Received: May 11, 2022. Revised: June 29, 2022.

INASS

International Journal of Intelligent Engineering & Systems

http://www.inass.org/

Performance and Robustness in Remaining Useful Life Estimation of the Singular Spectrum Analysis and Long Short Term Memory (SSA-LSTM) Algorithms with the Window Length Optimization

Prakit Intachai[1] **Peerapol Yuvapoositanon**[2]*

[1]*Department of Information and Communication Engineering, Faculty of Engineering and Industrial Technology, Phetchaburi Rajabhat University, Thailand*
[2]*Mahanakorn Institute of Innovation, Faculty of Engineering and Technology, Mahanakorn University of Technology, Bangkok, Thailand*
* Corresponding author's Email: peerapol@mut.ac.th

Abstract: In this paper, the performance and robustness in remaining useful life (RUL) estimation of turbofan engines of the singular spectrum analysis and long short term memory (SSA-LSTM) algorithm with the window length optimization are investigated. Specifically, we are interested in how the optimized SSA-LSTM algorithm, i.e., the SSA-LSTM algorithm with its window length being optimized, responds to the variation in the number of cells and in the number of hidden layers of the LSTM structure. The performance and robustness of an SSA-LSTM algorithm are evaluated by two metrics, namely the root mean square (RMSE) ratio and the sensitivity ratio. Four optimized SSA-LSTM algorithms and two non-optimized SSA-LSTM algorithms were tested with the prototype dataset from the turbofan engines datasets. The simulation results showed that all the optimized SSA-LSTM algorithms performed much better in terms of both the RMSE ratios and the sensitivity ratios than the non-optimized SSA-LSTM algorithms. The RMSE ratios of the optimized SSA-LSTM algorithms with the window length of $L = T/2$, $L = \mathrm{T}/4$, $L = \log(T)^{2.3}$ and L =24 were 38.737%, 33.967%, 22.896% and 11.780% respectively. In the same order, the sensitivity ratios of all the optimized algorithms were 2.067%, 1.495%, 1.555% and 1.444% respectively. Among all the algorithms being evaluated, the $L = 24$ optimized SSA-LSTM algorithm provided the best performance in terms of both the RMSE ratio and the sensitivity ratio and hence confirming its best performance and robustness for RUL estimation.

Keywords: Singular spectrum analysis, RUL estimation, Long short-term memory, Performance and robustness, Window length optimization.

1. Introduction

Recently, the process of predictive maintenance (PdM) has gained considerable interest from the industry mainly due to its ability of optimizing the use and management of assets [1]. PdM is a process that turns historical data of the machine of interest into valuable information for action plans in maintenance policies in various sectors, e.g., power plants, utilities, transportations, communications and emergency services [1].

The remaining useful life (RUL) estimation of a machine is an integral part of PdM since it allows early detection of failures before the machine actually reaches its lifetime or its end-of-life [2]. The RUL may be derived from different quantities for different fields of maintenance work. For mechanical engineers, RUL can be derived from fatigue life, crack propagation speed or speed wear rate or a corrosion rate in the field of engineering materials [2]. For wind turbine engineers, RUL can be derived from the gear-mesh frequency [3]. Essentially, RUL is a quantifiable term achieved by using the past as well as the current datasets of a system or a machine to provide early signs of failure

DOI: 10.22266/ijies2022.1031.32

needed for the proactive plan in the form of PdM [4].

The deep learning neural network is one of the most effective data-driven techniques for solving the RUL estimation problem [5]. The Long Short-Term Memory (LSTM) neural network proposed by Hochreiter and Schmidhuber in 1997 [6] is a powerful type of the recurrent neural network (RNN) architecture [7] and is often used as the building blocks for the deep learning structures. LSTM differs from RNN mainly because of the introduction of the forget gate, the input gate and the output gate to the structure of the LSTM cell or unit [8]. In this way, LSTM can store previous data and pass it on to the current step of the process [9]. LSTM was originally designed to solve the vanishing gradient descent problem of the RNN. It is therefore suitable for the RUL estimation problem since it can manage long term dependency of capacity degradation [10]. The LSTM structure has been used in numerous data applications on RUL estimation and prediction Turbofan engines [9, 11-14], Li-Ion Batteries [10, 15], gear [16] and bearing [17]. In comparison, the RUL estimation with LSTM on battery datasets presented in [15] was shown to be more accurate than do the other traditional methods such as RNN or Support Vector Machine (SVM) [18]. The RUL estimation of proton exchange membrane fuel cell (PEMFC) dataset has been presented in [19]. The technique can quickly and accurately forecast the residual service life of the fuel cell compared to the back propagation neural network (BPNN) [20]. In particular, the RUL estimation of turbofan engines by LSTM and the Singular Spectrum Analysis (SSA) algorithm [21-23] in the SSA-LSTM structure has been presented in [13, 14]. It is shown that by first decomposing and reconstructing the datasets with the SSA and then followed by the LSTM regression can gain accuracy in the results of RUL estimation than using the LSTM regression only [13, 14].

The deep learning structure of LSTM is established by using multiple cells and hidden layers and has been deployed for RUL estimation [5, 24]. The RUL estimation presented in [17] is derived from an end-to-end deep learning framework for bearing datasets based on convolutional LSTM recurrent units. In the literature, however, the guidelines for the suitable numbers of cells and the numbers of hidden layers of LSTM for the RUL estimation problem are diversified and inconclusive. For example, the choice of one hidden layer with 20 LSTM cells is used in [9], 50 hidden layers with a fully connected layer is used in [10], 30 cells with two hidden layers in [11], 100 cells with two hidden layers in [15], and varying the number of cells from 32 to 256 for the vanilla LSTM in [12]. Interestingly, while increasing the number of hidden layers of the LSTM structure generally improves the performance of RUL estimation, increasing the number of LSTM cells does not necessarily guarantee a better result. In [16], it is shown that an increase in the number of projection layer cells leads to fluctuating and inconsistent prediction errors.

In this paper, rather than looking for an optimal choice of the numbers of cells and the numbers of hidden layers of the LSTM neural networks, we focus on the performance of the SSA-LSTM algorithm [13] and its robustness to the variations in its LSTM part for the problem of RUL estimation. Specifically, we are interested on how the window length optimization in the SSA part can contribute to the robustness of the SSA-LSTM algorithm to the variations in the LSTM part. In order to answer this question, we introduce the metrics called *the root mean square error (RMSE) ratio* and *the sensitivity ratio* to respectively measure the performance in terms of RMSE and the robustness in terms of sensitivity of an underlying SSA-LSTM algorithm to the variations in the number of cells in the hidden layers of the LSTM parts of the SSA-LSTM algorithms.

For computational tractability, we constrained the testing the LSTM structures with one hidden layer for the unstacked LSTM [6] structure and with two hidden layers for the stacked LSTM [25] structure. The turbofan engine datasets from Prognostics Center of Excellence (PCoE) at Ames Research Center [26] were used for all of the testings. The decomposed and reconstructed datasets for LSTM were compiled by SSA [21-23] using the process of prototype dataset selection following the methodology presented in [13, 14].

The organization for the remaining of the paper is as follows. In Section 2, the definition of the Singular Spectrum Analysis (SSA) algorithm as well as that of the trajectory matrix of a dataset, the window length and the basis functions are presented. The Long Short Term Memory (LSTM) structures are also explained in this section. In Section 3, the SSA-LSTM algorithm of [13] for RUL estimation is revisited. Section 4 describes the metrics for robustness measuring, i.e., the RMSE ratio and the sensitive ratio of an optimized or a non-optimized SSA-LSTM algorithm. In Section 5, sensitivity testing and evaluation results are provided. Finally, the conclusion is provided in Section 6. For ease of reference, a list of symbols used in this paper is given in Table 1.

