Complex system health condition estimation using tree-structured simple recurrent unit networks

Weijie Kang1 · Jiyang Xiao2 · Junjie Xue2

Received: 24 November 2021 / Accepted: 27 March 2022 / Published online: 10 May 2022
© The Author(s) 2022

Abstract
Modern production has stricter requirements for the reliability of complex systems; thus, it is meaningful to estimate the health of complex systems. A complex system has diverse observation features and complex internal structures, which have been difficult to study with regard to health condition estimation. To describe continuous and gradually changing time-based characteristics of a complex system’s health condition, this study develops a feature selection model based on the information amount and stability. Then, a reliability tree analysis model is designed according to the selected relevant features, the reliability tree is developed using expert knowledge, and the node weight is calculated by the correlation coefficient generated during the feature selection process. Using the simple recurrent unit (SRU), which is a time series machine learning algorithm that achieves a high operating efficiency, the results of the reliability tree analysis are combined to establish a tree-structure SRU (T-SRU) model for complex system health condition estimation. Finally, NASA turbofan engine data are used for verification. Results show that the proposed T-SRU model can more accurately estimate a complex system’s health condition and improve the execution efficiency of the SRU networks by approximately 46%.

Keywords Complex system · Health condition estimation · Simple recurrent unit · Feature selection

Abbreviations

AHP Analytic hierarchy process
CN Component node
CNN Convolutional neural network
FRA Fault tree analysis
GRU Gated recurrent unit
IN Information node
ISS Information-stability selection
LSTM Long short-term memory
LVW Las Vegas wrapper
RNN Recurrent neural networks
RT Reliability tree
RTA Reliability tree analysis
SHI System health index
SRU Simple recurrent unit
TIEDVD Time interval equal discharge voltage difference
TSA Tree structure analysis
T-SRU Tree-structure SRU

Introduction
With the development of health management technologies, the scope of their application scenarios has broadened. In high-precision fields such as aviation and aerospace, the complexity of key systems is increasing. Establishing a health condition estimation model for complex systems has thus become an active area of research [1]. The typical process of complex system health condition estimation, as shown in Fig. 1, includes the following two key steps [2]. First, feature processing consists of feature selection and feature organization. Due to the diverse types of complex system health condition features, it is necessary to select those features that are closely related to the health condition. This process is called feature selection, which can alleviate problems of dimensionality and improve the execution efficiency of the follow-up health condition estimation model [3]. In addition,
because the complex system contains multiple components, it is also necessary to effectively organize these large numbers and multiple types of health condition features (i.e., feature organization). Hierarchical analysis can typically be performed according to the complex system internal structures, which can effectively improve the accuracy of the health condition estimation model [4]. Second, estimation modelling includes sequence (time series) characteristics mining and estimation standard formulation. Different from fault diagnosis research, health condition estimation only considers the performance degradation process of the system, which is a continuous and gradually changing time series process. Health condition feature sequences contain a lot of time series information, which require the estimation models to be mined [5]. In addition, because complex systems typically do not have explicit health indices, their health condition is relative, and corresponding estimation standards must be formulated [6] by analysing degradation process data. However, it is difficult to formulate relatively objective and reasonable estimation standards. Thus, the complex system health condition can be estimated through feature processing and estimation models.

Because there are diverse types of complex system health condition features, if no selection is performed and all features are directly used for health condition estimation, this will cause the dimensionality problems in the estimation model and may reduce the estimation accuracy of the model. In machine learning currently, features that have a positive impact on the current learning task are called relevant features, and features that have no impact or even a negative impact on the current learning task are called irrelevant features [7]. The process of selecting relevant features is called feature selection. Commonly used feature selection models primarily include filter selection, wrapper selection, and embedding selection. The difference between the three is whether the estimation result is used as the basis for feature selection [8]. Because the complex system health condition is typically difficult to accurately describe, it is difficult to combine the estimation results for feature selection. Also, due to the gradual change in those health conditions, its relevant features should show a certain degree of stability and relative monotonicity. Therefore, the filter feature selection model, which does not need to rely on the estimation results, that is based on statistics can quickly and more directly select the relevant features of the complex system degradation process [9].

For complex system health condition estimation, it is insufficient to only select the health condition-relevant features at an appropriate scale. Due to the high coupling of the internal structure and functions of the complex system, it is also necessary to select the relevant features and organize them effectively, which means assigning a reasonable logical relationship. Only in this way can a more accurate complex system health condition estimation be achieved. Commonly used feature organization models include the analytic hierarchy process (AHP) and tree structure analysis (TSA). AHP requires a lot of expert knowledge and is highly subjective and difficult to apply directly [10]. TSA uses a tree structure relationship by analysing the internal operating mechanism of the complex system and then combines it with other machine learning algorithms to estimate the health condition of the complex system, which has good adaptability [11].

In past studies of health condition estimation, feature selection and organization are often used separately and fail to be combined organically: a sufficient number of features are typically selected, and then, a comprehensive estimation is performed; or all features are organized, and then, a machine learning algorithm is used to obtain the estimation. However, for complex systems, due to the diverse types of health condition features and the complex internal structure, feature selection and organization must be combined. Only by selecting features first and then effectively organizing the selected features can the health condition of the complex system be more accurately estimated. This study develops an information-stability selection (ISS) model for feature selection, draws on the hierarchical analysis idea of fault trees, and then establishes a reliability tree analysis (RTA) model to effectively organize relevant features.

In previous studies, artificial neural networks [12], coupled neurons [13], adaptive particle filters [14] and other data-driven health estimation models used have a common mathematical foundation: input data are independent and are part of identical distributions. However, the health condition of a complex system is a continuous and gradually changing time-series process, which does not meet the characteristics of independence and identical distributions. The distribution changes as time goes on, which also agrees with the reality of ageing equipment in operation [15]. Due to its continuous and gradual characteristics, relevant features contain a large amount of time-series information. Therefore, time-series data mining algorithms such as recurrent
neural networks (RNN) are applicable [16] and can produce more accurate complex system health condition estimation. However, the linear execution characteristics of time-series data mining models typically yield low execution efficiencies. Particularly for a learning model that contains a large number of multiple types of observation features, such as a complex system, the algorithm execution speed will decrease [17]. Therefore, we use the latest research results of RNN called the simple recurrent unit (SRU), which is a time-series machine-learning algorithm with a more efficient execution rate [18]. Combined with the ISS model and RTA model mentioned above, a tree-structure SRU (T-SRU) model is developed to improve the efficiency of algorithm execution based on fully mining the time series characteristics of complex system health conditions.

