Employing Deep Learning for Predicting Remanufacturing Costs of EOL Products.

Jyoti Khanna[1], Prof. Devkinandan Nagar[2]

Abstract*— Remanufacturing cost prediction is conducive to visually judging the remanufacturability of end-of-life (EOL) products from economic perspective. However, due to the randomness, non-linearity of remanufacturing cost and the lack of sufficient data samples. The general method for predicting the remanufacturing cost of EOL products is very low precision. The approach is based on historical remanufacturing cost data to build a model for prediction. First of all, the remanufacturing cost of individual EOL product is arranged as a time series in reprocessing order. The long short term memory (LSTM) deep learning structure is used to predict the cost. The ADAM optimization is also applied to enhance the performance of the LSTM Model, where a combination of two gradient descent methodologies, momentum and a weighted average of the gradients is used to make the algorithm converge towards the minima in a faster pace. It has been shown that the proposed approach attains higher accuracy compared to existing work.

***Keywords*—Remanufacturing cost prediction, End of Life (EOL), Deep Learning, Long Short Term Memory (LSTM), Mean Absolute Percentage Error (MAPE).**

I. INTRODUCTION

Traditional manufacturing industry is under increasing pressures to be resource efficiency, thus to consume less resources and to reduce pollution on the environment [1]. Remanufacturing is a significant value-recovery approach industrial process that can return end-of-life (EOL) products to as-good-as-new ones, thus releases the huge hidden economic value of EOL products [2]-[3]. It is becoming a strategic emerging industry in many countries, e.g. China and USA. However, at present, there is an urgent need for an efficient and effective decision making system to decide whether an EOL product should be re- manufactured [4]. The assessment and decision-making of remanufacture ability for EOL products should be, concentrated on both technical and economic feasibility [5]. To this end, some evaluation models and reliability theories for remanufacturability have been presented. Cost prediction is basically a time series prediction problem. Mathematically [6]:

$$\boldsymbol{C} = \boldsymbol{f}(\boldsymbol{t}, \boldsymbol{v}) \quad \textbf{(1)}$$

Here,
Cost
f represents a function of
t is the time variable
v are other influencing global variables

The dependence of remanufacturing cost over time makes it somewhat predictable under similar other conditions of global influencing variables [7]-[8]. However, even the slightest of changes can derail the prediction completely [9]. Not only is the remanufacturing cost affected by uncertain market factors, but also as a special manufacturing process, re- manufacturing faces many uncertainties, e.g., the remanufacturing rate of recycled products, the arrival time of the EOL products, reprocessing route, the quantity of recycled products, and the purchase quantity of new parts, etc [10]. These factors of remanufacturing cost make it difficult to create a model to predict remanufacturing cost that possesses, good nonlinear approximation ability [11]. To this end, this paper takes the pre- diction of remanufacturing cost for EOL products as the research objective, focuses on the decomposition of less sample data and adopts a well-fitted algorithm, and the prediction accuracy of the model is further improved [12]-[13]. In the current researches, the remanufacturing cost prediction models can be divided into two categories: traditional linear prediction methods (such as vector auto-regression) and intelligent algorithm model with high fitting accuracy [14]. Statistical techniques are not found to be as accurate as the contemporary artificial intelligence and machine learning based approaches [15].

II. DEEP LEARNING

Deep learning has evolved as one of the most effective machine learning techniques which has the capability to handle extremely large and complex datasets [16]. It is training neural networks which have multiple hidden layers as compared to the single hidden layer neural network architectures [17]-[18].
The architectural view of a deep neural network is shown in figure 1. In this case, the outputs of each individual hidden layer is fed as the input to the subsequent hidden layer [19]. The weight adaptation however can follow the training rule decided for the neural architecture. There are various configurations of hidden layers which can be the feed forward, recurrent or back propagation etc [20].