Table 1. The list of symbols

Symbol	Description
$x(t)$	Time series data
t	Index of time series data
T	End of index of time series data
L	Window length of the SSA algorithm
K	Embedding dimension
$\boldsymbol{X}$	Trajectory matrix
$\boldsymbol{U}, \boldsymbol{V}$	Eigenvector matrices of $\boldsymbol{X}$
$\boldsymbol{D}$	Eigenvalue matrix of $\boldsymbol{X}$
σ	Singular value of $\boldsymbol{X}$
$\theta_k(t)$	The k^{th} basis function
$\hat{x}(t)$	Total reconstruction of time series
$x'^{(f)}(t)$	Raw feature data
$x^{(f)}(t)$	Normalized feature data
$\hat{x}^{(f)}(t)$	Reconstructed feature data
$x'^{(RUL)}(t)$	Raw RUL data
$\hat{y}^{(RUL)}(t)$	RUL estimate
$c(t)$	Internal state of LSTM
$h(t)$	Hidden layer state of LSTM
W	Weight parameter of LSTM
b	Bias of gate of LSTM
$g(t)$	Input node of LSTM
$i(t), f(t), o(t)$	Input, forget and output gates of LSTM Network respectively
N, M	Maximum of cells of unstacked and stacked LSTM respectively
n	The n^{th} cell of unstacked LSTM
m	The m^{th} cell of stacked LSTM
$\tilde{E}_1, \tilde{E}_2$	Tests of RMSE of unstacked and stacked LSTM respectively
$\tilde{\sigma}_1^2, \tilde{\sigma}_2^2$	Tests of variance of unstacked and stacked LSTM respectively
$\hat{E}$	The summation of errors of $\tilde{E}_1$ and $\tilde{E}_2$
$\hat{\sigma}^2$	The summation of variances of $\tilde{\sigma}_1^2$ and $\tilde{\sigma}_2^2$
ΔE	The RMSE ratio
$\Delta\sigma^2$	The sensitivity ratio

2. The methodologies of the singular spectrum analysis and the long short-term memory neural networks

2.1 The singular spectrum analysis (SSA)

The Singular Spectrum Analysis (SSA) is a statistical model for analysis of general time series [21]. It can be applied to stationary as well as non-stationary time series since no a priori information about the data structure is required [22, 23]. The singular value decomposition (SVD) [27] is the base model of SSA, and the trajectory matrix is applied to the process of SVD in the first step of SSA.

The trajectory matrix of a times series $x(t),\ t = 1,2,\ldots,T$, is arranged as an $L \times K$ Hankel matrix as

$$\boldsymbol{X} = \begin{bmatrix} x(1) & x(2) & x(3) & \cdots & x(K) \\ x(2) & x(3) & x(4) & \cdots & x(K+1) \\ x(3) & x(4) & x(5) & \cdots & x(K+2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x(L) & x(L+1) & x(L+2) & \cdots & x(T) \end{bmatrix} \quad (1)$$

where L is the window length and K is the embedding dimension which can be described as $K = T - L + 1$. The methods of selecting the optimized window length are proposed in [13] for $L = 24$, in [21] for $L = T/2$, in [28] for $L = T/4$ and in [29] for $L = log(T)^c$. For the rest of the paper, the SSA-LSTM algorithm of [13] whose window lengths defined by these optimized window lengths are then called the *optimized* SSA-LSTM algorithms whereas those with any other window-lengths are called the *non-optimized* SSA-LSTM algorithms.

The singular value decomposition (SVD) of **X** can be derived as follows:

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T \quad (2)$$

where $\boldsymbol{U} = (\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_L)$ and $\boldsymbol{V} = (\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_K)$ are the left and right eigenvector matrices respectively. The matrix of eigenvalues is denoted as $\boldsymbol{D} = diag(\sigma_1, \sigma_2, \ldots, \sigma_L)$, where σ_i is the i^{th} element of $\boldsymbol{D}$ [22], and the value of σ_i is ordered in the decreasing magnitude, i.e., $\sigma_1 > \sigma_2 >, \ldots, > \sigma_L$ [21].

The reconstruction of SSA for synthesis the one-dimension time series by $\boldsymbol{X}_i$ can be performed as grouping of matrix $x_{l,k}^{(i)}$ where $x^{(i)}$ is the group of the eigentriple $\boldsymbol{X}_i$ from the set of indices $i = 1, \ldots, d$ into m disjoint subsets, i.e., $I_1, \ldots, I_m$, and $d = rank(\boldsymbol{X})$ is the number of nonzero eigenvalues σ_i. The j^{th} subset has p components, i.e., $I_j = \{j_1, \ldots, j_p\}$. If $m = d$, the grouping is called the elementary grouping and each group has only one member, i.e., $I_j = \{j\}$, $j = 1, \ldots, d$. So the trajectory matrix can be decomposed as $\boldsymbol{X} = \boldsymbol{X}_{I_1} + \cdots + \boldsymbol{X}_{I_m}$ where $\boldsymbol{X}_{I_j} = \boldsymbol{X}_{j_1} + \cdots + \boldsymbol{X}_{j_p}$. The diagonal averaging of $x_{l,o}^{(k)}$, $1 \leq l \leq L$ and $1 \leq o \leq K$, of $\boldsymbol{X}_{I_k}$ for $I_k = I_1, \ldots, I_m$ can provide the element of the one-dimension time series of SSA $\theta_i(t), i = 1, \ldots, L$ and $t = 1, \ldots, T$, and the diagonal averaging operation is defined as:

$$\theta_k(t) = \begin{cases} \frac{1}{t}\sum_{q=1}^{t} x_{q,t-q+1}^{(k)} & ; 1 \leq t < L \\ \frac{1}{L}\sum_{q=1}^{L} x_{q,t-q+1}^{(k)} & ; L \leq t < K. \\ \frac{1}{T-t+1}\sum_{q=t-K+1}^{L} x_{q,t-q+1}^{(k)} & ; L \leq t < K \end{cases} \quad (3)$$

where $\theta_k(t),\ k = 1, \ldots, m$ and $t = 1, \ldots, T$, is denoted as the k^{th} *basis function* of the SSA decomposition, and the reconstruction of time series $\hat{x}(t)$ describes as the total of all order $\theta_k(t)$ at $k = 1, \ldots, m$ is determined by the summation of the basis functions,

$$\hat{x}(t) = \sum_{k=1}^{m} \theta_k(t). \tag{4}$$

2.2 The long short-term memory (LSTM)

The Long Short-Term Memory (LSTM) is an upgraded type of Recurrent Neural Networks (RNNs) and was introduced by Hochreiter and Schmidhuber [6]. The diagram of an LSTM cell is described in Fig. 1. An LSTM network performs forgetting, ignoring and selecting operations involving previous cell state $c(t-1)$, the previous output of the hidden layer $h(t\ -1)$ and the current input $x(t)$ to generate the prediction of the current cell state and the current hidden layer, i.e., $c(t), h(t)$, respectively. All the above-mentioned operations are associated with weights for state (W^x), weights for hidden layer input (W^h) and bias of the gate (b) and nonlinear activation function either a tanh (ϕ) or a sigmoid (σ) function [8, 30].

From Fig. 1, the elements of the cell, i.e., input node $g(t)$, input gate $i(t)$, forget gate $f(t)$, internal state $c(t)$ and output gate $o(t)$, are described as follows [8]:

$$g(t) = \emptyset\left(W^{gx}x(t) + W^{gh}h(t-1) + b_g\right), \tag{5}$$

$$i(t) = \sigma\left(W^{ix}x(t) + W^{ih}h(t-1) + b_i\right), \tag{6}$$

$$f(t) = \sigma\left(W^{fx}x(t) + W^{fh}h(t-1) + b_f\right), \tag{7}$$

$$o(t) = \sigma\left(W^{ox}x(t) + W^{oh}h(t-1) + b_o\right), \tag{8}$$

$$c(t) = g(t) \otimes i(t) + c(t-1) \otimes f(t), \tag{9}$$

$$h(t) = \emptyset(c(t)) \otimes o(t), \tag{10}$$

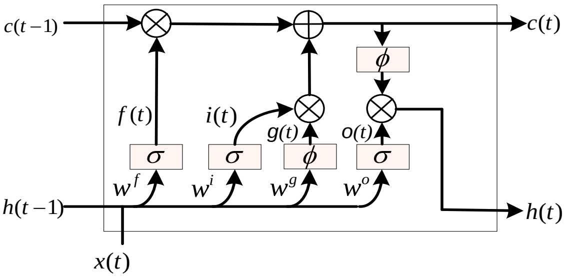


Figure. 1 The diagram of an LSTM cell

where $\otimes$ is the element wise multiplication which plays an important role in an LSTM network. First, it acts like a gate to the memory section to remember in forgetting particular information. It is also used as a selection gate to filter unrelated information before being released as a prediction. Finally, it performs as an ignoring gate for unimportant information containing in the output.

Three diagrams of LSTM structurers are shown in Fig. 2(a) to 2(c). In Fig. 2(a), an unstacked LSTM structure with N LSTM cells is shown where cell (N) represent the N^{th} cell. This is basically a cascaded version of multiple LSTM cells shown in Fig. 1. In Fig. 2(b), the stacked LSTM layer is described as the two hidden layers in LSTM neural networks and $h^{(1)}(t)$ and $h^{(2)}(t)$ are the output values of the first and second hidden layer respectively. In Fig. 2(c), the LSTM prediction network is set in the sequence to sequence prediction model by LSTM neural network [30].

