Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 55-10 (2022) 1207–1212

A Novel Predictive Selective Maintenance Strategy Using Deep Learning and Mathematical Programming

Ryan O'Neil * **Claver Diallo** ** **Abdelhakim Khatab** ***

* *Department of Industrial Engineering, Dalhousie University, Halifax, Canada (e-mail: ry855314@dal.ca).*
** *Department of Industrial Engineering, Dalhousie University, Halifax, Canada (e-mail: Claver.Diallo@Dal.ca).*
*** *Laboratory of Computer Engineering, Production and Maintenance, Lorraine University, Metz, France (e-mail: abdelhakim.khatab@univ-lorraine.fr).*

Abstract: Many systems are required to perform a series of missions with finite breaks between successive missions. For such systems, one of the most widely used maintenance strategies is selective maintenance (SM). Under certain maintenance constraints, the SM problem (SMP) consist in selecting an optimal subset of feasible maintenance actions to maximize the system reliability for the upcoming mission. Almost all SMP models proposed in the literature are focused on traditional physics-based reliability models, where component lifetimes can be modeled using a stochastic process. With the application of new technologies such as wireless sensors and Industrial Internet of Things (IIoT), and the recent advancements in Deep Learning (DL) algorithms for prognostics, predictive maintenance based on data-driven methods has become a very popular maintenance strategy. These data driven methods have shown extreme accuracy in predicting remaining useful life (RUL) of components and systems. The goal of this paper is to introduce a predictive selective maintenance strategy that can be used to solve complex and relatively large multi-component systems. A DL algorithm will be used to estimate the probability that each component will successfully complete the upcoming mission, a selective maintenance optimization model will then be used to identify the maintenance actions that will maximize the system reliability. An efficient solution method is devised to solve the resulting complex optimization problem. The NASA C-MAPSS (Commercial Modular Aero-Propulsion System Simulation) dataset is used to train and evaluate the DL algorithm. The numerical experiments carried out show that the proposed novel predictive maintenance strategy is accurate and yields valid decisions.

Keywords: Artificial intelligence, Mission-oriented systems, Deep learning, Predictive maintenance, Reliability optimization.

1. INTRODUCTION

Modern life depends upon complex and highly interconnected production and distribution networks for energy, goods, and services. These networks are designed to operate with little interruption, extraordinarily high reliability, and high readiness to better respond to disruptions caused by unforeseen events such as weather disruptions, natural disasters, and pandemics. Monitoring the health and degradation level of these systems and predicting when failures are to occur and proactively maintaining them will increase the performance of the whole integrated network. Many of the production-distribution assets are required to run consecutive missions interspersed with scheduled breaks for maintenance. These are called mission-oriented systems and include systems such as production lines, aircraft, and energy production and distribution assets. Selective maintenance (SM) is an innovative maintenance strategy applied to these mission-oriented systems aiming to find an optimal list of maintenance actions to be performed on a specific subset of components with the objective of either maximizing the system reliability for the upcoming mission or minimizing maintenance costs (Diallo et al., 2018).

The original selective maintenance model introduced by Rice et al. (1998) dealt with a series-parallel system with constant failure rate components and perfect repair of failed components. A full enumeration method was used to find the optimal solution. In the intervening years since Rice et al. (1998) proposed the first SMP model, many researchers have expanded upon their work. These studies have included complex system configurations (Cassady et al., 2001; Diallo et al., 2018), multistate systems (Liu and Huang, 2010; Pandey et al., 2013), component

⋆ The authors acknowledge the financial support received from the Discovery Grant Program of the Canadian Natural Science and Engineering Research Council (NSERC).

10.1016/j.ifacol.2022.09.554

dependence (Xu et al., 2016; Dao and Zuo, 2017), fleet level selective maintenance (Khatab et al., 2020; Schneider and Cassady, 2015), multimission (Chaabane et al., 2020), stochastic break and/or mission duration (Liu et al., 2018; Khatab et al., 2017), condition-based SMP (Khatab et al., 2018a), and multiple repair channels (Diallo et al., 2019; Khatab et al., 2018b). A literature review of the SMP is provided in (Xu et al., 2015). A more recent SMP literature review was conducted by Cao et al. (2018).