Few systematic publications exist about the health condition estimation method of complex systems. Previous research primarily focused on simple components such as oil pipelines, lithium batteries, and rotating bearings [19]. Health condition features, such as crack length [20], time interval equal discharge voltage difference (TIEDVD) [21] and gyro deflection angle [22], are considered and have a strong correlation with the health condition of the above-mentioned research objects, directly describing the declining trend of their health condition. Also, a complex system can be considered as a black box and directly predict the remaining useful life without real-time health condition estimation. However, real-time complex system health condition estimation is more important to high-precision equipment such as spacecraft. However, a complex system is more complex than these research objects. It is difficult to abstract a system health index because the health condition of the complex system is just a relative concept, and it is difficult to accurately describe it with a single index. Therefore, we use the transfer learning model [23] and first consider normal-operation data and failure data to train the estimation model to achieve coarse-grained health condition estimation. Then, combined with the characteristics of the gradual decline of health conditions, the coarse-grained model is fine-tuned using sampling data of the entire life cycle to obtain a fine-grained estimation model.

The classification and typical methods of the existing health condition estimation models, as well as the commonalities and differences between these methods and the T-SRU, are shown in Table 1.

This study is motivated by the following: (1) research on the health condition estimation of complex systems such as spacecraft has important practical significance, but there are few related studies currently primarily because complex systems typically do not have a comprehensive health index; and (2) complex systems have complex internal structures and diverse observation features. Feature selection, feature organization, and health condition estimation are thus required concurrently. However, there is no published collaborative model of these three components.

The primary contributions of this paper are as follows:

- We propose a comprehensive complex system health condition estimation method, which allows for the collaborative calculation of feature selection, feature organization and condition estimation.
- We propose a feature selection method based on information stability. Selection results can be combined with expert knowledge to construct a reliability tree, and this tree structure can be used to design a T-SRU estimation model.
- We use transfer learning to solve the lack of a single health index for complex systems and realize the health condition estimation of complex systems.

The remainder of this paper is organized as follows. ISS feature selection and RTA feature processing methods are described in detail in the following section. In the next section, we introduce the implementation process of the T-SRU model. In the next section, turbofan engine data are used to verify the proposed method. Conclusions and future directions are given in the next section.

**Complex system health condition feature processing model**

With increasing system complexity, the types of health condition features also expand. Finding relevant features that can effectively describe the health condition of a complex system from the various types of health condition features and effectively organizing these health conditions’ relevant features has become critical to complex system health condition estimation [24].

**Feature selection model based on information amount and stability**

Because there are many types of health condition features of the complex system, the health condition estimation is performed directly without feature selection. This process may cause the dimensionality problems with machine learning algorithms, which could make the algorithm inefficient and difficult to converge. Conversely, this process may be affected by irrelevant features, which may cause overfitting and cannot accurately describe the degradation process of the complex system health condition [7]. Therefore, the number of input features of the health estimation model and the estimation accuracy generally satisfy the relationship shown in the curve below (Fig. 2): the appropriate number of features can achieve the best estimation accuracy, and too few or too many features will reduce estimation accuracy [9].
Table 1 Literature review

| Estimation model | Typical method | Commonalities | Differences |
|------------------|----------------|---------------|-------------|
| Knowledge model  | Analytic hierarchy process, belief rule base, etc | The reliability tree construction in T-SRU requires the participation of expert knowledge. In addition, knowledge has an important impact on the final estimation results | The knowledge model directly uses expert knowledge to estimate the system health condition, while T-SRU is primarily using the input data |
| Physical model   | Physical equivalent circuits, electrochemical analysis, etc | The reliability tree in T-SRU is a simplification and simulation of a complex system internal structure. The coefficients of each node in T-SRU can be considered to describe a simple physical equivalent model of system reliability | The physical model obtains an approximate result of the reliability by analysing and simulating the internal operating state of the system, but this type of method is only suitable for relatively simple devices and is difficult to apply to complex systems |
| Data-driven model| Recurrent neural networks, convolutional neural network, etc | T-SRU is essentially a data-driven method with a combination of various statistical methods and neural networks. By setting parameters and combining with existing data, the black-box model is trained to obtain health condition estimation results | T-SRU analyses a complex system’s internal structure using expert knowledge and simple physical equivalent models such as reliability trees, whose output results are not completely determined by the input data |

Fig. 2 Schematic diagram of the relationship between the number of features and the accuracy of estimation

Therefore, it is necessary to select a specific number of features. The commonly used feature selection models primarily include filter selection, wrapper selection, and embedding selection. The classification is based on whether the selection model refers to the health condition estimation results.

The filter model does not depend on the health condition estimation results, directly selects the features by establishing a certain feature measurement model, and then uses these selected relevant features for estimation model training. Therefore, the filter model typically achieves good computational efficiencies, but feature selection is highly dependent on the feature measurement model, which is typically based on the prerequisite of a better understanding of the distribution of all the health condition features. A typical feature selection model such as Relief considers the importance of the features by designing a related statistic [25].

The wrapper model completely relies on the health condition estimation results, which means that relevant features are selected using the estimation results. Therefore, the wrapper model can achieve high-precision feature selection, but its execution efficiency is low. A typical wrapper feature selection model is the Las Vegas wrapper (LVW), which uses a random strategy to search for relevant features by combining the estimation results [26]. The embedding model partially uses the health condition estimation results and continuously optimizes the feature set during model training. Therefore, from a theoretical analysis, the embedding model partly improves the execution efficiency while ensuring accuracy. However, this ideal balance state is difficult to achieve in the real model training process, and multiple comparison experiments are required to determine the final set of relevant features. Typical embedding feature selection models such as L1 regularization can make the solution sparse during model training, effectively reduce the dimension of the problem, and improve the efficiency of the machine learning algorithm [27].

Due to the difficulty in accurately describing the complex system health condition, it is difficult for the wrapper model or the embedding model to obtain accurate estimation results as a selection reference. In addition, due to the gradual characteristics of the complex system health condition, its relevant features should show a certain degree of stability and relative monotonicity. It would be convenient to use the filter feature selection model based on statistics, which can
For the relevant features of the complex system health condition, two basic requirements must be met. The first is to include sufficient information. Because the health condition is a gradual decline process, the changes in some features may be weak, and it is difficult to identify after noise is superimposed. Therefore, only those features that change markedly during the entire life cycle are more practical, which is easier to apply to health condition estimation. The second is stability. For a complex system of the same type and different entities, the features of their health condition should show similar distribution characteristics throughout their entire life cycle.