3. The model for performance and robustness evaluation of the SSA-LSTM algorithm

The amalgamation of the SSA and LSTM neural networks has been studied in [13, 14] resulting in the SSA-LSTM algorithm. The accuracy of RUL estimation of various algorithms is investigated in [13] where the comparison results show that LSTM can offer the best performance compared to other neural network structures. The RUL datasets of multiple turbofan engines time series [26] have the same trendy profiles and there are 200 datasets recorded from several sensors under different conditions of both normal and fault modes. In [14], the application of features for the turbofan datasets is described.

Since the features of each sensor in dataset have different scales, normalization is needed before all the features are used. The normalized feature $x^{(f)}(t)$ is represented as:

$$x^{(f)}(t) = \frac{x'^{(f)}(t) - x'^{(f)}_{\min}(t)}{x'^{(f)}_{\max}(t) - x'^{(f)}_{\min}(t)}, \tag{11}$$

where $x'^{(f)}(t)$ is the raw feature data with its the minimum value and maximum value are described as $x'^{(f)}_{\min}(t)$ and $x'^{(f)}_{\max}(t)$ respectively.

The normalized feature $x^{(f)}(t)$ is included in the first step of system model at process of SSA for RUL estimation by LSTM in Fig. 3.

In Fig. 3, a structural illustration for the performance and robustness evaluation of the SSA-

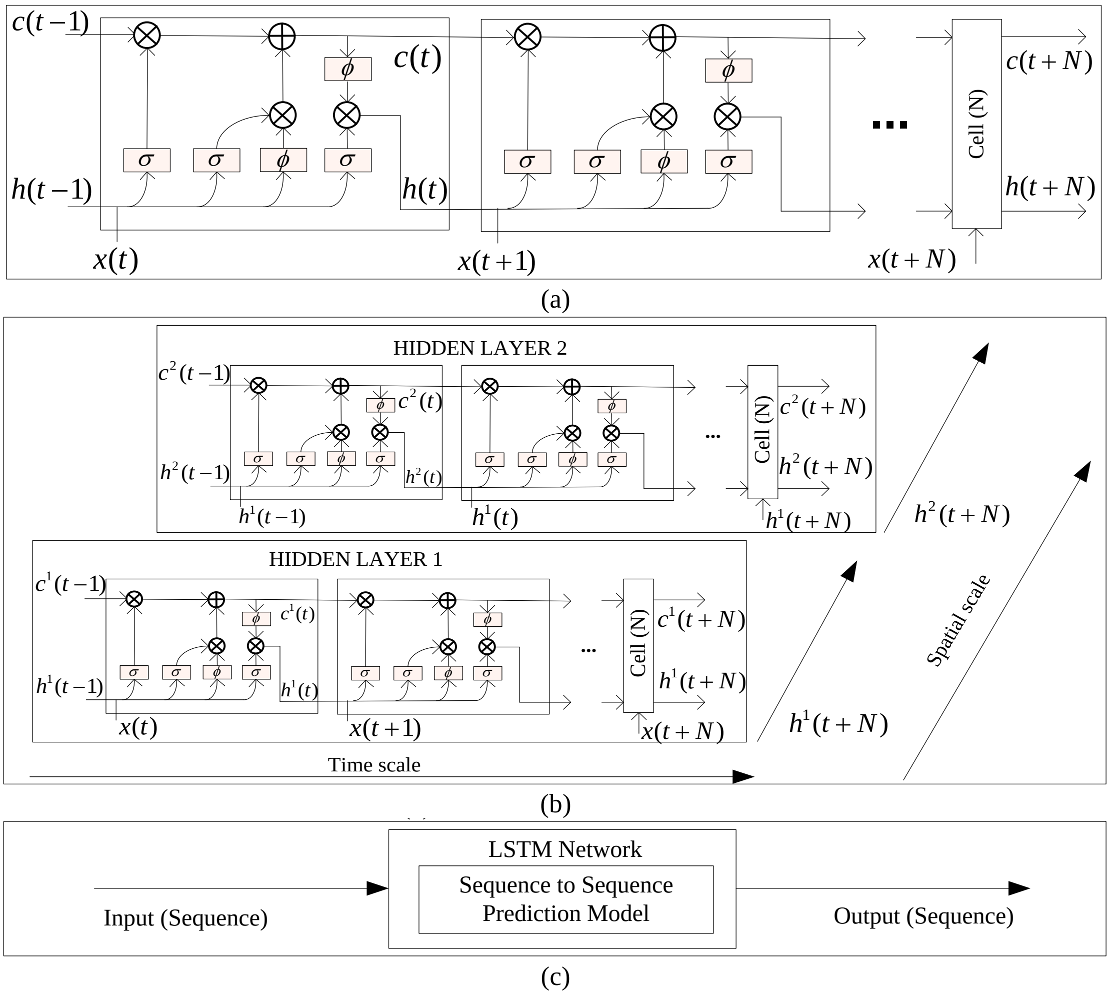


Figure. 2 The diagram of: (a) the unstacked LSTM, (b) the stacked LSTM, and (c) the LSTM prediction

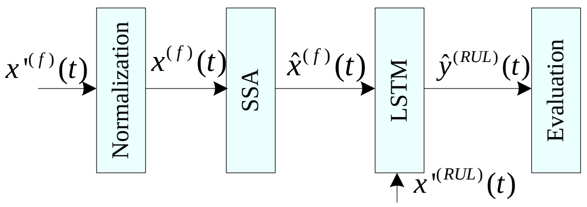


Figure. 3 The system model for performance and robustness evaluation of an SSA-LSTM algorithm

LSTM algorithm of [13, 14] are shown where $\hat{x}^{(f)}(t)$ is the reconstructed feature from process of reconstruction of SSA, $x'^{(RUL)}(t)$ the raw RUL data and $\hat{y}^{(RUL)}(t)$ the result of RUL estimation by this system model. We use the root mean square error (RMSE) and the variance of the RMSEs as metrics for the performance evaluation. The RMSEs of an unstacked LSTM case with n cells are denoted by the random variables $RMSE_n, n = 1, \dots, N$ where N is the maximun number of LSTM cells. The RMSEs for a stacked LSTM case with n cells of the first hidden layer and m cells of the second hidden layer are denoted by the random variable $RMSE_n^m, n = 1, \dots, N$ and $m = 1, \dots, M$ where N and M are the maximum numbers of the first hidden layer and the second hidden layer of LSTM cells respectively.

4. The metrics of performance and robustness

The performance and robustness of an SSA-LSTM algorithm are evaluated by two metrics: The RMSE ratio and the sensitivity ratio. The RMSE ratio is the ratio of the total RMSE of the underlying SSA-LSTM algorithm to that of the baseline prototype dataset. The sensitivity ratio is the ratio of the total variance of the underlying SSA-LSTM algorithm to that of the baseline prototype dataset. Both metrics are evaluated as a result from varying the number of cells and/or the number of hidden layers of the LSTM part in the SSA-LSTM structure.

For the unstacked LSTM case with N cells, we first calculate the test of RMSE ($\tilde{E}_1$) which is basically the averaged RMSE of N cells and the test of variance ($\tilde{\sigma}_1^2$) as the variance of the RMSEs of N cells. The equations of $\tilde{E}_1$ and $\tilde{\sigma}_1^2$ for the unstacked

LSTM architecture with N cells are defined as follows:

$$\tilde{E}_1 = \frac{\sum_{n=1}^{N} RMSE_n}{N}, \tag{12}$$

$$\tilde{\sigma}_1^2 = \sigma^2(RMSE_N), \tag{13}$$

where $\sigma^2(\cdot)$ is the variance operator. For the stacked LSTM case with two hidden layers with N and M cells for the first and second hidden layer repectively, the test of RMSE $\tilde{E}_2$ and the test of variance $\tilde{\sigma}_2^2$ are derived by averaging respectively the test of RMSE $\tilde{E}_1$ Eq. (12) and the test of variance $\tilde{\sigma}_1^2$ Eq. (13) up to M cells of the second hidden layer:

$$\tilde{E}_1^m = \frac{\sum_{n=1}^{N} RMSE_n^m}{N}, \tag{14}$$

$$\tilde{E}_2 = \frac{\sum_{m=1}^{M} \tilde{E}_1^m}{M}, \tag{15}$$

$$\tilde{\sigma}_2^2 = \frac{\sum_{m}^{M} \tilde{\sigma}_1^{2,m}}{M}, \tag{16}$$

where $\tilde{E}_1^m$ is the average of $\tilde{E}_1$ of Eq. (12) and $\tilde{\sigma}_1^{2,m}$ is the test of variance Eq. (13) considered up to m cells of the second hidden layer. For the stacked LSTM case, the total RMSE and the total variance are first defined respectively as:

$$\hat{E} = \tilde{E}_1 + \tilde{E}_2, \tag{17}$$

$$\hat{\sigma}^2 = \tilde{\sigma}_1^2 + \tilde{\sigma}_2^2, \tag{18}$$

where $\hat{E}$ is the summation of errors of $\tilde{E}_1$ and $\tilde{E}_2$ whereas $\hat{\sigma}^2$ the summation of variances of $\tilde{\sigma}_1^2$ and $\tilde{\sigma}_2^2$. Note that for the unstacked LSTM case, the total RMSE $\hat{E}$ and the total variance $\hat{\sigma}^2$ are simply $\tilde{E}_1$ and $\tilde{\sigma}_1^2$ respectively.

