Available online at www.ijapie.org

INTERNATIONAL JOURNAL OF ADVANCED PRODUCTION AND INDUSTRIAL ENGINEERING

IJAPIE-2020-07-341, Vol 5 (3), 01-10

https://doi.org/10.35121/ijapie202007341

IJAPIE
Connecting
Science & Technology
with Management.

A Journal for all
Products & Processes.

Predicting Production Cycle Time in a Real Disposable Medical Device Manufacturing System Using Semi-Supervised Deep Learning

Himansh Chandani[1], Mayank Tyagi[1], Rajiv Chaudhary[1], and Ranganath M S[1]

Delhi Technological University, Delhi, India

*Email: rch_dce@rediffmail.com

ABSTRACT: Manufacturing lot cycle time is the period required by a manufacturer for completion of a production process. It is an essential factor for determining the success of most manufacturing organizations, yet most research is based on studies made almost exclusively in the semiconductor industry and does not attempt to utilize the complete potential of recent breakthroughs in computational learning. Using real data collected from a medical device manufacturing company, this paper demonstrates the applicability of a semi-supervised deep learning frame-work for highly accurate cycle time prediction, using stacked Denoising Autoencoders to form fully connected deep neural networks and Convo-lutional Neural Network models. The proposed strategies for cycle time prediction can have significant impact on product design decision opti-mization within the system which, in turn, facilitates reduction of costs, energy use and the overall environmental impact.

Keywords: Cycle time prediction· Manufacturing · Deep learning regression, Semi-Supervised learning.

I. INTRODUCTON

Optimization of time utilization is an essential objective for most manufactur-ers and accurate predictions of cycle time (CT) can help provide accurate time quotes to customers, optimize speed of production, further contributing to in-creased performance, reduced costs and environmental impact.

Predictive data analysis is the process of making estimates about future outcomes through useful information and trends extracted from large amounts of historical data [1]. Aided by new technologies like the Internet of Things (IoT), cloud computing and the presence of elaborate data collection practices in modern manufacturing, data availability has grown multi-fold [2] thus enabling strategic use of data mining and machine learning for predictive data analysis. The CRISP-DM (Cross industry standard process for data mining) model defines the main stages of a predictive data analysis project as: i) business understand-ing, ii) data understanding, iii) data preparation,iv) modelling, v) evaluation and vi) deployment. Modelling (iv) utilizes machine learning (ML) for modelling

of predictive processes. The models giving best overall performance on the spec-ified task can be tested for deployment to real world manufacturing systems. ML refers to a series of systems concerning development of computational tech-niques that recognize information and trends in data, facilitating prediction, classification and other tasks associated with artificial intelligence [3–5].

Previous methods that rely on heuristics and rules of thumb (e.g., studying a small number of process parameters like work in process (WIP) and throughput rates for estimation [6, 7]) tend to be unreliable due to the complexity of the problem of CT prediction. Moreover, the available literature on CT prediction is mainly limited to the semiconductor industry and uses simulated data for model development [8–11]. Methods developed using real

data and more holistic approaches [12–14] are still limited by strategies and techniques that don't at-tempt to fully realize the capabilities of recent breakthroughs in computational learning. The research overlooks the relevance of product design factors for CT estimation and consistently aims to reduce the number of dimensions in the data –essentially eliminate features– and disregard the resulting loss of information which leads to unreliable generalised performance. CT prediction can be en-hanced by using methods that generalize well to new data (data that has not been seen and trained on by the model beforehand),by avoiding information loss due to dimensional reduction and by including elements of product design in the predictive analysis.

In this paper we focus particularly on Deep learning. Deep learning (DL) is a segment of ML techniques that enable learning of patterns of multi-level abstraction within data using computational models having multi-layered pro-cessing capabilities and are known to generalize well to unforeseen data [15]. For our goal, there can be a number of possible approaches: i) Supervised DL, for la-belled data, (ii) Unsupervised DL, for unlabelled data and (iii) Semi-supervised learning, for a mixture of labelled and unlabelled data.