Based on the above analysis, we propose an information-stability selection (ISS) model that is used to select relevant features of the complex system health condition, which primarily includes three steps. The first step is to select the information amount. Starting from the distribution of the features themselves because the health condition is a gradually declining process, there are many features that exhibit small changes, which make it difficult to provide sufficient information for the health condition estimation. A cumulative relative information amount is designed in this study to calculate the information amount coefficient matrix of the features and to select those health condition features with sufficient information. The second step is stability selection. Starting from the distribution of the same feature under different conditions, a stable feature distribution is required for health condition estimation. This study uses the Pearson coefficient to select the stable distribution in the time sequence (i.e., the feature that changes more regularly). The third step is to synthesize the correlation coefficient matrix of the relevant feature according to the coefficients generated by the information and stability selection. The implementation process of the ISS model is shown in Fig. 3.

The cumulative relative information amount calculation method is as follows.
The stability calculation uses the Pearson coefficient, and the computation equation is as follows:

\[
R(x_i, u) = \frac{\sum_{T=1}^T x_i^j u_k - \sum_{T=1}^T x_i^j \sum_{T=1}^T u_k}{\sqrt{\sum_{k=1}^p (x_i^k)^2} \sqrt{\sum_{k=1}^p (u_k)^2}} \tag{1}
\]

For the health condition features, the amount of information evaluates the time series changes, and the stability evaluates the distribution differences of the feature under different work conditions. Therefore, the features with sufficiently large variations and stable distributions under different conditions can describe the health degradation of the complex system more accurately. The calculation of the feature correlation is as follows:

\[
H(x^j) = \alpha F(x^j) + \beta \sum_{i=1}^p R(x_i^j, u) \quad \alpha + \beta = 1 \tag{2}
\]

Based on this analysis, the ISS model can select features for the complex system, but there is still a strong hierarchical coupling relationship between the complex system health condition and the relevant features. Therefore, it is necessary to construct an effective tree analysis model based on the internal structure of the complex system to use these relevant features more effectively.

### Feature organization model based on reliability tree analysis

For a complex system, because there is no objective health condition index, such as the remaining capacity of lithium batteries, the health condition of complex systems is a relative concept and has a certain degree of subjectivity. In the study of complex system health condition estimation, the key is how to effectively associate the relevant features with the system health condition. Currently, there are two common ideas used in the complex system health condition estimation. First, the health condition of the components with the most degradation is considered to be the overall health condition of the complex system; thus, the goal of the algorithm is to find the current shortest board of the complex system. Second, we perform a weighted summation of the different components’ health conditions to obtain the system health condition. These two solutions have their own scopes of application, and how to accurately describe the complex system health condition is a challenging research topic.

Because a complex system often contains multiple key components, the health condition of a single component is associated with multiple relevant features. This many-to-many hierarchical relationship brings certain difficulties to the complex system health condition estimation. Therefore, establishing an effective hierarchical analysis model for relevant features, component health conditions, and system health conditions becomes the key to realizing the health condition estimation of a complex system.
We use fault tree analysis (FTA) and the fuzzy reliability analyser (FRA) method used by Abdelgawad et al. [28] to develop a more general complex system health condition feature organization model, which we call reliability tree analysis (RTA). To standardize the RTA model, we first define several key concepts and define relevant features as the information node (IN), the health condition of a single component as the component node (CN), and the health condition of the system as the system health index (SHI). The organizational relationship of these parts is shown in Fig. 4.

By analysing the feature values of the IN, the current condition of the CN can be obtained. The fusion processing of the CNs can obtain the SHI. Although this model cannot completely express the internal coupling relationship of the complex system, it can be intuitive to understand the logical relationship between the various parts.

The reliability tree $RT = \{ X, Y, r, E, f, g \}$ is a six-tuple, where $X$ is the IN, which contains three parts: feature name, node weight, and feature value. $Y$ is the CN, which contains the component node, name, node weight, and health condition value. $r$ is the root node, which is SHI. $E$ is the set of edges between nodes, primarily including the names and directions of connecting nodes. $f$ is the corresponding relationship between the input data and the node. $g$ is the numerical mapping relationship between nodes, which must be used in conjunction with the $f$ function.

The above definition shows that RT is an ordered attribute tree, which requires certain expert knowledge to construct. The complex system has various types of components and features, and the hierarchical structure is more marked; thus, the RTA model is suitable.

The weight of each node in RT can be calculated from the correlation produced by the ISS model. The calculation method of information node weight and component node weight is as follows:

\[
I(x_u, v) = \frac{H(x_u, v)}{\sum_{v=1}^{V_u} H(x_u, v)},
\]

\[
C(x_u) = \frac{\sum_{v=1}^{V_u} H(x_u, v)}{\sum_{u=1}^{U} \sum_{v=1}^{V_u} H(x_u, v)}
\]

where $U$ is the number of component nodes and $V_u$ is the number of information nodes under component node $u$.

**T-SRU model for complex health condition estimation**

**Introduction to RNN and GRU**

Traditional neural networks have good data fitting capabilities and can better solve various classification and regression problems. However, they are not suitable for processing time series data because traditional neural networks assume that the input data meet independent and identical distributions, while time series data do not meet such assumptions. To effectively process time series data, Pollack et al. proposed a recurrent neural network (RNN) in 1990 [29] and achieved remarkable results in natural language processing. By establishing a time step-based cyclic iterative process in the RNN, the state information of the previous time node is transferred to the current time node, thereby remembering relevant information and being able to process time series data more effectively. The RNN contains two inputs and two outputs. The inputs include the time series data input by the current time node and the hidden state data of the previous time node obtained via iteration. The outputs include the output result of the current time node, and the hidden state must be transferred to the next time node. The architecture of the RNN is shown in Fig. 5.

In Fig. 5, $x$ is the input, $h$ is the hidden variable, $y$ is the output, $U$ is the conversion matrix from the input layer to the hidden layer, $V$ is the conversion matrix from the hidden layer to the output layer, the time series state transfer between the hidden layers is achieved by the weight matrix $W$, and $t$ is the discrete time series state. The transformation relationship between various variables in RNN can be described as:

\[
h_t = \sigma(Ux_t + Wh_{t-1}),
\]

\[
y_t = \sigma(Vh_t),
\]

where $\sigma(x)$ is a nonlinear activation function such as tanh and sigmoid, and the network parameters are updated through the gradient descent algorithm.