The current trend of automation, referred to as industry 4.0, utilizes embedded systems and sensors, machine-to-machine communication, Industrial Internet of Things (IIoT) and Cyber-Physical Systems (CPS) technologies to connect machines, operations, equipment, and people. With the introduction of new technologies, such as IIoT, predictive maintenance (PdM) is becoming a more attractive maintenance strategy in many industries. The idea of PdM is to plan maintenance based on the current state and predicted future state of a system. In recent times, deep learning (DL) algorithms or neural networks have made tremendous strides in remaining useful life (RUL) prediction and have improved the state of the art in PdM (Namuduri et al., 2020). Li et al. (2018) develop a deep convolutional neural network for the problem of predicting remaining useful life of an aircraft engine. Hsu and Jiang (2018) use a long short-term memory (LSTM) network to estimate the remaining useful life of aero-propulsion engines. The accuracy of their proposed model is compared to the multi-layer perceptron, support vector regression, and convolutional neural network. They show that their LSTM network performs better than the other methods in terms of root mean squared error. Huang et al. (2007) use a neural network for remaining useful life prediction and they demonstrate that their data-driven approach outperforms distribution-based reliability models.

To the best of our knowledge, the first and only predictive selective maintenance framework was introduced by Hesabi et al. (2021) where a SM optimization model interacts with an LSTM network used for reliability estimation. Only an elementary series system is considered and a full enumeration approach is used to solve the complex optimization problem. For systems of even moderate size the full enumeration approach would fail to provide solutions in a reasonable amount of time and thus more efficient solution methods must be utilized. It is also common to encounter complex structures such as k-out-of-n:G in a wide range of industrial applications and thus these structures must be explored. The objective of this paper is two-fold: (i) to extend the work of Hesabi et al. (2021) to include more complex system structures, and (ii) to solve the developed predictive SM optimization problem using a more efficient solution approach.

The remainder of this paper is structured around six additional sections. In Section 2 the notation and main working assumptions are listed. In Section 3, the system under consideration is described, as well as the computation of its reliability during the next mission. The modeling of the maintenance actions is also described in Section 3. In Section 4, the LSTM network is described and evaluated using the NASA C-MAPSS FD001 dataset. In Section 5, the mathematical formulation of the SMP as well as the solution approach are presented. Multiple numerical experiments are conducted in Section 6. Conclusions and future extensions are drawn in section 7.

2. NOTATION AND MAIN WORKING ASSUMPTIONS

2.1 Notation

$\mathcal{I}$	Set of subsystems, $\mathcal{I} = \{1, 2, ..., N\}$ with index i
$\mathcal{J}_i$	Set of components in subsystem i, $\mathcal{J}_i = \{1, 2, ..., N_i\}$ with index j
E_{ij}	The j^{th} component of subsystem i
K_i	Minimum number of components that must be functioning in subsystem i
$\mathcal{L}_{ij}$	Set of preventive maintenance levels available for component E_{ij}, $\mathcal{L}_{ij} = \{0, 1, ..., L_{ij}\}$ with index l
$\mathcal{P}_i$	Set of maintenance patterns generated for subsystem i, $\mathcal{P}_i = \{1, 2, ..., P_i\}$ with index p
t_{ijl}	Duration of PM level l on component E_{ij}
c_{ijl}	Cost of PM level l on component E_{ij}
U	Mission duration
D	Maintenance break duration
C	Maintenance budget
$\mathcal{R}_{ij}$	Probability of component E_{ij} to operate the next mission
$\mathcal{R}_i^s$	Reliability of subsystem i
$\mathcal{R}$	Overall system reliability

2.2 Main working assumptions

(1) The system is comprised of multiple k-out-of-n:G subsystems. Each subsystem is made up of multiple multi-state components.
(2) The system and components only degrade with usage. During the maintenance break the system is assumed to be switched off and therefore not experiencing any degradation.
(3) Maintenance actions are allowed only during the break duration.
(4) When a maintenance action is performed on a component, the component is brought back to a state that it was in previously, where the RUL is higher.

3. SYSTEM DESCRIPTION, RELIABILITY COMPUTATION, AND MAINTENANCE MODELING

3.1 System Description

The SMP addressed in this paper considers a system comprised of N GA(K_i, N_i) subsystems arranged in a series configuration, where the i^{th} subsystem ($i = 1, \cdots, N$) is comprised of N_i components. In reliability theory, the k-out-of-n:G configuration is usually denoted as GA(k, n) and specifies that the system is functioning if at least k among the n components are functioning; it is a generalization of both the series and parallel structures. Individual components in each subsystem are independent and the state of each component degrades and deteriorates with both operational time and usage. Each component is modeled as multi-state as there are multiple stages in the degradation process.