With Eqs. (17) and (18), we define $\hat{E}_{(prototype)}$ as the total RMSE and $\hat{\sigma}^2{}_{(prototype)}$ the total variance derived from the prototype dataset. Also, we define $\hat{E}_{(SSA-LSTM)}$ as the total RMSE and $\hat{\sigma}^2{}_{(SSA-LSTM)}$ as the total variance from any of an SSA-LSTM algorithm that uses either a non-optimized (arbitrary) window length or an optimized window length from one of the algorithms proposed in [13, 21, 28, 39].

We define the RMSE ratio (ΔE) as the ratio of $\hat{E}_{(SSA-LSTM)}$ to $\hat{E}_{(prototype)}$ and the sensitivity ratio ($\Delta\sigma^2$) as the ratio of $\hat{\sigma}^2{}_{(SSA-LSTM)}$ to $\hat{\sigma}^2{}_{(prototype)}$, i.e.,

$$\Delta E = \frac{\hat{E}_{(SSA-LSTM)}}{\hat{E}_{(prototype)}}, \tag{19}$$

$$\Delta\sigma^2 = \frac{\hat{\sigma}^2{}_{(SSA-LSTM)}}{\hat{\sigma}^2{}_{(prototype)}}. \tag{20}$$

In Algorithm 1, the performance and robustness of the SSA-LSTM algorithms are evaluated by means of the RMSE ratio and the sensitivity ratio. At different combinations of the number of cells and the number of hidden layers of the LTSM part, the RMSE ratio and the sensitivity ratio of an SSA-LSTM algorithm are computed for both the unstacked and stacked (with two hidden layers) LSTM structures. Algorithm 1 is divided into two conditions: the unstacked LSTM case and the stacked LSTM case. For the unstacked LSTM case, the number of hidden layer is one, i.e., $i_{max} = 1$ and the test of RMSE $\tilde{E}_1$ and the test of variance $\tilde{\sigma}_1^2$ are calculated by Eqs. (12) and (13) respectively.

The RMSE ratio ΔE and the sensitivity ratio $\Delta\sigma^2$ are then calculated by Eqs. (19) and (20) the maximum number of LSTM cells of the first hidden layer is 100 ($N = 100$). The second condition is for the stacked LSTM case where two hidden layers are considered, i.e., $i_{max} = 2$. In this case, the algorithm first calculates $\tilde{E}_n^m$ and $\tilde{\sigma}_1^{2,m}$ where the maximum number of LSTM cells of the first and the second hidden layers are 100 and 10 respectively, i.e., $N = 100$ and $M = 10$. The second step is to compute $\tilde{E}_2$ by Eq. (15) and $\tilde{\sigma}_2^2$ by Eq. (16). The third step is summing the values of $\hat{E}$ by Eq. (17) and $\hat{\sigma}^2$ by Eq. (18).

Algorithm 1: Algorithm for evaluating the performance and robustness of the SSA-LSTM algorithm.

```
Input:   $x^{(f)}(t)$ // Feature time series data.
         $x^{(RUL)}(t)$ // RUL time series data.
Define:  $\hat{x}^{(f)}(t)$ // Synthetic feature time series.
         $i\_max$ // The maximum number of hidden layers.
         $n$ // The $n^{th}$ cell of the first hidden layer
         $m$ //The $m^{th}$ cell of the second hidden layer.
         $N$ // The number of cells of the first hidden layer to be computed.
         $M$//The number of cells of the second hidden layer to be computed.
         RMSE // Root mean square error.
         $\tilde{E}$ // The test of RMSE.
         $\tilde{\sigma}^2$ // The test of variance.
         $\hat{E}$// The accumulating error.
         $\hat{\sigma}^2$// The accumulating variance.
```

Received: May 11, 2022. Revised: June 29, 2022.

```
Output: $\Delta E$ // the RMSE ratio.
        $\Delta\sigma^2$ // the sensitivity ratio.
Analysis and Synthesis of $x^{(f)}(t)$ by SSA.
Training LSTM neural network by $\hat{x}^{(f)}(t)$ and $x'^{(RUL)}(t)$.
$i\_max <= 2$
For $m$ to 10
  For $n$ to 100
    If $i\_max = 1$
        Evaluate error of testing LSTM neural network by $RMSE_n, n = 1, \dots, N$.
      End
      Else
        Evaluate error of testing LSTM neural network by $RMSE_n^m, n = 1, \dots, N$ and $m = 1, \dots, M$.
      End
    End
End
Compute $\tilde{E}_1$ by (12) and $\tilde{\sigma}_1^2$ by (13).
Compute $\tilde{E}_2$ by (15) and $\tilde{\sigma}_2^2$ by (16).
Summing value of $\hat{E}$ by (17) and $\hat{\sigma}^2$ by (18).
Evaluate $\Delta E$ by (19) and $\Delta\sigma^2$ by (20).
```

Finally, the last step is to evaluate the performance of the underlying SSA-LSTM in the RMSE ratio ΔE by Eq. (19) and in the sensitivity ratio $\Delta\sigma^2$ by Eq. (20) for different numbers of cells and hidden layers. The outputs ΔE and $\Delta\sigma^2$ are used to compare between the window length optimized and non-optimized SSA-LSTM algorithms as discussed in the next Section.

5. Testing and evaluation results

First, from the 200 turbofan engine datasets of [27], we selected a prototype as a dataset which provides the lowest RUL estimation error determined by the LSTM network [13]. The RMSEs of the RUL estimation of the prototype using an unstacked LSTM structure as the number of LSTM cells varied from one to 100 are plotted in Fig.4. This result showed that there was no straightforward relationship between the number of cells and the RMSE. It was noted that the maximum RMSE of 27.49 was obtained at 93 cells and the minimum RMSE of 19.62 at 16 cells.

We then tested the effect of the window length variations to the SSA-LSTM algorithms with the unstacked LSTM architecture in the LSTM part as the number of LSTM cells varied.

First, we considered four methods of selecting the optimized window length from [13, 21, 28, 29] for the SSA part in the SSA-LSTM algorithm described in [13]. The first optimized SSA-LSTM uses the window length $L = T/2$ from [21] where the second to the fourth ones were $L = 24$ [13], $L = log(T)^C$ [29] and $L = T/4$ [28] respectively. We from now on call the SSA-LSTM algorithm of [13] with its window length L in the SSA part from [13], [21], [28], or [29] as the *optimized SSA-LSTM* algorithm.

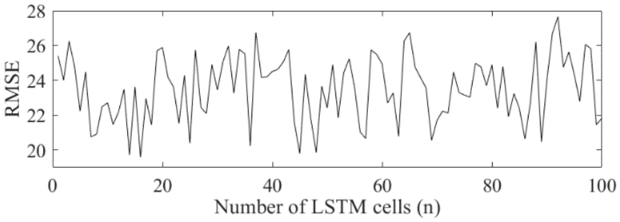


Figure. 4 The RMSE profile of estimated RUL for the prototype estimated by the LSTM algorithm as the number of LSTM cells varied from one to 100