In this paper, we present an acquisition strategy for semi-labelled data which includes elements of product design data and suggest the use of semi-supervised DL techniques, particularly, Autoencoder [16] pre-training [17] and stacking to give a simple feed-forward deep neural network (DNN) with seven hidden layers, further co-trained with a Resnet-18 [18] convolutional neural network (CNN) for accurate prediction of product cycle times using real data from a medical device manufacturing company. We also present a comparison of the developed technique with various existing state-of-the-art ML methods.

This paper is structured as follows: Section 2 describes the relevant litera-ture, including above stated methods for prediction of manufacturing cycle time. Section 3 focuses on the proposed methodology for formulating various semi-supervised regression methods which are then demonstrated for the collected data. In conclusion we discuss the key outcomes and future research capabili-ties.

II RELATED WORK

In this section we review the literature on ML methods used for CT estimation and describe the motivation for this work.

Manufacturing time prediction using machine learning: State-of-the-art and motivation

Raaymakers et al. [12] compared regression methods with neural networks for es-timation of the makespan of job sets in batch-process industries. Chien et al. [7] provided a method for predicting cycle time based on data mining algorithms using production status like WIP and throughput rates. Alenezi et al. [13] ad-dressed the problem of real flow-time prediction in make to order manufacturing using support vector machine regression and compared its performance to that of a simple one-layer ANN. Chen et al. [8] proposed non-linear quantile regres-sion for modelling the relationship between stationary cycle time quantiles and corresponding throughput rates of a manufacturing system.

One of few the papers that stood out, both in terms of the industry of study (aerospace) and the method used, was Juez et al. [19]. The authors eval-uated a regression technique for estimation of lead times of parts using the Cox model [20], a proportional hazards model and compared the performance to that of an SVM. Meidan et al. [9] compared the results of a selective naive bayesian classifier (SNBC), an artificial neural network (ANN) with a single hidden layer, multivariate linear regression and decision trees for classification of semiconduc-tor cycle time into low, medium and high categories with best average accuracy 73.2% achieved using the ANN. The use of ANNs for deployment however, was not recommended by this research due to computational instability and long training times; both of which have diminished greatly, following the extensive research in recent years by the deep learning community [15, 21, 22].

Li et al. [11] executed analytical queuing analysis and a stepwise regression model on data acquired using a job shop simulation, to determine relationships between job shop state at job start time and the resulting lead time.

Pfeiffer et al. [10] used discrete event simulation of a small scale parallel flow shop type production system, to predict lead time and identify most effective predictor features. The analysis revealed that random forest (RF) models out-performed linear regression and decision tree regression models. The degree of applicability of the model to real data is questionable and the authors note the possibility of decrease in deployment performance owing to high sensitivity of RF models to values that lie outside expected boundaries.

Lingitz et al. [14] used real world MES data from a semiconductor manufac-turing system for selection of key predictors and models, for estimation of manu-facturing lead time, provided comprehensive comparison of numerous regression techniques with multiple evaluation measures and recommended Random Forest regression model with eight features for most accurate prediction.

A survey of the domain literature demonstrates some methodology bias. It is observed that most of the recent research is: i) focused on prediction of man-

ufacturing lead time/product life and not cycle times, and neglects the impact that product design, development and implementation can have on the time con-sumed in a manufacturing system, ii) primarily based on studies made in the semiconductor industry or simulation data ,iii) limited by conventional methods, reasoning and experimentation strategies that don't attempt to utilize the full potential of the recent breakthroughs in the artificial intelligence community and

iv) attempting to reduce the dimensions of data to a handful of features hardly accounting for the information loss incurred.

III PROPOSED METHODOLOGY

The CRISP-DM (Cross industry standard process for data mining) model de-fines the main stages of a data mining project, independent of the industry of application as: i) business understanding, ii) data understanding, iii) data prepa-ration, iv) modelling, v) evaluation and vi) deployment. This research is focused on all stages of the CRISP-DM model, with special attention given to stages (iv) and (v).

