Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 86 (2019) 68–73

www.elsevier.com/locate/procedia

7th CIRP Global Web Conference
"Towards shifted production value stream patterns through inference of data, models, and technology"

A deep learning based-decision support tool for solution recommendation in cloud manufacturing platforms

Alessandro Simeone[a]*, Alessandra Caggiano[b,c], Bin Deng[a], Lev Boun[d]

[a]*Intelligent Manufacturing Key Laboratory of Ministry of Education, Shantou University, China*
[b]*Department of Industrial Engineering, University of Naples Federico II, Italy*
[c]*Fraunhofer Joint Laboratory of Excellence on Advanced Production Technology (Fh-J_LEAPT UniNaples), Italy*
[d]*Dynamic Computer Systems, PoB 1341, 30300, Atlit, Israel*

* Corresponding author. Tel.: +8613411903575. *E-mail address:* simeone@stu.edu.cn

Abstract

Industry 4.0 key enabling technologies such as cloud manufacturing allow for the dynamic sharing of distributed resources for efficient use at industrial network level. Interconnected users, i.e. suppliers and customers, offer and request manufacturing services over a cloud manufacturing platform, where an intelligent engine generates a number of solutions based on functional and geometrical requirements. A high number of suppliers leads to a higher number of solutions available for customers increasing the decision-making complexity from a customer perspective. Recommendation systems play a crucial role in expanding the opportunities in decision-making processes under complex information environments. In this scope, this paper proposes the conceptualization and the development of a recommendation decision support tool to be implemented in a cloud manufacturing platform to assist customers in appropriately selecting manufacturing services with reference to sheet metal cutting operations. In terms of solution selection, a Deep Neural Network (DNN) paradigm is adopted to allow for the automatic learning of optimal solution recommendation list based both on customers past experiences and new choices. In this respect, a virtual interaction environment is firstly built for system pre-training. Subsequently, users' data are inputted in the pre-trained model to predict a recommendation list. This is then subject to user interaction, i.e. selection, which will be fed back into the model to update the training parameters. This paper concludes with a simulated case study reported to exemplify the proposed methodology for a variety of decision-making scenarios.

Keywords: Cloud manufacturing; Deep learning; Sheet metal cutting; Neural Network

1. Introduction

Cloud Manufacturing (CMfg) is an emerging key enabling technology (KET) in the modern Industry 4.0 context which offers the opportunity to set up smart manufacturing networks enabling the large-scale exchange of manufacturing services. In this framework, cloud computing, web technologies, service-oriented technologies and Internet of Things (IoT) are integrated to allow for on-demand access to intelligent manufacturing resources and manufacturing services through the cloud [1, 2]. In CMfg, a cloud platform can be operated to aggregate a pool of distributed resources offered by distinct providers to build up cloud manufacturing services delivered on-demand to several industrial users connected to the cloud following a service-oriented paradigm [3]. The broad sharing and on-demand-delivery via CMfg of distributed manufacturing resources (computational, software, digital and physical) boosts up the smart manufacturing networks through enhanced resource efficiency, greater productivity and utilization rates, supporting competitiveness in modern global market [2, 4–6]. One of the manufacturing sectors that could significantly benefit from the employment of CMfg is the sheet

10.1016/j.procir.2020.01.019

metal cutting industry, where the frequent changes of demand volumes and product variety negatively affect the resource efficiency [7]. The latter is essentially driven by the reduction of material waste and can be improved by applying a nesting strategy, i.e. by combining multiple sheet metal cutting orders on the same metal sheet so as to minimize the total waste material [7]. The setup of smart manufacturing networks via CMfg can facilitate nesting as several industrial users, in the role of customers and suppliers, are globally connected to a common platform and the chances to appropriately combine sheet metal manufacturing tasks are notably increased [4, 5].