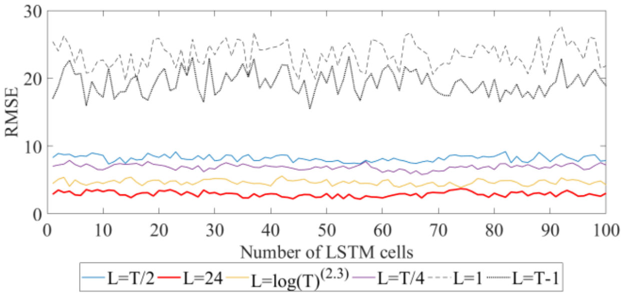
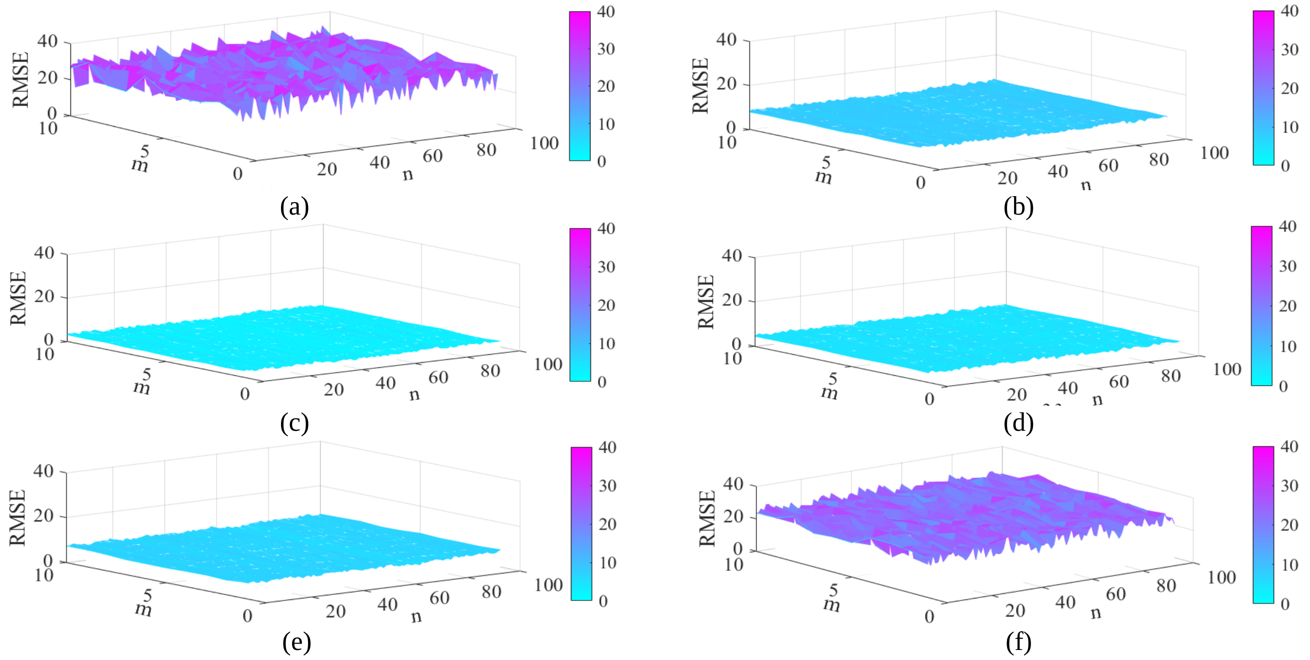


Figure. 5 The RMSE profiles of the estimated RULs for the prototype estimated by the SSA-LSTM algorithm as the number of LSTM cells varied from 1 to 100 of the four optimized window lengths, i.e., $L = T/2$ [21], $L = 24$ [13], $L = log(T)^C$ [29] and $L = T/4$ [28] and two non-optimized ones, i.e., $L = 1$ and $L = T - 1$

For arbitrary window length L SSA-LSTM, it is called as the *non-optimized SSA-LSTM* algorithm. To compare with the optimized SSA-LSTM, we adopted $L = 1$ and $L = T - 1$ for the non-optimized SSA-LSTM algorithms. The choice of $L = 1$ was chosen to revert the non-optimized SSA-LSTM algorithm to the prototype-based LSTM algorithm and $L = T - 1$ was the largest window length that can be taken.

In Figure. 5, the RMSE profiles of the estimated RUL results of the four optimized SSA-LSTM algorithms with $L = T/2$ [21], $L = 24$ [13], $L = log(T)^C$ [29] and $L = T/4$ [28] and two non-optimized SSA-LSTM algorithms, i.e., $L = 1$ and $L = T - 1$, are shown as the number of cells varied from 1 to 100. It was shown that, as the number of LSTM cells varied, the RMSE plots of the non-optimized SSA-LSTM algorithms were highly fluctuated while those of the optimized SSA-LSTM



(a) (b) (c) (d) (e) (f)

Figure. 6 The RMSE profiles of the estimated RULs of the optimized and non-optimized SSA-LSTM algorithms with the two hidden layered stacked LSTM architecture. All plots are performed as the numbers of cells of both hidden layers vary where n and m as the number of cells in first layer and in second layer respectively. The maximum number of cells of first layer is 100 and that of the second layer is 10. The RMSE plots of the non-optimized SSA-LSTM algorithms are shown in (a) for $L = 1$ and in (f) for $L = T - 1$ and those of the optimized SSA-LSTM algorithms are shown in (b) for $L = T/2$, in (c) for $L = 24$, in (d) for $L = log(T)^{2.3}$ and in (e) for $L = T/4$

Table 2. The test of variance ($\tilde{\sigma}^2$) for RMSEs of estimated RULs for a prototype with the optimized SSA-LSTM algorithms, i.e., $L = T/2$ [21], $L = 24$ [13], $L = log(T)^C$ [29] and $L = T/4$ [28] and two non-optimized SSA-LSTM algorithms, i.e., $L = 1$ and $L = T - 1$, as the number of LSTM cells varied from 1 to 100

Algorithms	$\tilde{\sigma}^2$
$L = 1$	3.657
$L = T/2$	0.213
$L = 24$	0.136
$L = log(T)^{2.3}$	0.140
$L = T/4$	0.163
$L = T - 1$	3.233

ones were mostly leveled off. It was also interesting to see that the RMSEs of the four optimized SSA-LSTM algorithms were significantly lower than those of the two non-optimized ones. As compared to all algorithms, the non-optimized $L = 1$ SSA-LSTM algorithm produced the highest RMSEs for all numbers of cells and was followed by the $L = T - 1$ non-optimized one. In contrast, the optimized SSA-LSTM algorithm with $L = 24$ from [13] offered the lowest RMSEs for all numbers of cells in both hidden layers.

We then performed the test of variance of the RMSEs ($\tilde{\sigma}^2$) by (13) for all algorithms and the results are shown in Table 2. It was clearly shown that $\tilde{\sigma}^2$ of all the optimized SSA-LSTM algorithms were significantly lower than those of the $L = 1$ and $L = T - 1$ non-optimized ones. Also, of all the optimized counterparts, $\tilde{\sigma}^2$ of the $L = 24$ optimized SSA-LSTM algorithm from [13] was the lowest.

We then progressed to consider the effect of the window length to the SSA-LSTM algorithm with the stacked LSTM architecture as the number of cells for both hidden layers varied. In this case, the RMSEs of the RUL estimates derived from four optimized and two non-optimized SSA-LSTM algorithms were evaluated at the combinational number of cells of both hidden layers. The RMSE profiles of each algorithm are plotted in Fig. 6(a) to 6(f) where the maximum number of cells in the first hidden layer was 100 and that of the second hidden layer was 10. The RMSE profiles of the non-optimized SSA-LSTM algorithms are shown in Fig. 6(a) and 6(f) for $L = 1$ and $L = T - 1$ respectively whereas those for the optimized ones are shown in Fig. 6(b), 6(c), 6(d) and 6(e) for $L = T/2$, $L = 24$, $L = log(T)^{2.3}$and $L = T/4$ respectively. Two remarks can be made from Fig. 6(a) to 6(f). First, it is clearly shown that the RMSE profiles of the optimized SSA-LSTM algorithms as shown in Fig. 6(b) to 6(e) were significantly lower than those of

Received: May 11, 2022. Revised: June 29, 2022.