The system under consideration is required to perform alternating series of missions and scheduled maintenance breaks of finite length. It is assumed that the system has just completed a mission and is entering the first break. During the break the system is switched off for a duration D during which maintenance actions can be performed. The objective will be to identify the the optimal set of maintenance actions to be carried out in order to maximize the system reliability under maintenance resource constraints. There is a cost and time associated with every maintenance action that can be selected for a given component. The total cost and time to perform the selected maintenance actions must not exceed the maintenance budget and break duration respectively.

Each component E_{ij} is continuously monitored by s sensors. After each operating cycle, new sensor measurements are recorded. The corresponding collected sensor data $\mathbf{X}_{ij}$ can be represented as a matrix:

$$\mathbf{X}_{ij} = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_s^1 \\ x_1^2 & x_2^2 & \dots & x_s^2 \\ \vdots & \vdots & & \vdots \\ x_1^{T_{ij}} & x_2^{T_{ij}} & \dots & x_s^{T_{ij}} \end{bmatrix},$$

where x_v^t represents the value of sensor v ($v = 1, \dots, s$) during cycle t ($t = 1, \dots, T_{ij}$), and T_{ij} refers to the current age (number of cycles completed) of component E_{ij}. At the start of the maintenance break, the sensor data and number of cycles completed for all components will be available. By first training an LSTM network on historic sensor data for the purpose of classifying whether the remaining RUL is greater than the mission length, the probability of a given component completing its next mission can be approximated. Then, using the two phased approach proposed by Diallo et al. (2018) the maintenance actions that will result in maximizing the system reliability for the upcoming mission can be identified and selected.

3.2 Reliability Computation

The probability that the system will successfully complete its next mission is given by its reliability $\mathcal{R}$. To compute $\mathcal{R}$, an LSTM classifier is first used to predict the probability $\mathcal{R}_{ij}$ of each component successfully completing the next mission. The LSTM network will be trained on historical sensor data to predict the class of a given component, where a component will be of class 0 if its RUL is greater than the specified mission length, and 1 otherwise. After estimating the component reliabilities, the subsystem reliability for a k-out-of-n:G system can be computed using the efficient algorithm by Kuo and Zuo (2003). Based on the assumption that the subsystems are arranged in a series configuration, the overall system reliability $\mathcal{R}$ is computed as:

$$\mathcal{R} = \prod_{i \in I} \mathcal{R}_i^s. \tag{1}$$

3.3 Maintenance Modeling

For each component E_{ij}, there is a list $\mathcal{L}_{ij} = \{0, \cdots, L_{ij}\}$ of $L_{ij} + 1$ maintenance actions $l \in \mathcal{L}_{ij}$ that can be selected during the break. These maintenance levels include do-nothing, imperfect maintenance (IM), and replacement. The do-nothing ($l = 0$) case refers to no maintenance being performed on the component. The replacement level ($l = L_{ij}$) will return the component to an "as good as new" state, while an IM level $0 < l < L_{ij}$ if selected will return the component to a previous state where the RUL is higher. When carried out on a component E_{ij}, a maintenance action l requires t_{ijl} time units, and costs c_{ijl} monetary units.

4. LSTM NETWORK

This section discusses the LSTM architecture used for classifying components. The dataset used to train and test the LSTM classifier is the NASA C-MAPSS (Commercial Modular Aero-Propulsion System Simulation) dataset. This dataset contains 4 different subsets each with run to failure information under different operational conditions and fault modes. The present work uses the first set (called FD001) which considers one fault mode. Both a training and test set are provided. The LSTM network is a specific type of recurrent neural network (RNN) that is suitable for making predictions from sequential or time series data. A common issue that arises when using standard RNNs is the tendency for the gradient to explode or vanish during back propagation resulting in poor performance. LSTMs have indeed been specifically designed to address the vanishing and exploding gradient problem. The LSTM network is a chain like structure comprised of multiple cells each comprised of 4 neural networks. The inner workings of each cell are shown in figure 1.