The high number of manufacturing solutions available for customers via CMfg represents a great opportunity but, at the same time, increases the complexity of decision-making from a customer perspective. To tackle this issue, this paper proposes the development of a recommendation system to be implemented in a cloud manufacturing platform as decision support tool. Such system aims at assisting customers in appropriately selecting the manufacturing solutions with reference to sheet metal cutting operations. The system is based on the use of a Deep Neural Network (DNN) for the automatic learning of the optimal solution recommendation ranking based on customers past experiences and new choices. DNN requires minimum human expertise in features extraction and provides an improved learning ability as the dataset size and variety increases [8]. Literature provides diverse applications of DNN for recommendation systems. Covington et al. developed a DNN architecture for recommending YouTube videos, splitting the problem into two distinct sub-problems: candidate generation and ranking [9]. Cheng et al. [10] proposed a Wide & Deep learning approach, made of jointly trained wide linear models and DNNs to combine the benefits of memorization and generalization for recommender systems. Tkachenko [11] adopted a model-free Q-learning to train a DNN to map a client position in the state space to rewards associated with possible marketing actions.

In this paper, to implement deep learning aimed at solution recommendation based on users data and past choices, a virtual interaction environment is built for system pre-training. Afterwards, the pre-trained DNN model is fed with users data to predict a recommendation list. The latter is exposed to user interaction, i.e. selection, which is fed back to the model to update the training parameters. To illustrate the proposed methodology for a variety of decision-making scenarios, a simulated case study is finally reported.

2. Framework

The cloud manufacturing platform developed with reference to sheet metal cutting processes is provided with a GUI [5] through which any customer requiring a manufacturing service can input a customer instance, C_i, represented as follows:

$$C_i = \{ID_{ci}, Q_{ci}, t_{ci}, m_{ci}, T_{ci}, DL_{ci}, CAD_{ci}, L_{ci}\} \quad (1)$$

where ID_{ci} is the automatically assigned instance number, Q_{ci} the batch quantity (units), t_{ci} the part thickness (mm), m_{ci} the material, T_{ci} the cutting technology, DL_{ci} the customer deadline, CAD_{ci} the CAD file and L_{ci} the customer location.

Conversely, a supplier instance, S_i, is expressed as follows:

$$S_i = \left\{ID_{sj}, Q_{sj}, M_{sj}, m_{sj}, t_{sj}, A_{sj}, DL_{sj}, CAD_{sj}, L_{sj}\right\} \quad (2)$$

where ID_{sj} is the automatically assigned instance number, Q_{sj} the batch quantity (units), M_{sj} the machine tool, with details on the machine model, technology, tolerance, power consumption, cutting parameters, kerf width, availability and scrap rate [5], A_{sj} the metal sheet size, CAD_{sj} the CAD file of supplier jobs to complete by the deadline DL_{sj}, and L_{sj} the supplier location.

To select compatible instances, the cloud manufacturing platform, shown in Fig. 1, performs a functional compatibility assessment by verifying that materials, thicknesses and technologies offered by the supplier match those required by the customer and that the tolerances indicated by the customer are not tighter than those allowed by the supplier machine tool.

Afterwards, the set of compatible instances is processed by an optimization engine to generate suitable combinations of customer and supplier instances based on parts geometries so as to maximize the surface utilization rate (SUR), i.e. the ratio of utilised area over the whole sheet metal area. In this respect, considering the sheet metal cutting process nature, the problem is configurable as 2D nesting [7, 12–14]. Based on the output of the geometric optimization procedure, the cloud platform is hence able to propose to suppliers and customers a number of manufacturing solutions to choose from.

On the one hand, the main concern of the supplier is to select the most efficient solutions, i.e. customer pairing, in terms of SUR to minimize the scrap rate considering the deadlines of the manufacturing tasks. On the other hand, the customer decision is carried out according to a number of different criteria, such as SUR, energy consumption, processing time and distance cost. To enable the customers to perform an effective selection of manufacturing solutions, this paper presents an intelligent decision-making support tool, to propose the best manufacturing solutions ranked according to the preferred criteria. In a high scale manufacturing network context, e.g. at national level, a high number of suppliers implies a higher number of solutions available for customers, hence increasing the decision-making complexity from a customer perspective.