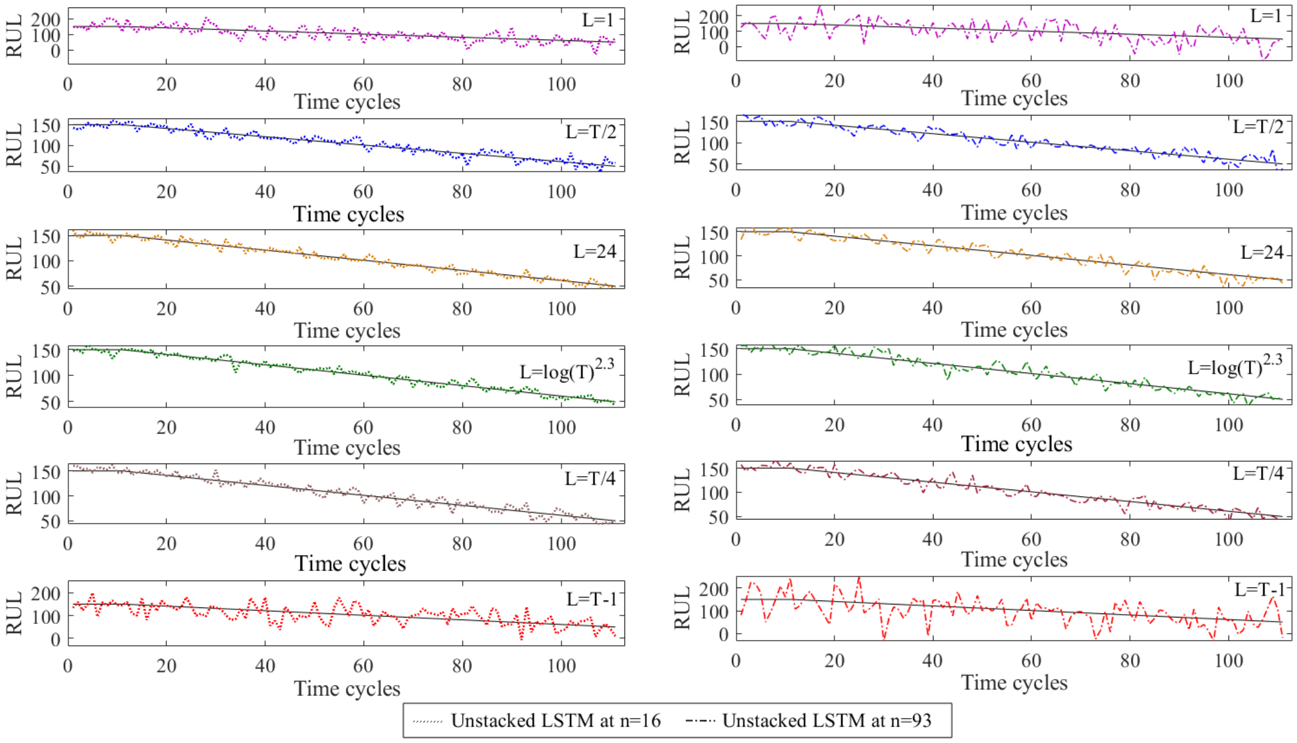


Figure. 7 The estimated RULs of the four optimized and the two non-optimized SSA-LSTM algorithms with the unstacked LSTM architecture are plotted in comparison with the true RULs plotted in the black solid lines. All the plots in the left column are for $n = 16$ and those in the right column are for $n = 93$. The plots in both columns of the first row to the sixth row in both columns are for the $L = 1$, $L = T/2$, $L = 24$, $L = log(T)^{2.3}$, $L = T/4$ and $L = T - 1$ respectively

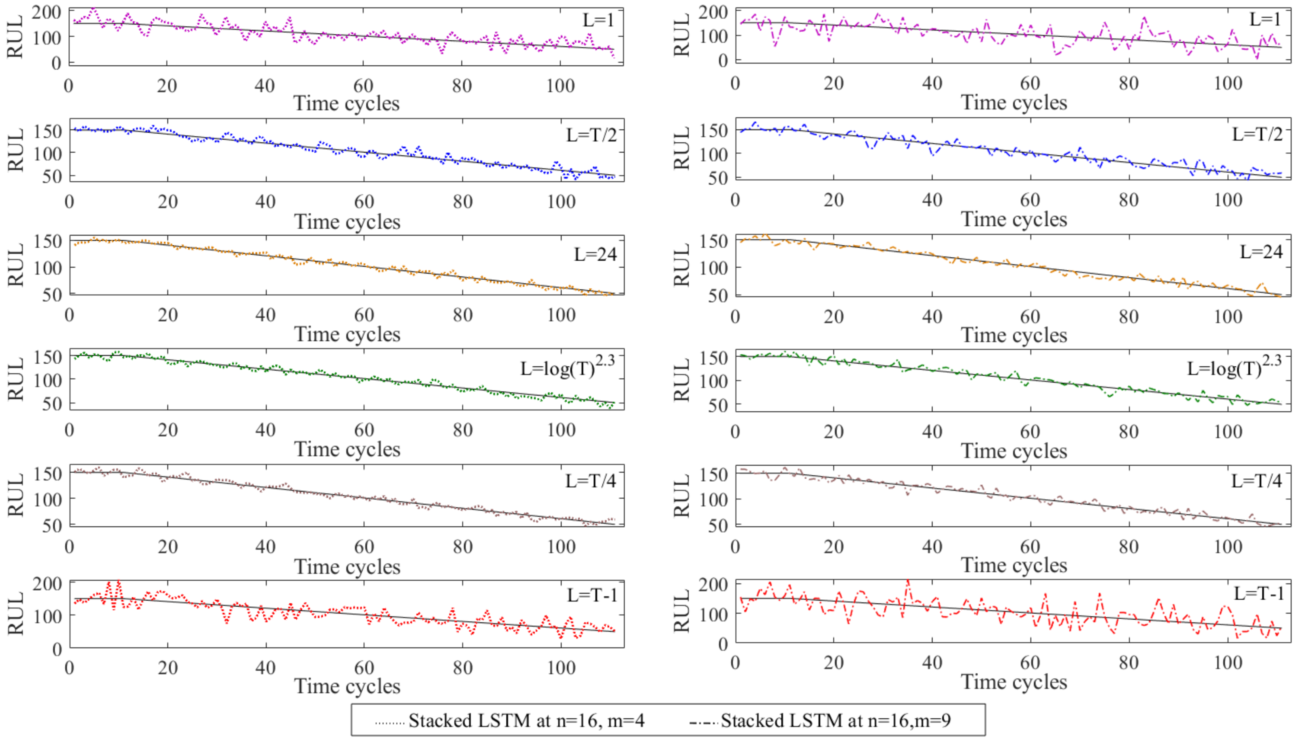


Figure. 8 The estimated RULs of the four optimized and the two non-optimized SSA-LSTM algorithms with the stacked LSTM structure are plotted in comparison with the true RULs plotted in the black solid lines. All the plots in the left column are for $n = 16$ and $m = 4$ and those in the right column are for $n = 16$ and $m = 9$. The plots in both columns of the first row to the sixth row in both columns are for the $L = 1$, $L = T/2$, $L = 24$, $L = log(T)^{2.3}$, $L = T/4$ and $L = T - 1$ respectively

the non-optimized ones in Fig. 6(a) and 6(f). Second, the RMSE profiles of all the optimized SSA-LSTM algorithms were much more leveled off than those of the non-optimized ones. Both remarks leads to a suggestion that, once the window length L is optimized, the SSA-LSTM algorithm can be much less sensitive to the choice of combinational numbers of LSTM cells of either layers.

To illustrate the relationship between the performance in RUL estimation and the number of cells in the hidden layers of the LSTM part, the estimated RULs of the optimized and non-optimized SSA-LSTM algorithms were compared with the true RUL evaluated with the unstacked and stacked LSTM structures in Fig. 7 and 8 respectively. In Fig.7, the unstacked LSTM case, the number of LSTM cells n was chosen according to the prototype-based LSTM algorithm in Fig. 4 where $n = 16$ was associated with the best performance and $n = 93$ with the worst performance. It was noticed that for the $L = 1$ and $L = T - 1$ non-optimized SSA-LSTM algorithms with larger numbers of cells at $n = 93$, all the plots of are noisier than those with smaller number of cells at $n = 16$. However, the RUL estimates for an optimized SSA-LSTM algorithm at the numbers of cells at $n = 16$ and at $n = 93$ were comparable to each other and were also noticeably less noisy than those of their non-optimized counterparts.

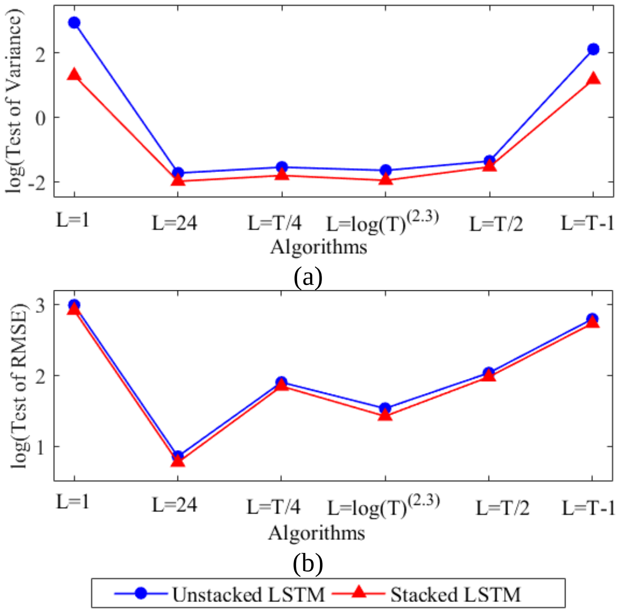


Figure. 9 (a) The tests of variance and (b) the tests of RMSE of the non-optimized ($L = 1$ and $L = T - 1$) and the optimized ($L = T/2$, $L = 24$, $L = log(T)^{2.3}$ and $L = T/4$) SSA-LSTM algorithms are shown for the unstacked and the stacked LSTM structures

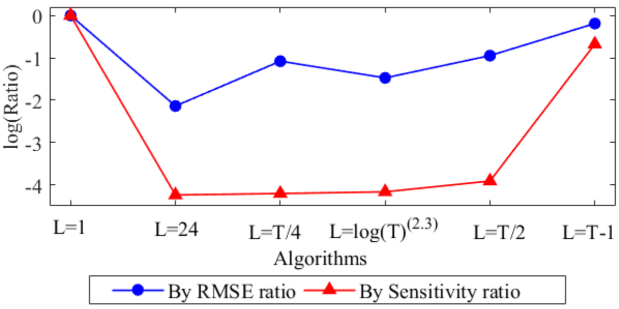


Figure. 10 The RMSE ratio and the sensitivity ratio of all SSA-LSTM algorithms are shown of the stacked LSTM architecture

Similar to the unstacked LSTM case in Fig. 7, the estimated RULs of all SSA-LSTM algorithms with the stacked LSTM architecture were compared for two combinational numbers the same numbers of cells of in the first and second hidden layers are plotted in Fig. 8.