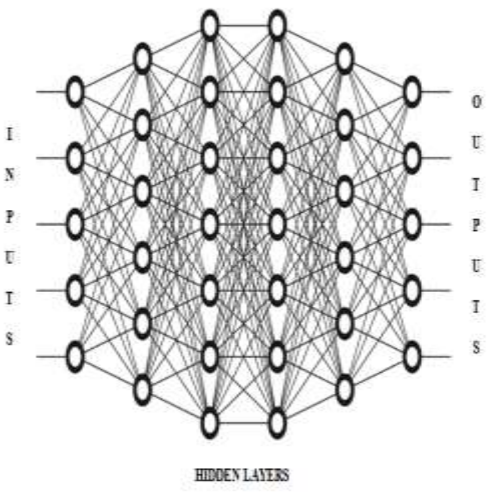

Fig.1 The Deep Neural Network Architecture

The figure above depicts the deep neural network architecture with multiple hidden layers. The output of the neural network however follows the following ANN rule:

$$Y = \sum_{i=1}^{n} \mathbf{X_i} \, . \, \mathbf{W_i} \; + \; \boldsymbol{\theta}_i \qquad \textbf{(2)}$$

Where,
X are the inputs
Y is the output
W are the weights
ϴ is the bias.
Training of ANN is of major importance before it can be used to predict the outcome of the data inputs.

III. THE LSTM MODEL

The Neural networks are practical AI models able to analyze large and complex data sets. The long short term memory (LSTM) is fundamentally a recurrent neural network (RNN) which is specially useful to time series analysis. RNNs have a feedback loop from output towards input [21].

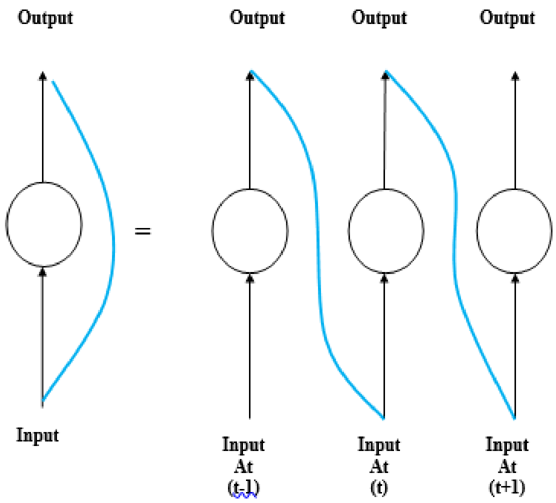

Fig.2 Recurrent Networks

Typically, the RNNs suffer from the vanishing gradient problem which is effectively sating that the gradient values near the input layer don't change as much as the gradient values close to the output layers. This issue is addressed in LSTMs. The LSTM primarily has 3 gates:

1) Input gate: This gate collects the presents inputs and also considers the past outputs as the inputs.

2) Output gate: This gate combines all cell states and produces the output.

3) Forget gate: This is an extremely important feature of the LSTM which received a cell state value governing the amount of data to be remembered and forgotten.

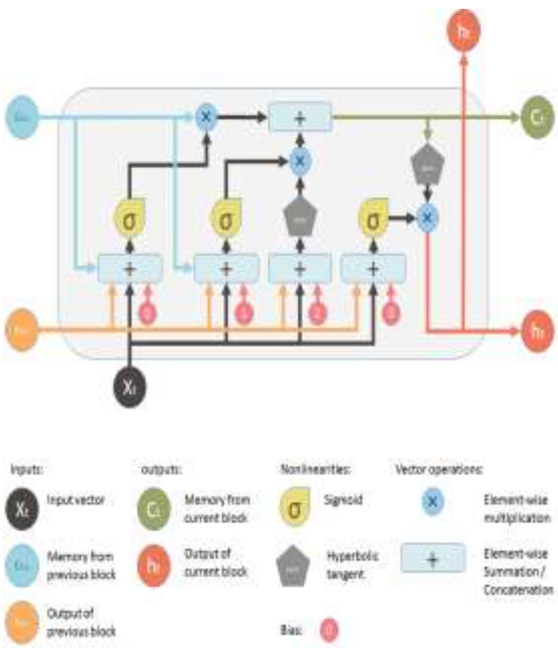

Fig.3 Structure of LSTM

The important aspect about LSTM is the fact that it repeatedly feeds the errors in every iteration to the network till the errors become constant or the maximum number of allowable iterations are over. This can be mathematically given by:

$if\ PF \neq constant$
$for\ (k = 1, k \leq k_{max} = constant, k = k + 1)$
{
$W_{k+1} = f(X_k, W_k, e_k)$
}
else
{
$W_{k+1} = W_k\ \&\&\ training\ stops$
}