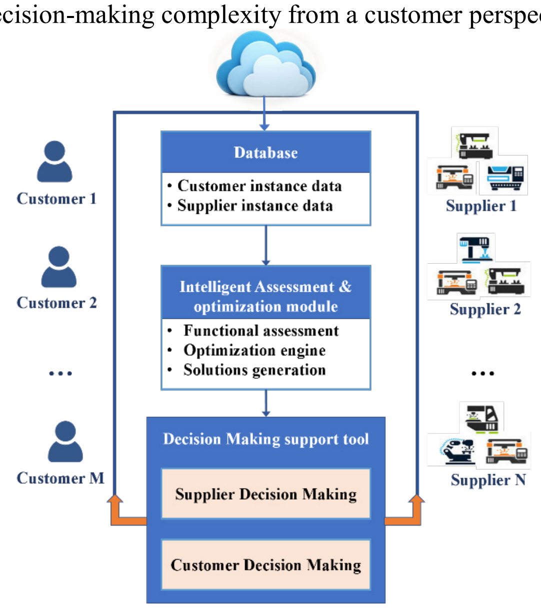


Fig. 1. Cloud manufacturing platform overview

To deal with this issue, a customer decision support tool is developed in this paper to recommend suitable manufacturing solutions via intelligent ranking. In this respect, a DNN [15] is used to capture patterns among the customers, suppliers, solutions and customer choices history.

Input features are represented by:

- Customer instance features CF_i

$$CF_i = [Q_{ci}, AT_{ci}, A_{ci}, p_{ci}] \quad (3)$$

where AT_{ci} is the available time given by the difference of the deadline and the submission date (hours), A_{ci} and p_{ci} are the area and the perimeter of the customer CAD file, respectively.

- Supplier instance features (SF_j)

$$SF_j = [Q_{sj}, v_{sj}, P_{sj}, A_{sj}, p_{sj}, R] \quad (4)$$

where v_{sj} represents the cutting speed, P_{sj} represents the machine power consumption with reference to the machine specifications defined in the supplier instance, A_{sj} and p_{sj} are the area and the perimeter of the supplier CAD file, respectively and R is the supplier rating.

- Solution features: $(SolF_k)$

$$SolF_k = [SUR_k, PT_k, E_k, D_k] \quad (5)$$

where SUR_k is the surface utilization rate, PT_k the processing time, E_k the energy consumption and D_k the distance cost.

Basic features about the material can be included as well:

$$MatF_k = [MC_k, t_k] \quad (6)$$

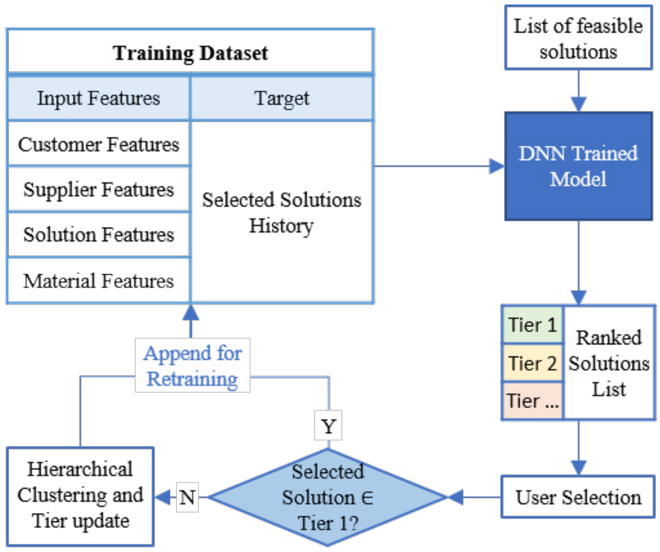
where MC_k is the material unitary cost, t_k the thickness.

By combining customer, supplier, solution and material features, an input feature vector made of 16 features is obtained. DNN training is performed to find patterns amongst customer and suppliers instances, solutions and users selection.

When inputting a new set, i.e. list of feasible solutions for a given customer, the DNN will output a ranked list of solutions (Fig. 2). The ranking problem is approached in this paper as a multi-class pattern recognition, where each class, i.e. Tier, represents the likelihood of a certain solution to be selected. Tiers are then sorted in descending order, meaning that Tier 1 solutions have more chances to be selected by the customer compared to Tier 2, Tier 3 and so on. The number of the Tiers can be set at cloud level based on the total number of available solutions, and as a general recommendation each Tier should not contain more than 25-30 solutions to allow for an easy handling by the customer.