In an RNN, because the state at any time needs the hidden state information of the previous time node, the algorithm can only be executed sequentially. As the dimensionality and length of time series data increase, the number of parameters in RNN increase at an exponential rate, resulting in a marked drop in network computing efficiency, and problems of gradient vanishing and gradient exploding are prone to occur. Therefore, traditional RNNs cannot effectively process long-term series of data.

To improve the execution efficiency of RNNs, networks such as long short-term memory (LSTM) networks and gated recurrent units (GRUs) have been proposed. LSTM achieved selective memory by designing gate units that mitigate gradient vanishing and gradient explosion in RNNs, as well as excessive dependence on recent data; thus, LSTM is more suitable for processing time series data with longer sequences [30]. However, due to the introduction of more parameters, the training of LSTM is more difficult. To solve the LSTM training problem, Cho Kyunghyun et al. proposed the GRU algorithm, which achieved a simplified gate unit and algorithm performance similar to LSTM [31]. The computational
efficiency of GRU has been markedly improved compared to earlier RNNs. The internal structure of the GRU is shown in Fig. 6.

The conversion relationship of each variable in GRU can be described by the following equations:

\[ r_t = \sigma (R \times [x_t, h_{t-1}]) \]

\[ z_t = \sigma (U \times [h_{t-1}, x_t]) \]

\[ h'_t = \tanh (W \times [r_t \odot h_{t-1}, x_t]) \]

\[ h_t = (1 - z_t) \odot h_{t-1} + z_t \odot h'_t \]

\[ y_t = \sigma (V \odot h_t) \]

where \( r \) is the reset gate, \( z \) is the update gate, \( \sigma (x) \) is the sigmoid function, \([a, b]\) represents the splicing operation of the vector, \( R \) is the weight matrix of the reset gate, \( h'_t \) is the candidate hidden state, and \( \odot \) is the Hadamard product. The
Tree-structured simple recurrent unit

RNN and its variants exhibit good performance for processing time series data, but even for LSTM or GRU, its calculation still has a serial design and thus cannot efficiently use the parallel computing ability of the computer. When faced with a large amount of data, such as the health condition of the complex system, the execution rate of the algorithm will decrease. To effectively improve the computational efficiency of RNNs, Tao Lei et al. proposed a simple recurrent unit (SRU) in 2017 [18]. Using a light recurrent unit, SRU can effectively extract the input data and hidden states in the time series data and decouples the front and back time series relationships so that the SRU can perform parallel calculations. In addition, SRU also uses high network technology, adding additional connections to the network, making the training process jumpy, and markedly improving the training rate of the network [32]. The structure of the SRU is shown in Fig. 7.

The relationship of various variables in the SRU can be described by the following equations:

\[
\begin{align*}
f_t &= \sigma(Fx_t + b_f), \\
h_t &= f_t \odot h_{t-1} + (1 - f_t) \odot Ux_t, \\
r_t &= \sigma(Rx_t + b_r), \\
y_t &= r_t \odot \tanh(h_t) + (1 - r_t) \odot x_t,
\end{align*}
\]

where \( F \) and \( b_f \) are the parameter matrix and offset of the forget gate, respectively, and the remaining parameters are consistent with the previous section. Based on the structure and variable relationship of the SRU, the forget gate control does not depend on the data at \( t - 1 \) time; thus, it has the basis for parallel computing. Concurrently, by introducing \( (1 - r_t) \odot x_t \) such a jump parameter to the output \( h_t \), the back propagation of the gradient is realized. Therefore, the SRU can achieve better training results when the number of network layers is large and can effectively avoid the occurrence of gradient vanishing.

In the health condition estimation of the complex system, each relevant feature presents a certain hierarchical relationship with the health condition of the system. With the help of the reliability tree constructed in the previous section, this paper proposes a tree-structure SRU (T-SRU) model for the health condition estimation of a complex system.

A memory module is added to the SRU, which is used to save the tree structure health condition relationship of the complex system. Combined with previous research on tree-structure RNNs, the commonly used tree structures can be divided into two types. The first is the child-sum tree, which directly uses the output value of each child node as the input of the parent node. In this structure, the parent node can selectively forget the input value of a child node through a forget gate but does not set dynamic weights for each child node. This tree structure is suitable for datasets with a large number of child nodes and no explicit hierarchical structure [33]. The second is the N-ary tree that considers the different effects of the output value of the child node on each gate and is implemented by setting the corresponding weight matrix. This type of tree structure is suitable for a dataset with a relatively small number of child nodes, and a dominant tree structure can be constructed [34].

Combining the ISS model and the RTA model proposed above, the N-ary tree structure should be used for the health condition estimation of the complex system; thus, we propose a new type of SRU algorithm with an N-ary tree structure, and its implementation structure is shown in Fig. 8.

The input of the T-SRU is no longer limited to input data and hidden states and can also be the output of other SRU units. Considering node SRU-2 in Fig. 8 as an example, its gate structure is shown in Fig. 9. The forget gate of node SRU-2 contains four partial inputs, which are its own input data, the data of information node \( x_4 \), and the output result of SRU-4 & SRU-5. The hidden state of node SRU-2 includes three partial inputs, which are the hidden state of the node at the previous time and the output hidden states of SRU-4 & SRU-5.

For the \( n \)th node, its internal parameter update method can be expressed as:

\[
\begin{align*}
f_t &= \sigma(Fx_t + \sum_{l=1}^{N} F_l^{(a)} h_{l,t} + b_f), \\
h_t &= f_t \odot \left( h_{t-1} + \sum_{l=1}^{N} W_l^{(a)} h_{l,t} \right) + (1 - f_t) \odot Wx_t, \\
r_t &= \sigma(Rx_t + \sum_{l=1}^{N} R_l^{(a)} h_{l,t} + b_r), \\
y_t &= r_t \odot \tanh(h_t) + (1 - r_t) \odot x_t,
\end{align*}
\]

where \( j \) is a certain child node and \( N \) is the number of child nodes of node \( n \). Thus far, this section has designed the internal parameter update method of the T-SRU model.
T-SRU model for complex system health condition estimation

Based on the above discussion, we propose a T-SRU-based complex system health condition estimation model, and its implementation process is shown in Fig. 10.

As shown in Fig. 10, the implementation process of the T-SRU model proposed in this paper is:

Step 1: The complex system health condition feature set is input, and the ISS model is used to select the relevant features and obtain the correlation coefficient.

Step 2: Expert knowledge is used to build a reliability tree of the complex system and calculate the weight of each node.

Step 3: We establish a T-SRU architecture based on the reliability tree and the weight of each node.