Production system and data description

The study was conducted using data from a medical product manufacturing sys-tem, producing 72 different kinds of intravenous catheters, 81 different kinds of intravenous tubing sets and extension tubes, 3 different kinds of oxygen delivery sets, 8 kinds of needles, umbilical cord clamps, gynecological Karman catheters, feeding tubes and surgical items like blades, scalpels, gloves and suction sets. The company houses both- special purpose machines (SPM) and general pur-pose machines for development and production. The type of production can be categorized as made-to-order batch production system. The machines are grouped according to the manufacturing process/stage and production takes place in large batches. A process schematic for the production system is given in Figure1. Each component/assembly passes through the different process stages only once, with the only exception being in the case of defective packaging where such items are separated in the inspection stage, opened and repacked. An inter-esting characteristic of this manufacturing system is that new products/variants are frequently added to production depending on advancing technological capa-bilities, changing market trends and customer requirements. This adds a layer of complexity in the manufacturing process as available resources and equip-ment are required for both maintaining the current level of production as well as running tests and prototypes for new product development.

The multi-phase design process for new products adopted by the studied company follows a sequential order: i) concept development, ii) system level design, iii) detailed design, iv) prototyping, v) testing and refinement and vi) production. Design of new or improved products is therefore a highly complex and fairly iterative task. The company develops most of the tools used for plastic injection moulding and extrusion in-house through its own tool making work-shop equipped with machines performing all required machining operations. Tool making is in itself a highly complex task comprised of many diverse iterative operations and each added component requires a new tool for production. Difficult decisions need to be made along the entire design process to turn concepts into realities [23]. All products stated above use some varying combination of plastics and/or rubber polymers as raw material. Needless to say, many of the design decisions and activities have a direct impact on the resulting cycle times. Severe problems and delays can arise from well-intentioned adjustments in the design process [24, 25]. Concurrent engineering is known to quicken the commencement of the development cycle which would otherwise be done later but it is also known to increase the time and cost for reworking and concurrency among different functions. A substantial amount of detailed data about the manufacturing process, products, equipment and past production was readily available for compilation owing mainly to the company's advanced data collection and representation processes implemented within the enterprise resource planning (ERP) and man-ufacturing execution software (MES) systems. Relevant data pertaining to the operations like production start and end times (used for calculating total cycle time), production line information, vendor turn around time, stock

information and stage wise production plans and data about product design like designer in-formation (ID), start and end date of development process, relevant dimensions, volumes, mass of the products, raw materials used, tool number, machines used for tool development and prototyping (IDs), etc. was provided by the company for a period of 6 years. Out of these, about one-fourth (2703 of 8200) of the production start and end time values (labels) were available for the data points and the rest were unlabeled.

Table 1 describes the attributes of the raw data. The first column gives the name of the data variable, while the second column gives a sense of the scope of the data – range/number of unique values it can take. The collected data included 192 different products each having a special product code/ID. The in time and out time for the manufacturing process was taken to be the time at which the first Bill of Materials (BOM) is generated for a production batch and the time at which the entire batch reaches its final destination (FGS) before dispatch. Start time is subtracted from end time to give total cycle time in hours.

Table 1: Features in the raw data.