In Fig. 8, those of the stacked SSA-LSTM algorithms with $n = 16$ and $m = 4$ are plotted in the left column and those with $n = 16$ and $m = 4$ are in the right column. Note that for both of the non-optimized $L = 1$ and $L = T - 1$ SSA-LSTM algorithms, the RUL estimates at the larger number of cells in the second hidden layer ($n = 16$ and $m = 9$) were noisier than those at the smaller one ($n = 16$ and $m = 4$). For any of the optimized SSA-LSTM algorithms, however, the level of noise of their RUL estimates were comparable regardless of the number of cells of the second hidden layer. Also, as in the unstacked case, the RUL estimates were also noticeably less noisy than those of the non-optimized counterparts.

For the non-optimized SSA-LSTM algorithms, both Fig. 7 and 8 show that there is no straightforward relationship between the number of LSTM cells in either hidden layers and their RUL estimation performance: Using more cells does not guarantee a better performance. However, the performance in RUL estimation of the optimized SSA-LSTM algorithm was not sensitive to the choices of the numbers of cells in either hidden layers.

In Fig. 9 (a) and (b), the test of variance and the test of RMSE of all SSA-LSTM algorithms are shown respectively. Note that the y-axes of the plots in Fig. 9 (a) and 9(b) are in logarithmic scales in order to cover large differences of the maximum and the minimum values produced by all algorithms. In Fig. 9(a), the tests of variance of each algorithm for the unstacked ($\tilde{\sigma}_1^2$) and the stacked ($\tilde{\sigma}_2^2$) LSTM structures were computed by Eqs. (13) and (16)

Table 3. The test of variance ($\tilde{\sigma}^2$) and the test of RMSE ($\tilde{E}$) of each SSA-LSTM algorithm for the unstacked ($\tilde{\sigma}_1^2$ and $\tilde{E}_1$) and stacked ($\tilde{\sigma}_2^2$ and $\tilde{E}_2$) LSTM structures

Algorithm	$\tilde{\sigma}^2$		$\tilde{E}$	
	$\tilde{\sigma}_1^2$	$\tilde{\sigma}_2^2$	$\tilde{E}_1$	$\tilde{E}_2$
$L = 1$	9.024	3.657	19.824	18.418
$T/2$	0.255	0.213	7.618	7.196
$L = 24$	0.176	0.136	2.346	2.159
$L = log(T)^{2.3}$	0.191	0.140	4.618	4.138
L=T/4	0.211	0.163	6.674	6.316
L=T-1	8.31	3.233	16.298	15.312

Table 4. The total RMSE ($\hat{E}$) and the total variance ($\hat{\sigma}^2$) of each SSA-LSTM algorithm for the stacked LSTM architecture

Algorithm	$\hat{E}$	$\hat{\sigma}^2$
$L = 1$	38.242	1.442
$T/2$	14.814	0.753
$L = 24$	4.505	0.405
$L = log(T)^{2.3}$	8.756	0.674
$L = T/4$	12.99	0.497
$L = T - 1$	31.61	1.297

Table 5. The RMSE ratio (ΔE) and the sensitivity ratio ($\Delta\sigma^2$)

Algorithm	ΔE	$\Delta\sigma^2$
$L = 1$	100%	100%
$T/2$	38.737%	2.067%
$L = 24$	11.780%	1.444%
$L = log(T)^{2.3}$	22.896%	1.555%
$L = T/4$	33.967%	1.495%
$L = T - 1$	82.657%	50.893%

respectively. Both $\tilde{\sigma}_1^2$ and $\tilde{\sigma}_2^2$ of all the optimized SSA-LSTM algorithms, i.e., $L = T/2$, $L = 24$, $L = log(T)^{2.3}$ and $L = T/4$, were much lower than those of the $L = 1$ and $L = T - 1$ non-optimized ones and were comparable to each other for the same unstacked or stacked architecture. The numerical values of $\tilde{\sigma}_1^2$ and $\tilde{\sigma}_2^2$ of all algorithms are listed in Table.3. The $L = 24$ optimized SSA-LSTM algorithm produced the lowest $\tilde{\sigma}_1^2$ and $\tilde{\sigma}_2^2$ at 0.176 and 0.136 respectively.

The test of RMSE of each algorithm for the unstacked ($\tilde{E}_1$) and the stacked ($\tilde{E}_2$) LSTM structures as computed by Eqs. (12) and (15) respectively are shown in Fig. 9 (b). Similar to the case of the test of variance, both $\tilde{E}_1$ and $\tilde{E}_2$ of all the optimized SSA-LSTM algorithms, i.e., $L = T/2$, $L = 24$, $L = log(T)^{2.3}$ and $L = T/4$, were much lower than those of the $L = 1$ and $L = T - 1$ non-optimized ones. However, unlike in the case of the tests of variance, the plots of $\tilde{E}_1$ and $\tilde{E}_2$ of all the optimized SSA-LSTM algorithms in the same unstacked or stacked architecture exhibited fluctuations with the $L = 24$ optimized SSA-LSTM algorithm produced the lowest $\tilde{E}_1$ and $\tilde{E}_2$ at 2.346 and 2.159 respectively.

Next, we investigated the RMSE ratio and the sensitivity ratio of all algorithms. In order to do that, the total RMSE and the total variance of the SSA-LSTM algorithm must be first evaluated. The total RMSE $\hat{E}$ was derived from the tests of RMSE $\tilde{E}_1$ and $\tilde{E}_2$ as in Eq. (17), whereas the total variance $\hat{\sigma}^2$ was from the tests of variance $\tilde{\sigma}_1^2$ and $\tilde{\sigma}_2^2$ as in Eq. (18). Both $\hat{E}$ and $\hat{\sigma}^2$ for all SSA-LSTM algorithms are shown in Table 4.

The RMSE ratio ΔE and the sensitivity ratio $\Delta\sigma^2$ were then calculated from $\hat{E}$ and $\hat{\sigma}^2$ in Eqs. (19) and (20) respectively. For relative comparisons, the values of ΔE and $\Delta\sigma^2$ were normalized by those of the $L = 1$ non-optimized SSA-LSTM algorithm which were at the maximum. All the RMSE ratios and the sensitivity ratios were ranging from 0% to 100% in the percentage form. All ΔE and $\Delta\sigma^2$ are plotted in the logarithmic scale as in Fig.10 and their values are shown in Table 5. It was shown how the optimized and non-optimized SSA-LSTM algorithms responded differently to the variation in the number of cells. Both the RMSE ratio and sensitivity ratio of any of the optimized SSA-LSTM algorithms were lower than those of the non-optimized counterparts. The RMSE ratios ΔE of the optimized SSA-LSTM algorithms were ranging from 11.780% for L=24 to 38.737% for $L = T/2$ as compared to those of the non-optimized counterparts which were 100% for $L = 1$ and 82.657% for $L = T - 1$. More interestingly, the sensitivity ratios $\Delta\sigma^2$ derived from the non-optimized and the optimized SSA-LSTM algorithms can differ by an order of magnitude. The sensitivity ratios $\Delta\sigma^2$ of the optimized SSA-LSTM algorithms were ranging from 1.444%% for $L = 24$ to 2.067% for $L = T/2$ as compared to those of the non-optimized counterparts which were 100% for $L = 1$ and 50.893% for $L = T - 1$.

6. Conclusion

In this paper, we studied the performance and robustness in RUL estimation by using the turbofan engines datasets [26] of the optimized and non-optimized SSA-LSTM algorithms of [13]. The effects of the window length optimization to the performance and robustness of the SSA-LSTM algorithms under the variation of the number of cells and the numbers of hidden layers of the LSTM structures were investigated.