The customer will select the best solution according to its own requirements. In case the customer will actually select a solution belonging to Tier 1, such selection will be appended in the training dataset to allow for a retraining in order to improve the learning capabilities.

When a customer selects a solution that does not belong to Tier 1, then a Tier update procedure needs to be carried out to incorporate the information regarding the customer selection applying a hierarchical clustering algorithm [16] as follows



.Fig. 2. Customer decision support tool flow chart

Using a bottom-up approach, each data point, i.e. solution feature vector starts in its own cluster, these clusters are then joined greedily, by taking the two most similar clusters together and merging them [16]. In this respect, the corresponding customer solutions list dataset, including customer, supplier and solutions features (see Table 1) is subject to a normalization procedure using z-score method [17] due to the physical and numerical heterogeneity of the considered features.

Subsequently, the pairwise Euclidean distance is computed between all the solution pairs [18], then a linkage measure is adopted to define clusters. In this work, a single-link distance measure was used [18]. In this way, the cluster to which the selected solution belongs will be updated as Tier 1, then assigning to the farther clusters a descending Tier rank.

The assessment of the DNN learning capabilities is carried out by evaluating the learning classification performance in terms of training and testing accuracy, as well as by comparing the customers selections with solutions ranked in the Tier 1. Being the solution recommendation problem configured as a classification task, the system accuracy can be expressed in classification success rate, i.e. the number of correctly classified cases over the total number of cases.

3. Case study

To exemplify the methodology proposed in this paper, a case study was simulated taking into account 30 customers. Each customer instance has the possibility to choose from 83 compatible solutions resulting from as many suppliers.

An excerpt of the input dataset is reported in Table 1. The following simplifying assumptions were taken into account to facilitate the understanding and the computation complexity, such as:

- All the solutions listed in the dataset are perfectly compatible with both suppliers and customers
- A single material and a single thickness value were used for all the customers and suppliers, therefore material features are excluded in this case study.

Table 1. Training Dataset.

ID	Input Features														Target Class
	Customer Features				Supplier Features						Solution Features				
	Q	AT	A	p	Q	v	P	A	p	R	SUR	PT	E	D	
$[C_1Sol_1]$	3200	144	48800	2520	4000	14	15.751	141691	1709	86	53.37	15.76	160.78	508	1
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
$[C_1Sol_{22}]$	3200	144	48800	2520	3000	13	7.5	60000	1100	82	42.93	25.25	125.83	768	2
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
$[C_1Sol_{43}]$	3200	144	48800	2520	3000	31	4.326	90000	2000	81	41.17	54.187	109.23	889	3
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
$[C_1Sol_{83}]$	3200	144	48800	2520	4000	14	15.751	141691	1709	79	53.37	15.76	151.91	1270	4
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
$[C_{30}Sol_{83}]$	6000	144	233827	1800	2400	13	8	443529	3038	78	63.91	40.20	135.12	19390	4

The 2D nesting was computed using Deepnest® software while the solutions were generated using Matlab® according to [4].

3.1. User selection criteria target definition

With reference to the case study datasets reported in Table 1, the simulation of the customer choices history was carried out by adopting the following criteria to build a ground truth:

$$\text{IF} \begin{cases} AT_{ci} \leq 96\ \text{h} \\ D_{Solk} < 550\ \text{CNY/Ton} \\ \dfrac{p_{ci}}{A_{ci}} > 0.01 \\ Q_{ci} \geq 6000 \\ Q_{ci} \leq 2000 \end{cases} \quad \text{THEN} \quad \begin{matrix} \min PT_k \\ \min D_k \\ \max R_{sj} \\ \max SUR_k \\ \min E_k \end{matrix} \qquad (7)$$

These criteria are meant to represent the likely choices for a variety of customer requirements scenarios listed below

- Customers who are in a hurry, i.e. with an available time not longer than 96 hours, will likely prefer solutions characterised by minimum processing time.
- Customers who can choose from a very close supplier will select the solution with least distance cost
- Customers with particular geometrical requirements, here identified by a geometrical complexity ratio, i.e. area-to-perimeter ratio higher than 0.01 will select suppliers with the highest Rating score
- Customers requesting a large batch size (more than 6000 units) will likely prefer solutions with very high SUR while customers requesting a very low batch size (less than 2000 units) will choose the solution corresponding to the minimum energy consumption.