Step 4: The normal operation data and failure data of the time series are used in the training set to pretrain the T-SRU to obtain a coarse-grained estimation model.

Step 5: To mitigate the effects of noise in the data, we use the entire life cycle health condition sampling data to fine-tune the coarse-grained estimation model to obtain a fine-grained model.

Step 6: We input the time series test data into the fine-grained health condition estimation model to obtain the test estimation results.

Case study

Considering the characteristics of a complex system with multiple features and components, we use turbofan engine data from NASA [35] to perform complex system health condition estimation experiments. The dataset includes four types of engines, and each type of engine contains a training set and a test set. The training data record 24 monitoring data in each flight cycle before failure, and the test data only contain incomplete life cycle monitoring data. The first type of engine (FD001) contains only one working condition and one failure mode. This section primarily uses this type of engine to verify the performance of the complex system health condition estimation models.

The feature selection, feature organization, and condition estimation methods proposed in this paper are both interrelated and independent of each other. Different programming
languages and platforms are used in the real coding process, among which feature selection and feature organization are coded in M language and run on the MATLAB R2020a platform. Condition estimation is performed using Python 3.8 and TensorFlow 2.4. The CPU of the experimental platform is Intel i7 1165G7, its primary frequency is 2.80 GHz with a maximum turbo frequency of 4.7 GHz, and the L3 cache is 12 MB. The system memory is 16 GB, the GPU is NVIDIA GeForce MX450, and the software environment is Windows 10.

**Feature processing**

The FD001 dataset contains 24 types of monitoring data. The first three data types are used to describe operating conditions, and the other 21 are performance features of related components. The internal structure of this engine type can be expressed as the Fig. 11.

First, we perform feature selection and select the first 20 engines of FD001 data. To mitigate the influence of different data lengths on the experimental results, 30 time series data are extracted at equal intervals in each engine. The average information amount and stability (Pearson coefficient) are calculated as shown below.

Because the amount of information changes, it is difficult to quantitatively evaluate it; thus, it can only be used as one of the estimation criteria to remove those features that hardly change, such as features 6, 10, 11, and 15 in Fig. 12. Therefore, $\alpha$ should be small and is set equal to 0.3 in this section. The Pearson coefficient has a more intuitive response to the stability of the feature, and the corresponding relationship is as follows:

$$R(\mathbf{x}, \mathbf{u}) = \begin{cases} 
0.8 - 1, & \text{Highly correlation} \\
0.6 - 0.8, & \text{Strong correlation} \\
0.4 - 0.6, & \text{Moderate correlation} \\
0.2 - 0.4, & \text{Weak correlation} \\
0 - 0.2, & \text{Uncorrelated} 
\end{cases}$$  \tag{19}

To test the effectiveness of the ISS model, different numbers of engines ($P$) and sampling times ($T$) are selected, and the correlation coefficient is as follows.

Figure 13 shows that when $P$ (the number of selected engines) increases, $H$ (feature correlation) decreases because as the number of selected engines increases, more environmental impacts are included; thus, the relevant features are representative declines. When $T$ (the number of samples per group of engines) increases, $H$ decreases due to the presence
of noise. As the number of samples increases, the degradation curve fluctuates more widely, resulting in a decrease in stability and thus a decrease in $H$. However, under different experimental conditions, the results obtained by the ISS model are similar, which verifies the effectiveness of the ISS model.

The final selected features are the features ‘T50’, ‘P30’, ‘Ps30’, ‘phi’, ‘BPR’, ‘W31’ and ‘W32’, which are in the green box.

Combining the corresponding expert knowledge, the following R-tree is obtained. When $P = 60$ and $T = 60$, we calculate the weight of each node (Fig. 14).

After computation, seven INs and four associated CNs are selected. The INs T50 and W32 are the outlet temperature and cooler bleed of the CN LPT, and the INs P30, Ps30 and BPR are the outlet total pressure, outlet static pressure and ratio of fuel flow to Ps30 of the CN HPC. The IN Phi is the bypass ratio of the CN Fan, and the IN W31 is the cooler bleed of the CN HPT.

The failure mode of FD001 data is HPC degradation, the 3/7 features (P30, Ps30 and BPR) selected by the proposed ISS model are directly related to HPC, and the weight of CN HPC is 0.433, which is much higher than other CN weights. This weight indicates the effectiveness of the ISS model and RTA model proposed in this paper.

Comparison of the different estimation models

We set the SRU parameters according to the R-tree structure in the previous section and set a three-layer network structure for each of the four CNs. The input unit number of each CN is the number of INs corresponding to this CN. The size of the hidden layer is 20, and the node number of the output layer is 1. The SHI node also has a 3-layer network structure, the input units are those 4 CNs, the size of the hidden layer is 20, and the node number of the output layer is 1. The initial weight of each input layer uses the corresponding weight in the R-tree. The loss functions of the network pretraining and
In the literature about device health management, health condition estimation is often used as part of remaining useful life (RUL) prediction. For example, Liu et al. used the fuzzy clustering method for health condition estimation and then used LSTM for prediction [36]. Kim et al. used a deep CNN combined with a multitask learning (MT-CNN) framework to perform health condition estimation and RUL prediction [37]. The application of these methods is often complex, involving data standardization, smoothing, feature clustering, life stage division and other preparation stages. These preparation stages require considerable expert knowledge and control parameters, which make these methods difficult to generalize. However, the proposed ISS selection model can automatically build a reliability tree and assign the initial weights to corresponding network nodes, which produces good generalizability. To perform a comparative experiment, we use logistic regression [38], fuzzy clustering [36] and MT-CNN [37], and refer to the corresponding literature for the appropriate parameter settings. The initial learning rate of each SRU is 0.01, the learning rate is attenuated by 90% per 1500 iterations, the maximum number of iterations is 6000, the number of engines during fine-tuned training is 60, and the number of samples is 60.

To effectively compare the estimation accuracy of different types of methods, the original data are not smoothed and staged to avoid the influence of subjective parameters on the experimental results. This experiment uses the last 99 engines in train_FD001 as the training data and uses the first engine as the test data. The first five flight cycles of each engine are selected as the normal operating state, and their health condition value is 1. The last two flight cycles are the failure state, and their health condition values are 0, which are used to train each estimation model. All initial data are normalized so that their values fall within [0,1]. When not performing feature selection, the SRU network based on the transfer learning idea is used to compare experiments with other models. The results are shown in Fig. 15.