Here,
X_k is the input to the kth iteration
W_k is the weight to the kth iteration
W_{k+1} is the weight to the (k+1)st iteration
e_k is the error to the kth iteration
k is the iteration number
PF is the performance function deciding the end of training
k_{max} is the maximum number of iterations

Thus if the error is within tolerance, which is generally not feasible to find beforehand in time series data, the training is stopped if the performance function (which can be the training error) becomes constant for multiple iterations or the maximum number of iterations are over. Now there are various ways in which the error can be minimized. This would be inferred from the number of iterations which are required to stop training. Thus the number of iterations would be a function of the gradient with which the error falls. Auto regression tries to find out the relation among a variable and its delayed version. Mathematically:

$$C_{Auto} = \int_0^T x(k)x(t-k)dk \quad (3)$$

Here,
C_{Auto} is the auto correlation of the variable
x is the variable
t is the time delay
k is the variable of integration
The auto correlation tells us about the changing nature of a variable.

ADAM Optimization for LSTM

As data is fed to a neural network for pattern recognition, the weights keep updating. However, it has been found that in case of time series problems, the latest data sample have the maximum impact on the latest output. Hence it is logical to calculate a moving average of latest (previous) data and apply it to the neural network. This fastens the convergenc and is termed as ADAM optimization [22]. This is also called a moving average. Mathematically,

$$I_k = X_{1,k}, Mean(X)_{k,k-n}, Y_k \quad (4)$$

Here,
I_k is the kth input sample to the neural network
$X_{1,k}$ are the data samples from the first to the kth sample
$Mean(X)_{k,k-n}$is the mean of the data samples form k-n to k, i.e. it is a moving average depending on the value of k
Y_k is the target.

LSTM's performance can depend upon the choice of optimization algorithm for the cost function.

The conventional LSTM works on the gradient descent with the **least squares (LS) optimization** can be used to define the cost function given by :

$$f_{cost} = \underbrace{min}_{k} \quad \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (5)$$

Where,
f_{cost} is the least squares cost function.
y_i is the target.
$\hat{y}_i$ is the predicted values.
n is the number of samples.

However, the **LS** optimization is being replace with the **ADAM optimization** which is is an extension to stochastic gradient descent that has recently seen broader adoption for deep learning. The gradient descent learning rule is given by:

$$w_{k+1} = w_k - \alpha\frac{\partial e}{\partial w} \quad (6)$$

Here,
w_{k+1} is the weight of the next iteration
w_k is the weight of the present iteration
e is the error
α is the learning rate

$$\frac{\partial e}{\partial w} = \frac{\partial e}{\partial y}.\frac{\partial y}{\partial w} \quad (7)$$

The chain rule in can be used for computing the error gradient. The training is stopped based on the mean square error or mse given by:

$$mse = \frac{\sum_{i=1}^{n} e_i^2}{n} \tag{8}$$

The final computation of the performance metric is the mean absolute percentage error given by:

$$MAPE = \frac{100}{M} \sum_{i=1}^{N} \frac{E - E_i|}{i} \tag{9}$$

Here,
n is the number of errors
i is the iteration number
E is the actual value
E_i is the predicted value

IV. RESULTS

The experimental results have been simulated on MATLAB and the prediction model used is the LSTM.

Fig.4 LSTM Parameters

Figure 4 depicts the LSTM model parameters such as dropout, fully connected layers and activation function.

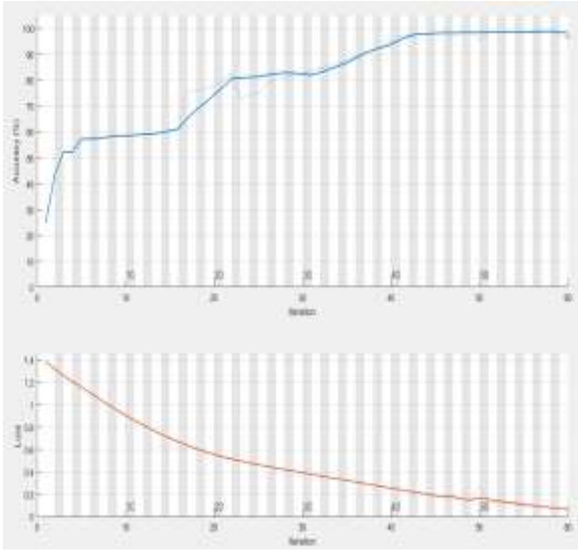

Fig.5 Training Progress for LSTM

Figure 5 depicts the training progress of the LSTM network, with the variation in the cost function as a function of iterations.