In this way, target vectors were built for both training and test datasets (see Table 1), created by sorting the solutions according to the criteria listed in Eq. 7, then defining a 4-class target, splitting the target vector in four equal parts, namely Tier 1, Tier 2, Tier 3 and Tier 4.

The training was carried out by implementing a Back Propagation DNN [15], in Python™ environment, with the following architecture (Fig. 3): 14 input nodes corresponding to users and solution features as per Table 1, 4 Hidden Layers with 28 Nodes per Hidden Layer, Activation Function: Hyperbolic Tangent Activation Function (Tanh) [19], Output: SoftMax [19], Learning algorithm: gradient descent [20] with a Learning Rate = 0.02. The number of epochs was experimentally set to 6000 to avoid overfitting [19].

The dataset reported in Table 1 was split into a training and a test dataset, resulting in 20 customers solutions sets for training and 10 customers solutions sets for test. A graphical representation of the DNN architecture is reported in Fig. 3 utilizing a TensorBoard [21] representation.

4. Results and discussion

This section reports the results of the DNN training and testing, along with the tiers update via Hierarchical Clustering to show the system performance as new sets of data are added.

4.1. Training and Test #1

The DNN was trained using the 20 customers training dataset reported in Table 1. The training performance after 6000 epochs resulted to be 79.56%.