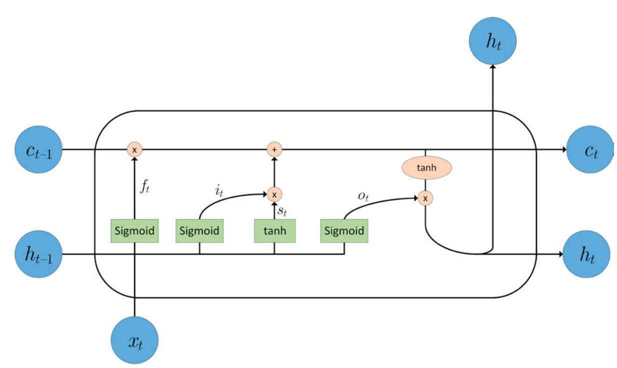

Fig. 1. The LSTM cell structure

Just like RNNs, LSTMs have a hidden state that carries information from immediately previous events and is denoted as h_t. What makes the LSTM network unique is the addition of what is called the cell state denoted as c_t. The cell state is able to store information of events that occurred many time steps in the past (long term memory).

The first sigmoid layer in each unit, referred to as the forget gate, is used to identify what information from the previous cell state c_{t-1} should be removed. This is done by first concatenating the previous hidden state h_{t-1} with the new input x_t and sending it though a sigmoid layer. The output values f_t of the sigmoid layer is a vector of the same dimension as c_{t-1} with values ranging from 0 to 1. The vector f_t is given by:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right). \tag{2}$$

An element wise multiplication between f_t and c_{t-1} is then performed. This operation is what allows some information to be "forgotten". The next sigmoid and "tanh" layers contribute together to determine the new information that will be added to the cell state, these are often referred to as the input gate and new candidate gate respectively. The sigmoid layer produces a vector i_t that determines what values in the cell state will be updated. The tanh layer produces a vector s_t that will determine possible candidate solutions. These two vectors can be expressed as:

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) \tag{3}$$
$$s_t = tanh\left(W_s \cdot [h_{t-1}, x_t] + b_s\right). \tag{4}$$

An element wise multiplication is performed between vectors i_t and s_t and the result is added to the previous cell state to produce the new cell state c_t. The final operation is to decide what the value of the new hidden state h_t will be. The new hidden state is computed by first sending h_{t-1} and x_t through a sigmoid layer and obtaining o_t as shown in Equation (5).

$$o_t = \sigma\left(W_o \cdot [h_{t-1}, x_t] + b_o\right). \tag{5}$$

A hyperbolic tangent operation is then performed on all elements of c_t and the result is multiplied by o_t to obtain the new hidden state h_t. The LSTM network used in this paper was a stacked network comprised of two layers. By stacking LSTM layers, additional model complexity is achieved allowing for more complex feature representations of the input to be learned. A random search was applied to determine the best model hyper-parameters. The optimal parameters found through the random search are provided in Table 1.

Table 1. Optimal hyper-parameters for the LSTM network

LSTM units layer 1	LSTM units layer 2	Dropout	Epochs	Batch size
150	100	0.2	15	32

4.1 LSTM Evaluation

Using the best parameters found through the random search, the LSTM network was trained using the training set and evaluated using the testing set of the FD001 dataset. The classification report displaying the models performance on the test set is shown in Table 2. The mission length was set to $U = 40$ cycles. The results in Table 2, show that the LSTM classifier is indeed able to achieve a high f1-score for both classes with extremely high accuracy. As we are dealing with an unbalanced dataset, the f1-score is preferred over accuracy for evaluating the performance of the model. Based on the models high performance on the test set, we can conclude that it can provide accurate estimates for the probability of a component to successfully complete the upcoming mission.