These results show that the fuzzy clustering model cannot achieve complex system health condition estimation without feature selection. This result likely occurs because this model uses the distance between the feature set and the initial state as a parameter. However, the feature set contains many irrelevant features, which exhibit a strong change (e.g., features 7, 8 and 22 from the experiment in the previous section). Therefore, the distance-based model cannot achieve accurate estimation. The logistic regression model is based on an exponential function, which can describe the degradation process of the complex system, but there is an important phenomenon of premature ageing. From the 90th set of data, there is a significant performance degradation, which is inconsistent with the conclusion in the mainstream literature that this engine began to decline after approximately 130 groups [39]. Therefore, the logistic regression model cannot describe the engine’s performance degradation accurately. However, the MT-CNN model can roughly describe the degradation of the engine, but its estimation results exhibit marked fluctuations during degradation, which is consistent with the experimental results of [37] under multiple operating conditions. This result likely occurs because the data in the degradation period account for a low proportion of the entire life cycle, and the deep network model is prone to overfitting. Although this fluctuation persists in the SRU model, due to the characteristics of time series learning, the SRU model achieves the health condition estimation of the complex system in a relatively stable manner. Therefore, time series learning models such as SRU are more suitable for complex system health condition estimation research.

Next, based on feature selection, we use the proposed T-SRU model, where the number of engines and the number
of samples during fine-tuning training are still equal to 60, and those other models are used for comparison experiments. The results are shown in Fig. 16.

As shown in Fig. 16, based on feature selection, each model can roughly fit the engine’s health degradation process. The premature ageing phenomenon of the logistic regression model and the mid-stage fluctuation phenomenon of the MT-CNN model still exist. Although the estimation results have improved, the limitations of the above model are still evident when estimating the health condition of a complex system. Because the T-SRU model uses the full life cycle time series sampling data to fine-tune the coarse-grained model, it can more stably and accurately describe the degradation of the health condition of the complex system.

Based on this analysis, the selected SRU model is suitable for complex system health condition estimation, and the T-SRU model constructed with the ISS model and RTA model can achieve more stable and accurate complex system health condition estimation.

**Health condition estimation effectiveness of the T-SRU model**

In this section, the influence of the ISS model, RTA model and sampling numbers on the convergence speed of the T-SRU model is discussed. We consider no feature selection, feature selection without tree organization, and feature selection with tree organization without calculating node weights as the comparison group with T-SRU. Under the conditions of different engine numbers and sample numbers, the algorithm running time of the above four cases is computed, and the results are shown in Fig. 17.

The parts marked by the red circles in Fig. 17 show that without feature selection, the SRU algorithm cannot converge when a lot of data is present primarily due to the increase in irrelevant features and the amount of data, which greatly increases the learning cost of the algorithm. Considering feature selection without tree organization, when the number of engines is 90, and the number of samples of each engine is 150 (when less than 150, all samples are selected), convergence is not achieved within the limited number of
iterations. Considering feature selection and tree organization without using the RTA model to calculate the node weight, the run time is typically longer than that of the T-SRU model. Therefore, the T-SRU model markedly improves the operating speed of the SRU algorithm. Considering the no-tree-structure SRU as a baseline, the computational efficiency of the T-SRU method improves by approximately 46%. Therefore, the T-SRU model can more efficiently estimate the health condition of a complex system. However, the high execution speed of the model is important in real production.

All previous experiments are only for a certain engine to verify the general validity of the proposed T-SRU model. This section uses 80% of the data in train_FD001 as the training set and the remaining 20% as the test set. After feature selection by the ISS model, logistic regression, fuzzy clustering, MT-CNN, and T-SRU were used for health condition estimation. Experimental results show that the premature ageing phenomenon of the logistic regression model and the mid-stage fluctuation phenomenon of the MT-CNN model are still relatively important, which is consistent with the single-engine experimental results in the previous subsection. However, the...
estimation results of the fuzzy clustering and T-SRU methods on the test set are shown in Fig. 18.

As shown in Fig. 18, the engines in the FD001 data show a marked degradation trend and eventually fail, which conforms to the distribution characteristics of the FD001 training dataset and verifies the general validity of the proposed T-SRU model. Although the fuzzy clustering model can also describe the degradation trend of FD001 data, it has more significant fluctuations in the early stage, which is inconsistent with the reality that the early performance of the engine is maintained well. For the estimation results of the entire life cycle, the stability of the T-SRU model is also better than that of the fuzzy clustering model. From a theoretical analysis, this result occurs because fuzzy clustering is based on distance and cannot assign effective weights to data in each dimension; thus, the model is more sensitive to data fluctuations. The T-SRU method calculates the weight of each node separately through the RTA model, which effectively improves the stability of the estimation model.

This section shows that the T-SRU model proposed in this paper could improve the convergence speed of the SRU algorithm, avoid dimensionality problems, and effectively estimate the health condition of all the FD001 training engines, indicating a general validity of the complex health condition estimation. Therefore, the proposed T-SRU model can achieve relatively efficient and accurate health condition estimation of a complex system.

Control parameters of the T-SRU model

The T-SRU model includes basic parameters such as $\alpha$, $\beta$, $P$, and $T$, as well as network parameters such as the number of network layers, the number of hidden neurons, and the type of activation function that controls the SRU estimation model. $\alpha$ and $\beta$ are primarily used to balance the respective proportions of information amount and stability in feature selection. Because the amount of information is relative, stability is often more important in practical processing applications. In the previous subsection, we set $\alpha = 0.3$ and $\beta = 0.7$. However, under the conditions of $P = 60$ and $T = 60$, $\alpha = 0.7$, $\beta = 0.3$; $\alpha = 0.5$, $\beta = 0.5$; $\alpha = 0.3$, $\beta = 0.7$; $\alpha = 0.1$, $\beta = 0.9$, and the correlation coefficients are described by shown in Fig. 19.

When $\alpha$ is large, the correlation coefficient of each feature is typically large, resulting in small differences. Therefore, making $\alpha$ small maximizes the difference of each feature. $\alpha$ has little effect on the feature selection result but does affect the calculation of the weight of each node in the subsequent RTA model. To achieve a differentiated distribution of the weights of each node, the effect is better when $\alpha$ is between $[0.2, 0.4]$; this relationship is found experimentally. However, $\alpha$ and $\beta$ are determined by the distribution characteristics of the data and must be adjusted for different datasets. The principle of adjustment is to make the correlation coefficient and node weight have a sufficiently differentiated distribution.