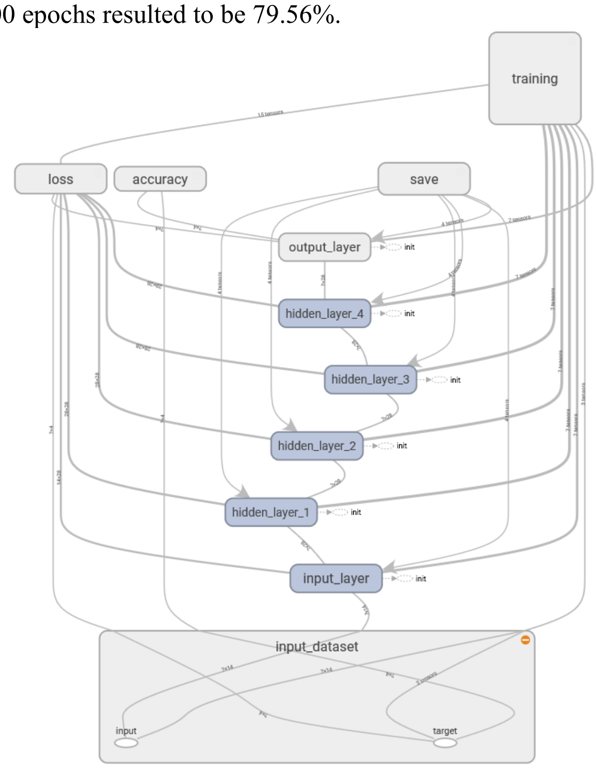


Fig. 3. DNN Architecture

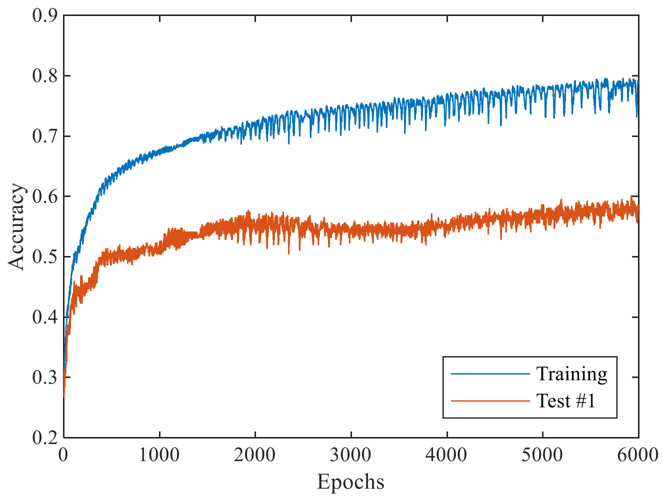


Fig. 4. Training and Test #1 DNN Performance

The trained model was used for a test on 3 new customers solutions list from Table 1, namely C_{21} C_{22} and C_{23}. The accuracy achieved for Test #1 resulted to be 60.12%. The training and Test #1 accuracies over the epochs are reported in Fig. 4. According to the customer selection criteria described in Eq. 7, the solutions selected by these three new customers in reported in Table 2, from which it results that two customers selected Tier 1 solutions while one customer selected a Tier 3 solution.

4.2. Tiers update

Following the Test #1, the results concerning the solutions selections for C_{22} and C_{23} shown in Table 2 indicate that the selected solutions belong both to Tier 1, this means that such results can be appended to the training dataset for retrained without any further processing.

As regards C_{21}, results show that this customer selected a solution which belongs to Tier 3. In this case, before appending the user selection, the Tiers update procedure needs to be applied. In this respect, the hierarchical clustering procedure described in section 2.2 was applied using Matlab® Machine Learning Toolbox to the list of available solutions for C_{21}. Such solutions were clustered according to the Euclidean distance and results are partially shown in the Dendrogram reported in Fig. 5. The dendrogram shows only 30 solutions to facilitate the visualization. In the chart it can be noted how the selected solution by C_{21}, namely $C_{21}Sol_8$, has now been updated and assigned to Tier 1. As the distance from Tier 1 increases, new clusters are formed defining progressively Tier 2, Tier 3 and Tier 4.

4.3. First Retraining and Test #2

Following the training dataset enlargement by appending the selection results related to C_{21} C_{22} and C_{23} , the system was retrained using 23 customers solutions sets. The retraining performance resulted to be 79.34%. A second test was carried out by inputting a new set of data related to 3 new customer solution sets, namely C_{24} C_{25} and C_{26}. The first retraining and test #2 results are reported in Fig. 6. In this case, according to the customer selection criteria described in Eq. 7, the results reported in Table 2 show that two customers, namely C_{24} and C_{26} have selected Tier 1 solutions while C_{25} selected a Tier 2 solution. In this scenario, the system performed a Tier update procedure for the solution set related to C_{25} as already done for C_{21} after Test #1.

4.4. Second Retraining and Test #3

The results from the first were appended to the training dataset which was retrained using 26 customers solutions sets. The retrain performance resulted to be 86.18% showing a sensible improvement if compared to the first training.

A third test was carried out by inputting a new set of data related to 4 new customer solution sets. In this test the accuracy is 62%. The second retraining and Test #3 accuracies over the epochs are reported in Fig. 7. The results reported in Table 2 show that all the four customers selected Tier 1 solutions.