Table 2. Classification report for the LSTM model

	Precision	Recall	f1-score	Support
Class 0	0.99	0.99	0.99	12143
Class 1	0.90	0.91	0.90	853
Accuracy			0.99	12996
Macro Avg.	0.95	0.95	0.95	12996
Weighted Avg.	0.99	0.99	0.99	12996

5. SM OPTIMIZATION MODEL AND SOLUTION METHOD

In this section, a formulation for the selective maintenance problem is presented. This formulation relies on a full enumeration of all maintenance patterns $\mathcal{P}_i$ for each subsystem $i \in \mathcal{I}$ and was first proposed by Diallo et al. (2018). A maintenance pattern $p \in \mathcal{P}_i$ is defined as a combination of components and related maintenance levels to be performed during each break. A pattern $p \in \mathcal{P}_i$ is represented as a column-vector of N_i elements whose values are the maintenance levels performed on the components. To illustrate the generation of maintenance patterns, consider a simple parallel subsystem comprised of two components with two levels of maintenance: Do nothing ($l = 0$) and replacement ($l = 1$). All 4 possible maintenance patterns that would be generated are as follows:

$$\begin{bmatrix}0\\0\end{bmatrix} \quad \begin{bmatrix}0\\1\end{bmatrix} \quad \begin{bmatrix}1\\1\end{bmatrix} \quad \begin{bmatrix}1\\0\end{bmatrix}$$

The first pattern (column-vector) means that no maintenance is performed on both components. The second pattern means that only component E_{12} is replaced during the break. According to the third pattern, components E_{11} and E_{12} are replaced during the break, and the fourth pattern would mean that only component E_{11} is replaced.

For each pattern $p \in \mathcal{P}_i$ corresponds a total expected cost $\mathcal{C}_{ip}$ and time $\mathcal{T}_{ip}$ to perform the selected maintenance actions during the break, as well as a reliability $\mathcal{R}^s_{ip}$ of subsystem i. This reliability is computed by first using our LSTM network to predict the reliability of each component, and then using the algorithm proposed by Kuo and Zuo (2003) to compute the subsystem reliability. This is repeated for every feasible pattern.

Now, assuming that for each subsystem $i \in \mathcal{I}$ complete pattern information $\mathcal{P}_i$ is available, the BIP formulation of the SMP as proposed by Diallo et al. (2018) with the objective of maximizing system reliability is written as:

$$\max_{z_{ip}} \quad \sum_{i\in\mathcal{I}} \sum_{p\in\mathcal{P}_i} \ln(R^s_{ip}) \cdot z_{ip} \tag{6}$$

Subject to:

$$\sum_{p\in\mathcal{P}_i} z_{ip} = 1, \quad \forall i \in \mathcal{I} \tag{7}$$

$$\sum_{i\in\mathcal{I}} \sum_{p\in\mathcal{P}_i} \mathcal{T}_{ip} \cdot z_{ip} \le D \tag{8}$$

$$\sum_{i\in\mathcal{I}} \sum_{p\in\mathcal{P}_i} \mathcal{C}_{ip} \cdot z_{ip} \le C \tag{9}$$

$$z_{ip} \in \{0,1\}, \quad \forall i \in \mathcal{I}, p \in \mathcal{P}_i \tag{10}$$

In the above optimization model, the objective function is the linearization of the nonlinear system reliability function in equation (1). Constraints (7) ensure that a single maintenance pattern is selected for each subsystem i. Constraints (8) ensure that the maintenance time does not exceed the available working time. Constraints (9) guarantee that the total maintenance cost does not exceed the maintenance budget.

6. ILLUSTRATIVE EXAMPLE

In this section multiple numerical experiments are conducted on a system comprised of three subsystems and containing a total of 12 components (Figure 2). Each component of the system is assigned the sensor values of an engine from the test set of the NASA CMAPSS FD001 dataset. It is assumed that the previous mission has just concluded and the state of each component is defined by the current sensor values. The engine that each component takes it's sensor values from as well as the number of cycles that the engine has completed and its predicted reliability for the upcoming mission is reported in Table 3.

For all components a common list of $L = 5$ maintenance options is available. For each maintenance level, Table 4 gives the corresponding RUL increasing amount, in addition to the cost and the required time to be performed.

All experiments are run on a Intel$^{\text{TM}}$ i5 2.9GHz desktop computer with 12GB of RAM running Windows 10$^{\text{TM}}$. All algorithms were coded in Python 3.8. The optimization runs were carried out by Gurobi 9.1 using gurobipy.

Table 5 shows the results when the maintenance break duration is fixed at $D = 16$ hours while the maintenance budget C varied from \$400 to \$700. As the maintenance budget is increased, more expensive maintenance actions can be selected leading then to a higher system reliability. Table 6 shows the results for a varying maintenance break duration and a fixed maintenance budget ($C = \$600$). As the break duration is increased more maintenance actions can be performed and thus a higher system reliability is achieved. The number of feasible maintenance patterns increases as the break duration increases leading to a higher computational time. A similar result is seen in the first experiment when the maintenance budget is increased.