Because SRU networks have many control parameters, and there are many more targeted studies, we only discuss the influence of the number of network layers ($L_n$) and the number of hidden neurons ($H_n$) on the health condition estimation results. When $P = T = 60$, $\alpha = 0.3$, and $\beta = 0.7$, $L_n$ is set to 4 and 5, and $H_n$ is set to 30 and 50. Neurons between different layers are fully connected; the other parameter settings and training methods are the same as in the previous section; and the estimation results are obtained in Fig. 20.

These results show that the estimation results are the most stable when $L_n$ is 4 and $H_n$ is 30. However, when $L_n$ is 5...
and Hn is 50, the stability of the estimation results is lowest, and there is a strong fluctuation in the middle period. This result indicates that with increasing Ln and Hn, the estimation model appears to overfit. When Ln and Hn exceed a certain number, the stability of the estimation model will thus decrease. In this case, the change in Hn is larger. Through experiments, we find that when Ln is 4, and the Hn layer is 30, the estimation stability of the model is higher.

This section validates the complex system health condition estimation ability of the T-SRU model with NASA turbofan engine degradation data. Results show that the T-SRU can effectively integrate feature selection, feature organization and condition estimation to achieve a more accurate health condition estimation of complex systems and improve the execution efficiency of the SRU networks by approximately 46%.

**Conclusion**

A complex system is characterized by its diverse feature types, complex internal structure, and lack of an explicit health index; thus, its health condition estimation models have always been a challenging research topic. In response to the abovementioned difficulties, we propose a feature selection model based on the amount of information and stability. This feature selection model is verified using NASA turbofan engine data, and results show that this model can stably and effectively select relevant features with a larger amount of information and a more stable distribution, which are suitable for complex system health condition estimation. In combination with the internal tree-structure distribution characteristics of the complex system, a reliability tree analysis model is designed, and the node weight is calculated based on the correlation coefficient generated during the feature selection process. The effectiveness of reliability tree analysis is verified by analysing the turbofan engine data. Finally,
because health condition features are rich in time-series information, the SRU algorithm with the characteristics of fast time series learning combined with the reliability tree was used to construct the T-SRU model for the health condition estimation of the complex system. Turbofan engine data are used to verify the estimation accuracy of different estimation models and the estimation effectiveness of the T-SRU model separately to verify the complex system health condition estimation ability of the T-SRU model. Experimental results show that the T-SRU model fully considers the characteristics of the complex system and can achieve a more efficient and accurate health condition estimation of the complex system.

This study provides a preliminary exploration of the health condition estimation of complex systems, but there are still many difficulties in performing related research. In the proposed T-SRU method, feature selection, feature organization, and condition estimation are executed sequentially, and there is one-way information transfer between each component. However, this one-way information transmission may accumulate errors; for example, if feature selection is inaccurate, feature organization and condition estimation accuracy will also be inaccurate. Further research should consider dynamic information transmission between feature selection, feature organization and condition estimation based on the embedded feature selection method, and use condition estimation accuracy as the final index for global optimization. Conversely, SRU networks are limited by their timing execution characteristics. Although the tree structure is designed in this study, it still cannot fully and accurately describe the internal structure of the complex system. Recently, health management technology based on graph neural networks has been developed to a certain extent, with the characteristic of expressing graphical structure data. Further research should pay consider the application and development of data-driven methods, which could accurately represent the complex structures of the system.

Acknowledgements This research was supported in part by the National Social Science Foundation of China (No. 2019-SKJJ-C-025 and 2020-SKJJ-C-033), the Space Science and Technology Innovation Foundation (No. SAST2020-009), and the Natural Science Foundation of Shaanxi Province (No. 2021JQ-368).

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

Appendix A: Variables

Variables in this paper

| Number | Variables | Meaning |
|--------|-----------|---------|
| 1      | \(x_{i,j}^t\) | The \(j\)th feature sequence of device \(i\) |
| 2      | \(x_{i,k}^t\) | The \(k\)th value of the \(j\)th feature sequence of device \(i\) |
| 3      | \(\bar{x}_{i,j}^t\) | The sampling sequence of the \(j\)th feature sequence of device \(i\) |
| 4      | \(T\) | The number of sampling times |
| 5      | \(C\) | The number of feature types |
| 6      | \(P\) | The number of selected engines (devices) |
| 7      | \(\bar{x}_i\) | The relative information amount of all sample features |
| 8      | \(\alpha\) | The control coefficient of information amount part |
| 9      | \(\beta\) | The control coefficient of stability part |
| 10     | \(x_{u,v}\) | The value of the information nodes \(v\) under component node \(u\) |
| 11     | \(V_u\) | The number of information nodes under component node \(u\) |
| 12     | \(U\) | The number of component nodes |
| 13     | \(x_i\) | The component nodes \(u\) |
| 14     | \(x_t\) | The input variables of time \(t\) |
| 15     | \(h_{t-1}\) | The hidden variable of time \(t\) |
| 16     | \(U\) | The conversion matrix from the input layer to the hidden layer |
| 17     | \(W\) | The conversion matrix from the hidden layer to the output layer |
| 18     | \(V\) | The conversion matrix between the hidden layers |
| 19     | \(y_t\) | The output variables of time \(t\) |
| 20     | \(R\) | The weight matrix of the update gate |
| 21     | \(r_t\) | The reset gate value of time \(t\) |
| 22     | \(h'_t\) | The candidate hidden state of time \(t\) |
| 23     | \(z_t\) | The update gate value of time \(t\) |
| 24     | \(F\) | The parameter matrix of the forget gate |
| 25     | \(b_f\) | The offset of the forget gate |
| 26     | \(f_t\) | The forget gate value of time \(t\) |
| 27     | \(b_r\) | The offset of the reset gate |
| 28     | \(L_1\) | The loss functions of the network pretrain stages |
| 29     | \(L_2\) | The loss functions of the network fine-tune stages |
References