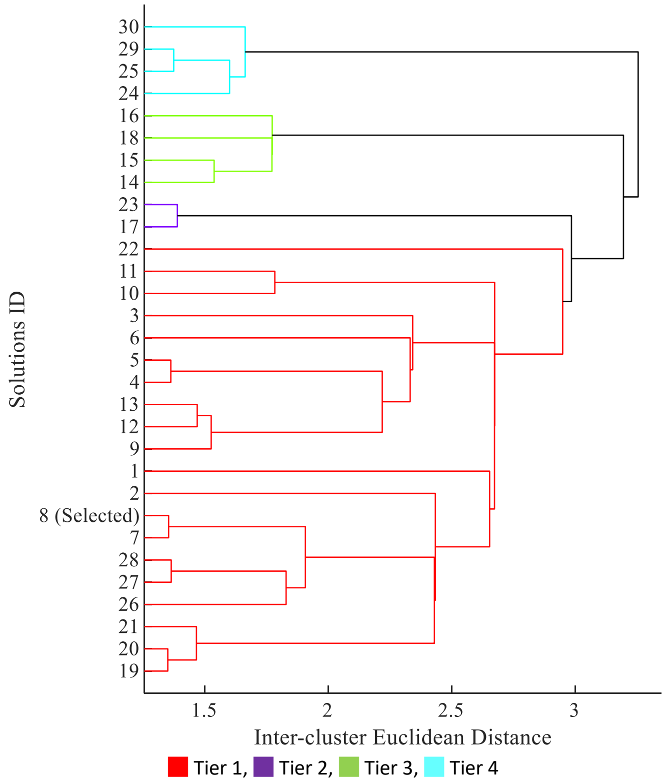


Fig. 5. Clustering and Tiers update results

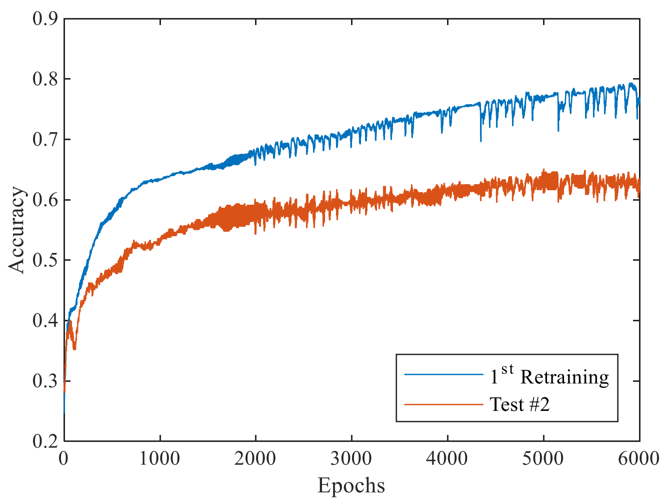


Fig. 6. 1st Retraining and Test #2 DNN Performance

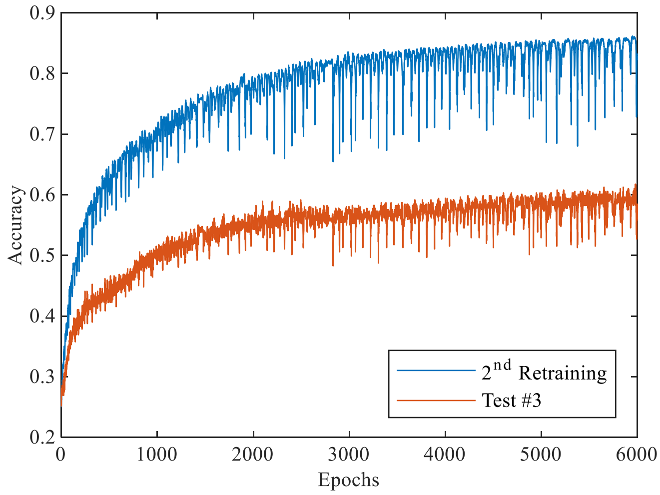


Fig. 7. 2nd Retraining and Test #3 DNN Performance

Table 2. DNN Test results.

Customer ID	Selected Solution ID	Selected Solution Tier		
		Test #1	Test #2	Test #3
C_{21}	$[C_1Sol_8]$	3		
C_{22}	$[C_{22}Sol_2]$	1		
C_{23}	$[C_{23}Sol_3]$	1		
C_{24}	$[C_{24}Sol_{63}]$		1	
C_{25}	$[C_{25}Sol_{54}]$		2	
C_{26}	$[C_{26}Sol_{38}]$		1	
C_{27}	$[C_{27}Sol_{21}]$			1
C_{28}	$[C_{28}Sol_{25}]$			1
C_{29}	$[C_{29}Sol_5]$			1
C_{30}	$[C_{30}Sol_2]$			1

Table 2 highlights the impact of dataset enlargement built by appending the customer selection along with the Tiers update and retraining on the customer solution selection with respect to the DNN Tier-based outputted list.

5. Conclusions

This paper proposes an intelligent decision-making support tool based on deep learning and continuous updating to be implemented in a cloud manufacturing architecture as a recommendation system for customer manufacturing solution selection with reference to sheet metal cutting operations. Automatic learning of the optimal solution recommendation ranking is based on customers past experiences and new choices. Despite a non-exceedingly outstanding performance in terms of learning accuracy, the system proved to be an effective decision-making support tool able to recognize patterns within user requirements and generated solutions, recording a positive trend in terms of training accuracy with the help of retraining.