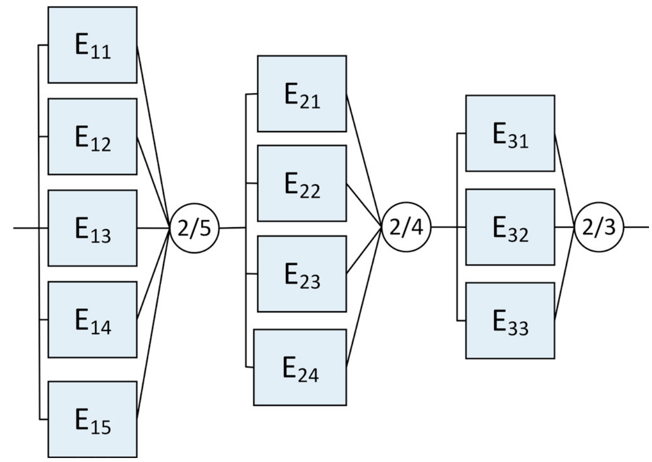

Fig. 2. System structure used in all experiments

Table 3. Engine data used for the experiments

Components	Engine	Num. cycles completed	$\mathcal{R}_{ij}$
E_{11}	17	165	0.893
E_{12}	18	133	0.003
E_{13}	20	184	0.000
E_{14}	18	133	0.003
E_{15}	20	184	0.000
E_{21}	84	172	0.860
E_{22}	24	186	0.000
E_{23}	31	196	0.000
E_{24}	34	203	0.000
E_{31}	32	145	0.770
E_{32}	36	126	0.000
E_{33}	37	121	0.000

Table 4. Maintenance data used for the experiments

Maintenance level	RUL Increase	c_{ijl}	t_{ijl}
$l = 0$	Do-nothing	0.0	0.0
$l = 1$	8	30.0	0.8
$l = 2$	16	75.0	2.0
$l = 3$	32	150.0	3.5
$l = 4$	Replacement	250.0	5.0

Table 5. System reliability for varying values of C and $D = 16$ hours

C (\$)	$\mathcal{R}^*$	D^* (hours)	C^* (\$)	CPUt (s)
400	0.55	9.0	375	357.8
500	0.87	12.2	495	526.6
600	0.94	13.7	595	714.7
700	0.99	15.3	655	847.8

Table 6. System reliability for varying values of D and $C = \$600$

D (hours)	$\mathcal{R}^*$	D^* (hours)	C^* (\$)	CPUt (s)
8	0.20	7.5	300	255.3
10	0.71	9.8	405	416.0
12	0.87	11.4	465	611.3
14	0.94	13.7	595	705.2

7. CONCLUSION

This paper improved upon the predictive selective maintenance framework proposed in Hesabi et al. (2021) by considering more complex subsystem structures and by utilizing an improved solution method. The stacked LSTM network used for reliability prediction was shown to achieve high accuracy and f1-score on the testing set implying its capability to accurately predict the components reliability for the upcoming mission. The results obtained from numerical experiments demonstrated that the proposed framework can solve the selective maintenance problem in a reasonable amount of time for complex systems.

Future extensions of the present work would be to explore different Deep Learning algorithms such as convolutional neural networks for component classification and reliability prediction. It would also be very interesting to develop a deep multimodal learning model in which different modalities or types of information are combined. For example, combining both thermal imaging and time series sensor data to make a prediction on RUL.

REFERENCES

Cao, W., Jia, X., Hu, Q., Zhao, J., and Wu, Y. (2018). A literature review on selective maintenance for multi-unit systems. *Quality and Reliability Engineering International*, 1–22.

Cassady, C.R., Pohl, E.A., and Murdock, W.P. (2001). Selective maintenance modeling for industrial systems. *Journal of Quality in Maintenance Engineering*.

Chaabane, K., Khatab, A., Diallo, C., Aghezzaf, E.H., and Venkatadri, U. (2020). Integrated imperfect multimission selective maintenance and repairpersons assignment problem. *Reliability Engineering & System Safety*, 199, 106895.