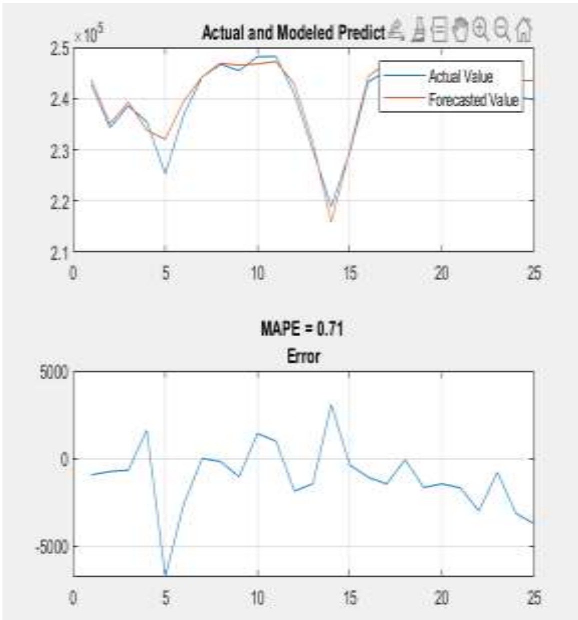

Fig.6 Actual and Forecasted Values

Figure depicts the predicted and actual stock behavior. It can be observed that the proposed work attains an MAPE value of 0.71%

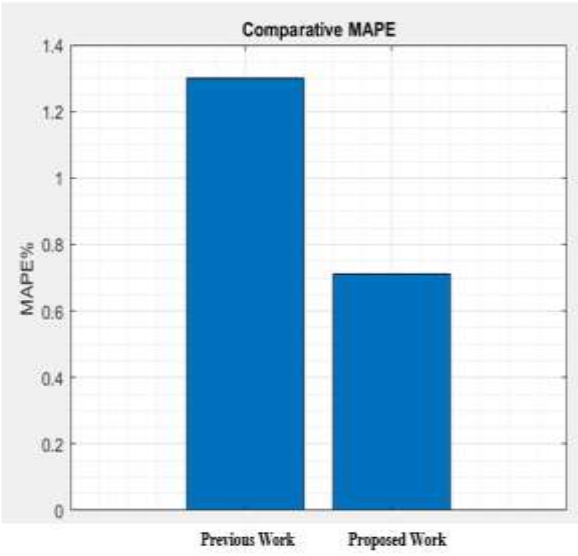

Fig.7 Comparison with previous work

Figure 7 depicts a comparative MAPE analysis with respect to previous work

A summary of the obtained results in the proposed work has been tabulated in table 1.

Table.1 Summary of Results

S.No.	Parameter	Value
1.	Data Source	Kaggle
2.	Splitting Ratio	70:30
3.	ML Model	LSTM
4.	Optimization	ADAM
5.	LSTM iterations	60
6.	Hidden Units	125
7,	Dropout	20%
8.	MAPE (proposed work)	0.71%
9.	MAPE (previous work)	1.30

It can be observed from the previous discussions that the proposed system employing LSTM with ADAM optimization outperforms the previous work [1] in terms of the MAPE of the system thereby making the proposed system more accurate.

CONCLUSION

It can be concluded from previous discussions that there are two main obstacles to remanufacturing cost prediction of EOL products. One is that the uncertainty of the remanufacturing process leads to the cost instability, the traditional linear prediction is difficult. The other is that the scale of the remanufacturing industry is limited, and the amount of available data accumulated is less, which results in the low accuracy of the prediction. For this purpose, a data- driven accelerated learning based LSTM network has been proposed for the cost prediction model. It can be observed from the obtained results that the proposed system attains better performance compared to previously existing system, due to the ADAM optimized LSTM model.