Future works will focus on improving the system capabilities by adopting a larger dataset along with the development of alternative learning techniques such as reinforcement learning.

Acknowledgements

The research results presented in this paper are based on the activities carried out in the framework of the following projects:

- CLOUD MODE “CLOUD Manufacturing for On-Demand manufacturing sErvices” (000011-ALTRI _DR_3450_2016_RICERCA_ATENEO-CAGGIANO) funded by the University of Naples Federico II, Italy, within the “Programma per il finanziamento della ricerca di Ateneo” (2016-2019).
- Research Start-up Fund Subsidized Project of Shantou University, China (No. NFT17004).

References

[1] Zhang L, Luo Y, Tao F, et al. Cloud manufacturing: a new manufacturing paradigm. Enterp Inf Syst 2014; 8(2):167–87.
[2] Xu X. From cloud computing to cloud manufacturing. Robot Comput Integr Manuf 2012; 28(1):75–86.
[3] Li BH, Zhang L, Wang SL, et al. Cloud manufacturing: A new service-oriented networked manufacturing model. Comput Integr Manuf Syst CIMS 2010; 16(1):1–7.
[4] Simeone A, Caggiano A, Deng B, et al. Resource Efficiency Optimization Engine in Smart Production Networks via Intelligent Cloud Manufacturing Platforms. Procedia CIRP Elsevier 2018; 78:19–24.
[5] Simeone A, Caggiano A, Boun L, et al. Intelligent cloud manufacturing platform for efficient resource sharing in smart manufacturing networks. Procedia CIRP 2019; 79(July):233–38.
[6] Wu D, Rosen DW, Schaefer D. Scalability Planning for Cloud-Based Manufacturing Systems. J Manuf Sci Eng 2015; 137(4):040911.
[7] Verlinden B, Cattrysse D, Oudheusden D Van. Integrated sheet-metal production planning for laser cutting and bending. Int J Prod Res 2007;
[8] Wang J, Ma Y, Zhang L, et al. Deep learning for smart manufacturing: Methods and applications. J Manuf Syst 2018;
[9] Covington P, Adams J, Sargin E. Deep Neural Networks for YouTube Recommendations. 2016[Online] 2016.
[10] Cheng H-T, Ispir M, Anil R, et al. Wide & Deep Learning for Recommender Systems. Proceedings of the 1st Workshop on Deep Learning for Recommender Systems - DLRS 2016 2016[Online] ACM Press: New York, New York, USA 2016.
[11] Tkachenko Y. Autonomous CRM control via CLV approximation with deep reinforcement learning in discrete and continuous action space. arXiv Prepr arXiv150401840 2015;
[12] Gomes AM, Oliveira JF. Solving Irregular Strip Packing problems by hybridising simulated annealing and linear programming. Eur J Oper Res 2006;
[13] Herrmann JW, Delalio DR. Algorithms for sheet metal nesting. IEEE Trans Robot Autom 2001; 17(2):183–90.
[14] Adamowicz M, Albano A. Nesting two-dimensional shapes in rectangular modules. Comput Des 1976;
[15] Schmidhuber J. Deep Learning in neural networks: An overview. Neural Networks 2015[Online] 2015.
[16] Hennig C. Handbook of Cluster Analysis,. Handbook of Cluster Analysis 2015[Online] Chapman and Hall/CRC 2015.
[17] Nawi NM, Atomi WH, Rehman MZ. The Effect of Data Pre-processing on Optimized Training of Artificial Neural Networks. Procedia Technol 2013;
[18] Szekely GJ, Rizzo ML. Hierarchical clustering via joint between-within distances: Extending Ward’s minimum variance method. J Classif 2005;
[19] Agostinelli F, Hoffman M, Sadowski P, et al. Learning Activation Functions to Improve Deep Neural Networks 2014;
[20] Burges C, Shaked T, Renshaw E, et al. Learning to rank using gradient descent. Proceedings of the 22nd international conference on Machine learning - ICML ’05 2005[Online] 2005.
[21] Nelli F, Nelli F. Deep Learning with TensorFlow. In: Python Data Analytics 2018;