For the optimized SSA-LSTM algorithms, their window lengths L were derived from one of the optimized window lengths proposed in [13, 21, 28, 29]. The optimized window lengths for RUL estimation of the turbofan engines datasets [26] for [13, 21, 28, 29] were $L = 24, L = T/2$, $L = T/4$ and $L = log(T)^c$ respectively. For a non window-length optimized SSA algorithm, the window length L can be any number from 1 to $T - 1$ where $L = 1$ and $L = T - 1$ were chosen in this paper. The choice of $L = 1$ was coincided with the method of selecting the prototype which offers the lowest RUL estimation error [13] and $L = T - 1$ was the largest window length that can be chosen.

By using the RMSE ratio ΔE and the sensitivity ratio $\Delta\sigma^2$, the performance and the robustness of the SSA-LSTM algorithms can be quantified. It was shown that the values of ΔE and $\Delta\sigma^2$ of the optimized SSA-LSTM algorithms were much lower than the non-optimized counterparts. In fact, the values of $\Delta\sigma^2$ derived from the non-optimized and the optimized SSA-LSTM algorithms can differ by an order of magnitude. In terms of the RMSE ratio, it is then confirmed that the optimized SSA-LSTM algorithms using the optimized window lengths of [13], [21], [28] or [29] can perform much better than their non-optimized window length counterparts. Likewise, in terms of the sensitivity ratios, the optimized SSA-LSTM algorithms exhibit their robustness to the perturbation caused by the variation is the numbers of cells of both unstacked and stacked LSTM structures. To summarize, the SSA-LSTM algorithm can better perform and is more robust by using any of the window length optimization method chosen from [13], [214 [28] or [29] than those without any optimization. Among all the algorithms being evaluated, the $L = 24$ optimized SSA-LSTM algorithm of [13] provided the best performance in terms of both the RMSE ratio and the sensitivity ratio and hence confirming its best performance and robustness for RUL estimation.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

The software, formal analysis and synthetic, testing, investigation, evaluation system, resources management, data curation, writing—original draft preparation and editing visualization have been done by 1st author. The paper conceptualization, methodology, writing—review, editing, validation content, supervision and project administration have been done by 2nd author.

References

[1] T. P. Carvalho, F. A. A. M. N. Soares, R. Vita, R. P. Francisco, J. P. Basto, and S. G. S. Alcala, "A systematic literature review of machine learning methods applied to predictive maintenance", *Computers & Industrial Engineering*, Vol. 137, p. 106024, 2019.

[2] F. Ahmadzadeh and J. Lundberg, "Remaining useful life estimation: review", *International Journal of System Assurance Engineering and Management*, Vol. 5, No. 4, pp. 461-474, 2013.

[3] J. Carroll, S. Koukoura, A. McDonald, A. Charalambous, S. Weiss, and S. McArthur, "Wind turbine gearbox failure and remaining useful life prediction using machine learning techniques", *Wind Energy*, Vol. 22, No. 3, pp. 360-375, 2019.

[4] S. Seluck, "Predictive maintenance, its implementation and latest trends", In: *Proc. of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, Vol. 9, pp. 1670-1679, 2017.

[5] Y. Wanga, Y. Zhaoa, and S. Addepalli, "Remaining Useful Life Prediction using Deep Learning Approaches: A Review", In: *Proc. of 8th International Conf. on Through-Life Engineering Service – TESConf. 2019*, pp. 81-88, 2019.

[6] S. Hochreiter and J. Schmidhuber, "Long short-term memory", *Neural Computation*, Vol. 9, No. 8, pp. 1735-1780, 1997.

[7] J. T. Connor, R. D. Martin, and L. E. Atlas, "Recurrent Neural Networks and Robust Time Series Prediction", *IEEE Transactions on Neural Networks*, Vol. 5, No. 2, pp. 240-254, 1994.

[8] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSTM cells and network structures", *Neural Computation*, Vol. 31, No. 7, pp. 1235-1270, 2019.

[9] Y. Cheng, J. Wu, H. Zhu, S. W. Or, and X. Shao, "Remaining Useful Life Prognosis Based on Ensemble Long Short-Term Memory Neural Network", *IEEE Transactions on Instrumentation and Measurement*, Vol. 70, pp. 1-12, 2020.

[10] B. Chinomona, C. Chung, L. K. Chang, W. C. Su, and M. C. Tsai, "Long Short-Term Memory Approach to Estimate Battery Remaining Useful Life Using Partial Data", *IEEE Access*, Vol. 8, pp. 165419-165431, 2020.

[11] J. Zhang, P. Wang, R. Yan, and R. X. Gao, "Long short-term memory for machine remaining life prediction", *Journal of Manufacturing Systems*, Vol. 48, pp. 78-86, 2018.

[12] Y. Wu, M. Yuan, S. Dong, L. Lin, and Y. Liu, "Remaining Useful Life Estimation of Engineered Systems using vanilla LSTM Neural Networks", *Neurocomputing*, Vol. 275, pp. 167-179, 2018.

[13] P. Intachai and P. Yuvapoositanon, "A Prototype Similarity-based System for Remaining Useful Life Estimation for Future Industry by Singular Spectrum Analysis-Long Short Term Memory Neural Networks Algorithm", *Journal of Mobile Multimedia*, Vol. 16, No. 1-2, pp. 181-202, 2020.

[14] P. Yuvapoositanon and P. Intachai, "A Singular Spectrum Analysis-based Synthetic Dataset Generation Method for Remaining Useful Life Estimation of Turbo Fan Engines", *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 4, pp. 359-372, 2021, doi: 10.22266/ijies2021.0831.32.

[15] Y. Zhang, R. Xiong, and H. He, "Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries", *IEEE Transactions on Vehicular Technology*, Vol. 67, No. 7, pp. 5695-5705, 2018.

[16] S. Xiang, Y. Qin, C. Zhu, Y. Wang, and H. Chen, "Long short-term memory neural network with weight amplification and its application into gear remaining useful life prediction", *Engineering Applications of Artificial Intelligence*, Vol. 91, p. 103587, 2020.

[17] A. Z. Hinchi and M. Tkiouat, "Rolling element bearing remaining useful life estimation based on a convolutional long-short-term memory network", *Procedia Computer Science*, Vol. 127, pp. 123-132, 2018.

[18] D. Meyer, F. Leisch, and K. Hornik, "The support vector machine under test", *Neurocomputing*, Vol. 55, No. 1-2, pp. 169-186, 2003.

[19] J. Liu, Q. Li, W. Chen, Y. Yan, Y. Qiu, and T. Cao, "Remaining useful life prediction of pemfc based on long short-term memory recurrent neural networks", *International Journal of Hydrogen Energy*, Vol. 44, No. 11, pp. 5470-5480, 2019.

[20] R. H. Nielsen, "Theory of the backpropagation neural network", *Neural Networks for Perception*, pp. 65-93, 1992.

[21] N. Golyandina and A. Zhigljavsky, "Basic SSA", *Singular Spectrum Analysis for Time Series, Springer Science and Business Media*, Ch. 2, pp. 11-90, 2013.

[22] N. Golyandina and A. Korobeynikov, "Basic singular spectrum analysis and forecasting with r", *Computational Statistics & Data Analysis*, Vol. 71, pp. 934-954, 2014.

[23] H. Hassani and R. Mahmoudvand, "Applications of Singular Spectrum Analysis", *Singular Spectrum Analysis: Using R, Springer Business and Economics*, Ch. 3, pp. 87-102, 2018.

[24] Y. Zhang, P. Hutchinson, N. A. J. Lieven, and J. N. Yanez, "Remaining Useful Life Estimation Using Long Short-term Memory Neural Networks and Deep Fusion", *IEEE Access*, Vol. 8, pp. 19033-19045, 2020.

[25] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks", In: *Proc. of 2013 IEEE International Conf. on Acoustics, Speech and Signal Processing*, pp. 6645-6649, 2013.

[26] NASA, "Prognostics Center of Excellence Data Repository", *Turbofan Engine Degradation Simulation Data Set*, Accessed on: Apr. 7, 2022. [Online]. Available: https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/.

[27] V. Klema and A. Laub, "The singular value decomposition: Its computation and some applications", *IEEE Transactions on Automatic Control*, Vol. 25, No. 2, pp. 164-176, 1980.

[28] J. B. Elsner and A. A. Tsonis, "Foundations of SSA", *Singular Spectrum Analysis: a New Tool in Time Series Analysis, Springer Science and Business Media*, Ch. 4, pp. 39-50, 2013.

[29] M. A. R. Khan and D. Poskitt, "Window length selection and signal-noise separation and reconstruction in singular spectrum analysis", *Monash Econometrics and Business Statistics Working Papers*, Vol. 23, No. 11, pp. 2011-23, 2011.

[30] N. K. Manaswi, "RNN and LSTM", *Deep Learning with Applications Using Python*, Ch. 9, pp. 115-126, 2018.