1. Yu J, Song Y, Tang D, Dai J (2021) A digital twin approach based on non-parametric Bayesian network for complex system health monitoring. J Manuf Syst 58:293–304
2. Yang D, Zhang X, Pan R, Wang Y, Chen Z (2018) A novel Gaussian process regression model for state-of-health estimation of lithium-ion battery using charging curve. J Power Sources 384:387–395
3. Elattar HM, Elminir HK, Riad AM (2018) Towards online data-driven prognostics system. Complex Intell Syst 4(4):271–282
4. Qiao Z, Elhattach A, Shu X, He C (2021) A second-order stochastic resonance method enhanced by fractional-order derivative for mechanical fault detection. Nonlinear Dyn 106(1):707–723
5. Li C, Li S, Zhang A, He Q, Liao Z, Hu J (2021) Meta-learning for few-shot bearing fault diagnosis under complex working conditions. Neurocomputing 439:197–211
6. Lin B, Song D, He L (2016) Complex system health assessment based on Mahalanobis distance and bin-width estimation technique. Chin J Sci Instrum 37(9):2022–2028
7. Bolón-Canedo V, Alonso-Betanzos A (2019) Ensembles for feature selection: a review and future trends. Inf Fusion 52:1–12
8. Yao X, Wang XD, Zhang YX, Qun W (2012) Summary of feature selection algorithms. Control Decis 27(2):161–166
9. Mendez JR, Cotos-Yanez TR, Ruano-Ordas D (2019) A new semantic-based feature selection model for spam filtering. Appl Soft Comput 76:89–104
10. Saaty TL (2008) Decision making with the analytic hierarchy process. Int J Serv Sci 1(1):83–98
11. Hughes AJ, Barthorpe RJ, Dervilis N, Farrar CR, Worden K (2021) A probabilistic risk-based decision framework for structural health monitoring. Mech Syst Signal Process 150:107339
12. Sun Q, Yu X, Li H, Fan J (2021) Adaptive feature extraction and fault diagnosis for three-phase inverter based on hybrid-CNN models under variable operating conditions. Complex Intell Syst 8(1):29–42
13. Qiao Z, Shu X (2021) Coupled neurons with multi-objective optimization benefit incipient fault identification of machinery. Chaos Solitons Fractals 145:110813
14. Ahwia M, Wang W (2020) An adaptive particle filter technique for system state estimation and prognosis. IEEE Trans Instrum Meas 69(9):6756–6765
15. Zhang YM, Wang H, Bai Y, Mao JX, Chang XY, Wang LB (2021) Switching Bayesian dynamic linear model for condition assessment of bridge expansion joints using structural health monitoring data. Mech Syst Signal Process 160:107879
16. Chen L, Xu G, Zhang S, Yan W, Wu Q (2020) Health indicator construction of machinery based on end-to-end trainable convolution recurrent neural networks. J Manuf Syst 54:1–11
17. Dey R, Sulem FM (2017) Gate-variants of gated recurrent unit (GRU) neural networks. In: 2017 IEEE 60th midwest symposium on circuits and systems (MWSCAS). IEEE, pp 1597–1600
18. Yao D, Li B, Liu H, Yang J, Jia L (2021) Remaining useful life prediction of roller bearings based on improved 1D-CNN and simple recurrent unit. Measurement 175:109166
19. Berecibar M, Gandiaga I, Villarreal I, Omar N, Van Mierlo J, Van den Bossche P (2016) Critical review of state of health estimation models of Li-ion batteries for real applications. Renew Sustain Energy Rev 56:572–587
20. Tian L, Wang Z, Liu W, Cheng Y, Alsaaied FE, Liu X (2021) An improved generative adversarial network with modified loss function for crack detection in electromagnetic nondestructive testing. Complex Intell Syst 8(1):467–476
21. Liu D, Wang H, Peng Y, Xie W, Liao H (2013) Satellite lithium-ion battery remaining cycle life prediction with novel indirect health indicator extraction. Energies 6(8):3654–3668
22. Allseits E, Kim KJ, Bennett C, Gailey R, Gaunaurd I, Agrawal V (2018) A novel model for estimating knee angle using two leg-mounted gyroscopes for continuous monitoring with mobile health devices. Sensors 18(9):2759
23. Li Y, Li K, Liu X, Wang Y, Zhang L (2021) Lithium-ion battery capacity estimation—a pruned convolutional neural network approach assisted with transfer learning. Appl Energy 285:116410
24. Roman D, Saxena S, Robu V, Pecht M, Flynn D (2021) Machine learning pipeline for battery state-of-health estimation. Nat Mach Intell 3(5):447–456
25. Urbanowicz RJ, Meeker M, La Cava W, Olson RS, Moore JH (2018) Relief-based feature selection: introduction and review. J Biomed Inform 85:189–203
26. Tong S, Yang J, Zong H (2021) A prediction model for complex equipment remaining useful life using gated recurrent unit complex networks. Enterp Inf Syst 1–17. https://doi.org/10.1080/17517755.2021.2008515
27. Ng AY (2004) Feature selection, L 1 vs. L 2 regularization, and rotational invariance. In: Proceedings of the twenty-first international conference on Machine learning, p 78
28. Abdelgawad M, Fayek AR (2011) Fuzzy reliability analyzer: quantitative assessment of risk events in the construction industry using fuzzy fault-tree analysis. J Constr Eng Manag 137(4):294–302
29. Pollick JB (1990) Recursive distributed representations. Artif Intell 46(1–2):77–105
30. Gers FA, Schmidhuber J, Cummins F (2000) Learning to forget: continual prediction with LSTM. Neural Comput 12(10):2451–2471
31. Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation. http://arxiv.org/abs/1406.1078
32. Cui X, Chen Z, Yin F (2020) Speech enhancement based on simple recurrent unit network. Appl Acoust 157:107019
33. Ahmed M, Samee MR, Mercer RE (2019) Improving tree-LSTM with tree attention. In: 2019 IEEE 13th international conference on semantic computing (ICSC). IEEE, pp 247–254
34. Peng N, Poon H, Quirk C, Toutanova K, Yih WT (2017) Cross-sentence n-ary relation extraction with graph lstms. Trans Assoc Comput Linguist 5:101–115
35. Saxena A, Goebel K (2008) Turbofan engine degradation simulation data set. NASA Ames Prognostics Data Repository (http://ti.arc.nasa.gov/project/prognostic-data-repository), NASA Ames Research Center, Moffett Field
36. Liu J, Lei F, Pan C, Hu D, Zuo H (2021) Prediction of remaining useful life of multi-stage aero-engine based on clustering and LSTM fusion. Reliab Eng Syst Saf 214:107807
37. Kim TS, Sohn SY (2021) Multitask learning for health condition identification and remaining useful life prediction: deep convolutional neural network approach. J Intell Manuf 32(8):2169–2179
38. Nefeslioglu HA, Gokceoglu C et al (2008) An assessment on the use of logistic regression and artificial neural networks with different sampling strategies for the preparation of landslide susceptibility maps. Eng Geol 97(3–4):171–191
39. Xiang S, Qin Y, Luo J, Pu H, Tang B (2021) Multicellular LSTM-based deep learning model for aero-engine remaining useful life prediction. Reliab Eng Syst Saf 216:107927

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.