. *J. Comput. Inf. Sci. Eng.* **2020**, *20*, 021004. [CrossRef]
17. Li, X.; Ding, Q.; Sun, J.-Q. Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliab. Eng. Syst. Saf.* **2018**, *172*, 1–11. [CrossRef]
18. Ren, L.; Sun, Y.; Wang, H.; Zhang, L. Prediction of Bearing Remaining Useful Life With Deep Convolution Neural Network. *IEEE Access* **2018**, *6*, 13041–13049. [CrossRef]
19. Hadi, R.H.; Hady, H.N.; Hasan, A.M.; Al-Jodah, A.; Humaidi, A.J. Improved Fault Classification for Predictive Maintenance in Industrial IoT Based on AutoML: A Case Study of Ball-Bearing Faults. *Processes* **2023**, *11*, 1507. [CrossRef]
20. Sun, M.; Wang, H.; Liu, P.; Huang, S.; Wang, P.; Meng, J. Stack Autoencoder Transfer Learning Algorithm for Bearing Fault Diagnosis Based on Class Separation and Domain Fusion. *IEEE Trans. Ind. Electron.* **2022**, *69*, 3047–3058. [CrossRef]
21. Wen, B.C.; Xiao, M.Q.; Wang, X.Q.; Zhao, X.; Li, J.F.; Chen, X. Data-driven remaining useful life prediction based on domain adaptation. *PeerJ Comput. Sci.* **2021**, *7*, e690.
22. da Costa, P.R.d.O.; Akçay, A.; Zhang, Y.; Kaymak, U. Remaining useful lifetime prediction via deep domain adaptation. *Reliab. Eng. Syst. Saf.* **2020**, *195*, 106682. [CrossRef]
23. Mao, W.; He, J.; Zuo, M.J. Predicting Remaining Useful Life of Rolling Bearings Based on Deep Feature Representation and Transfer Learning. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 1594–1608. [CrossRef]
24. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
25. Zhang, Z.; Song, W.; Li, Q. Dual-Aspect Self-Attention Based on Transformer for Remaining Useful Life Prediction. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–11. [CrossRef]
26. Su, X.; Liu, H.; Tao, L.; Lu, C.; Suo, M. An end-to-end framework for remaining useful life prediction of rolling bearing based on feature pre-extraction mechanism and deep adaptive transformer model. *Comput. Ind. Eng.* **2021**, *161*, 107531. [CrossRef]
27. Ganin, Y.; Lempitsky, V. Unsupervised domain adaptation by backpropagation. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 1180–1189.
28. Su, J.; Lu, Y.; Pan, S.; Wen, B.; Liu, Y. Roformer: Enhanced transformer with rotary position embedding. *arXiv* **2021**, arXiv:2104.09864.
29. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
30. Ba, J.L.; Kiros, J.R.; Hinton, G.E. Layer normalization. *arXiv* **2016**, arXiv:1607.06450.
31. Narang, S.; Chung, H.W.; Tay, Y.; Fedus, W.; Fevry, T.; Matena, M.; Malkan, K.; Fiedel, N.; Shazeer, N.; Lan, Z. Do transformer modifications transfer across implementations and applications? *arXiv* **2021**, arXiv:2102.11972.
32. Zhu, J.; Chen, N.; Shen, C. A New Multiple Source Domain Adaptation Fault Diagnosis Method Between Different Rotating Machines. *IEEE Trans. Ind. Inform.* **2021**, *17*, 4788–4797. [CrossRef]

33. Nectoux, P.; Gouriveau, R.; Medjaher, K.; Ramasso, E.; Chebel-Morello, B.; Zerhouni, N.; Varnier, C. PRONOSTIA: An experimental platform for bearings accelerated degradation tests. In Proceedings of the IEEE International Conference on Prognostics and Health Management, PHM'12, Denver, CO, USA, 20–23 June 2012; pp. 1–8.
34. Guo, L.; Li, N.; Jia, F.; Lei, Y.; Lin, J. A recurrent neural network based health indicator for remaining useful life prediction of bearings. *Neurocomputing* **2017**, *240*, 98–109. [CrossRef]
35. Hinchi, A.Z.; Tkiouat, M. Rolling element bearing remaining useful life estimation based on a convolutional long-short-Term memory network. *Procedia Comput. Sci.* **2018**, *127*, 123–132. [CrossRef]
36. Rathore, M.S.; Harsha, S. Prognostics Analysis of Rolling Bearing Based on Bi-Directional LSTM and Attention Mechanism. *J. Fail. Anal. Prev.* **2022**, *22*, 704–723. [CrossRef]
37. Sutrisno, E.; Oh, H.; Vasan, A.S.S.; Pecht, M. Estimation of remaining useful life of ball bearings using data driven methodologies. In Proceedings of the 2012 IEEE Conference on Prognostics and Health Management, Denver, CO, USA, 18–21 June 2012; pp. 1–7.

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.