Dao, C.D. and Zuo, M.J. (2017). Optimal selective maintenance for multi-state systems in variable loading conditions. *Reliability engineering & system safety*, 166, 171–180.

Diallo, C., Venkatadri, U., Khatab, A., and Liu, Z. (2018). Optimal selective maintenance decisions for large serial k-out-of-n: G systems under imperfect maintenance. *Reliability Engineering & System Safety*, 175, 234–245.

Diallo, C., Venkatadri, U., Khatab, A., Liu, Z., and Aghezzaf, E.H. (2019). Optimal joint selective imperfect maintenance and multiple repairpersons assignment strategy for complex multicomponent systems. *International Journal of Production Research*, 57(13), 4098–4117.

Hesabi, H., Nourelfath, M., and Hajji, A. (2021). A deep learning predictive model for selective maintenance optimization. *Reliability Engineering & System Safety*, 108191.

Hsu, C.S. and Jiang, J.R. (2018). Remaining useful life estimation using long short-term memory deep learning. In *2018 ieee international conference on applied system invention (icasi)*, 58–61. IEEE.

Huang, R., Xi, L., Li, X., Liu, C.R., Qiu, H., and Lee, J. (2007). Residual life predictions for ball bearings based on self-organizing map and back propagation neural network methods. *Mechanical systems and signal processing*, 21(1), 193–207.

Khatab, A., Diallo, C., Aghezzaf, E.H., and Venkatadri, U. (2020). Optimization of the integrated fleet-level imperfect selective maintenance and repairpersons assignment problem. *Journal of Intelligent Manufacturing*, 1–16.

Khatab, A., Aghezzaf, E.H., Diallo, C., and Djelloul, I. (2017). Selective maintenance optimisation for series-parallel systems alternating missions and scheduled breaks with stochastic durations. *International Journal of Production Research*, 55(10), 3008–3024.

Khatab, A., Diallo, C., Aghezzaf, E.H., and Venkatadri, U. (2018a). Condition-based selective maintenance for stochastically degrading multi-component systems under periodic inspection and imperfect maintenance. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 232(4), 447–463.

Khatab, A., Diallo, C., Venkatadri, U., Liu, Z., and Aghezzaf, E.H. (2018b). Optimization of the joint selective maintenance and repairperson assignment problem under imperfect maintenance. *Computers & Industrial Engineering*, 125, 413–422.

Kuo, W. and Zuo, M.J. (2003). *Optimal reliability modeling: principles and applications*. John Wiley & Sons.

Li, X., Ding, Q., and Sun, J.Q. (2018). Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliability Engineering & System Safety*, 172, 1–11.

Liu, Y., Chen, Y., and Jiang, T. (2018). On sequence planning for selective maintenance of multi-state systems under stochastic maintenance durations. *European Journal of Operational Research*, 268(1), 113–127.

Liu, Y. and Huang, H.Z. (2010). Optimal selective maintenance strategy for multi-state systems under imperfect maintenance. *IEEE Transactions on Reliability*, 59(2), 356–367.

Namuduri, S., Narayanan, B.N., Davuluru, V.S.P., Burton, L., and Bhansali, S. (2020). Deep learning methods for sensor based predictive maintenance and future perspectives for electrochemical sensors. *Journal of The Electrochemical Society*, 167(3), 037552.

Pandey, M., Zuo, M.J., and Moghaddass, R. (2013). Selective maintenance modeling for a multistate system with multistate components under imperfect maintenance. *Iie Transactions*, 45(11), 1221–1234.

Rice, W., Cassady, C., and Nachlas, J. (1998). Optimal maintenance plans under limited maintenance time. In *Proceedings of the seventh industrial engineering research conference*, 1–3.

Schneider, K. and Cassady, C.R. (2015). Evaluation and comparison of alternative fleet-level selective maintenance models. *Reliability Engineering & System Safety*, 134, 178–187.

Xu, Q.z., Guo, L.m., Shi, H.p., and Wang, N. (2016). Selective maintenance problem for series–parallel system under economic dependence. *Defence technology*, 12(5), 388–400.

Xu, Q.Z., Guo, L.M., Wang, N., and Fei, R. (2015). Recent advances in selective maintenance from 1998 to 2014. *Journal of Donghua University (English Edition)*, 32(6), 986–994.