REFERENCES

1. Z Jiang, Z Ding, Y Liu, Y Wang, X Hu, Y Yang, A data-driven based decomposition–integration method for remanufacturing cost prediction of end-of-life products, Journal of robotics and computed aided manufacturing, Elsevier, 61 (2022), 101838.
2. Y. Zhang, S. Ren, Y. Liu, et al., A big data analytics architecture for cleaner man- ufacturing and maintenance processes of complex products, J Clean Prod 142 (2021) 626–641.
3. L. Li, C. Li, Y. Tang, Y. Du, An integrated approach of reverse engineering aided remanufacturing process for worn components, Robot Comput Integr Manuf 48 (2020) 39–50.
4. B. Ye, The research of application big data analytics to the parts remanufacturing economic evaluation, Huazhong University of Science and Technology, (2018), pp.1-8..
5. W.L. Ijomah, C.A. Mcmahon, G.P. Hammond, S.T. Newman, Development of design for remanufacturing guidelines to support sustainable manufacturing, Robot Comput Integr Manuf 23 (6) (2017) 712–71916.
6. Z. Jiang, Y. Jiang, Y. Wang, H. Zhang, H. Cao, G. Tian, A hybrid approach of rough set and case-based reasoning to remanufacturing process planning, J Intell Manuf 30 (1) (2019) 19–32.
7. X. Zhang, Remanufacturing scheme decision model and application based on re- maining useful life estimation, J Mech Eng 49 (7) (2013) 51.
8. Y. Du, H. Cao, X. Chen, B. Wang, Reuse-oriented redesign method of used products based on axiomatic design theory and QFD, J Clean Prod 39 (1) (2013) 79–86.
9. Z. Ghazalli, A. Murata, Development of an AHP-CBR evaluation system for re- manufacturing: end-of-life selection strategy, Int J Sustainable Eng 4 (1) (2011) 14.
10. S.G. Lee, S.W. Lye, M.K. Khoo, A multi-objective methodology for evaluating pro- duct end-of-life options and disassembly, Int J Adv Manuf Technol 18 (2) (2001) 148–156.

11. J. Ma, G.E. Okudan Kremer, A sustainable modular product design approach with key components and uncertain end-of-life strategy consideration, Int J Adv Manuf Technol 85 (1–4) (2016) 741–763.
12. S. Liyan, Reproducibility evaluation of mechanical and electrical products, Shan Dong University, 2008.
13. L. Li, H.H. Huang, F. Zhao, Z. Xiang, Q. Lu, Y. Wang, Z.F. Liu, J.W. Sutherland, Variations of energy demand with process parameters in cylindrical drawing of stainless steel, ASME J Manuf Sci Eng (2019), https://doi.org/10.1115/1.4043982.
14. X. Zhang, X. Ao, Z. Jiang, H. Zhang, W. Cai, A remanufacturing cost prediction model of used parts considering failure characteristics, Robot Comput Integr Manuf 59 (2019) 291–296.
15. P. Goodall, I. Graham, J. Harding, P. Conway, S. Schleyer, A. West, Cost estimation for remanufacture with limited and uncertain information using case based rea- soning, J Remanuf 5 (1) (2015) 7.
16. S. Sabharwal, S. Garg, Determining cost effectiveness index of remanufacturing: a graph theoretic approach, Int J Prod Econ 144 (2) (2013) 521–532.
17. F. Sang, E. Li, H. Zheng, S. Peijing, Prediction of remanufacturing cost combination of heavy-duty engine based on state, Surf Eng Remanuf 3 (2017) 18–22.
18. P.P. Qin, S. Yun, J.W. Shu, Estimation of the remanufacturing cost of engineering machinery hydraulic cylinder, 3D international conference on advanced information and communication technology for education, 2015.
19. H. Wang, Z.G. Jiang, H. Zhang, Y. Wang, Y.H. Yang, Y. Li, An integrated MCDM approach considering demands-matching for reverse logistics, J Clean Prod 208 (2019) 199–210.
20. X. Zhang, X. Ao, H. Zhang, Research on remanufacturing cost prediction model of used parts considering failure features, China Mech Eng 29 (21) (2018) 2594–2599.
21. J Zhang, Y Zhu, X Zhang, M Ye, J Yang, Developing a Long Short-Term Memory (LSTM) based model for predicting water table depth in agricultural areas, J of Hydrology, Elsevier, 561 (2018), 918-929.
22. KK Chandriah, RV Naraganahalli, RNN/LSTM with modified Adam optimizer in deep learning approach for automobile spare parts demand forecasting, Multimedia Tools and Applications, Springer, 80, (2021), 26145–26159.