Name	Range/Scope
Product ID	192
In time (BOM)	Date-time
Out time (FGS)	Date-time
Batch ID	8200
Process routes	39
Quantities	1-10
Machine ID	288
Tool ID	396
Dev start	Date
Dev complete	Date
Dev Rework cycles	3-34
Dimensions	192
No. of components	4-22
Designer ID	19
No. of tool cavities	4-64
Prototyping equipment used	65
Type of Runner in PIM	12
Type of Gates in PIM	27
Raw Materials used	22
Vendor turn around	Days
Batch importance	low/mid/high
Inventory status	3

Lot quantities were divided into 10 categories, 1 being the smallest and 10 being the largest. Each of the different products contain 2-15 individual com-ponents which are molded/cut/extruded separately using the raw materials and assembled in the assembly stage. Tool ID refers to the list of unique alphanumeric identifiers of PIM/Extrusion machine tools designed for making each component of the different products. No. of tool cavities is the number of components put out in one single cycle of operation by a machine, for PIM this number is usually a power of 2 with few exceptions. Inventory status refers to the state of available stock of materials for utilisation in the production process before the start time. This status can take the values of low, moderate or satisfactory. The product and manufacturing data was collected and used to generate a semi-labelled data set for semi-supervised machine learning. This data set consisted of 8200 ob-servations, 5497 of which did not include cycle time values (target values)–thus forming the unlabelled data set– and 285 features derived from the raw data. The labelled data set was split by random sampling into a train/validation/test split ratio of 70/10/20 for the purpose of model training and evaluation. Total cycle times were calculated (in hours) by subtracting In time from Out time and their frequency distribution is represented in Fig. 2.

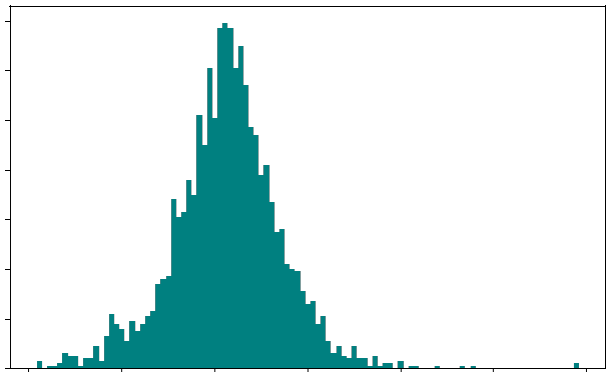

Fig. 2: Histogram of cycle time (target) values (in hours)

3.2 Selection of Algorithms

Machine learning is concerned with a large set of tools that can be used for understanding data. These tools can be classified as supervised, unsupervised and semi-supervised learning. Supervised learning involves developing statistical models for estimating an output from a given set of inputs utilizing a given set of n labelled data observations $L = ((x_i; y_i))^n_{i=1}$. By training on these data points, supervised learning methods infer a function from inputs to targets that can successfully determine the label y of a never-before-seen input x. In numerous real-world applications however, practitioners usually also have access to a set of m unlabeled data points

$$U = ((xi))^{m+n} \quad .$$

Unsupervised methods can be

$$i=n+1$$

trained using these unlabelled data points to learn relations and structures to enhance understanding of the data. Semi-supervised learning methods attempt to integrate the first two method groups to develop models whose performance transcends that of models developed only using the labelled data.

An important class of unsupervised learning methods used for semi-supervised learning are clustering algorithms. Clustering algorithms attempt to group or cluster objects according to discovered or inferred intrinsic characteristics. One of the most widely used clustering methods called K-means clustering [26] aids in discovery of the natural groupings in the data. The most critical parameter required by the algorithm is the number of clusters for grouping, K. Usually, mul-tiple experiments are conducted with different values of K and a suitable number is chosen based on the results. Autoencoders are learning methods that attempt to learn latent representations of data by transforming inputs into outputs while minimizing the degree of distortion from the inputs. These are another class of methods that have gained significance in unsupervised learning since their in-troduction in [16]. Each autoencoder layer can be pre-trained in turn to extract useful higher-level representations spanning across multiple autoencoders, the layers from all of which can be stacked to form the starting point for an ANN that is likely to give much better prediction performance in terms of generaliza-tion [27]. A denoising autoencoder (DAE) reconstructs a clean "repaired" version of the input from a corrupted version.This is done by first adjusting the input x to x^ by utilising a stochastic mapping x^ Q_L(^xjx). The corrupted input is then mapped to a latent representation of x^, having a different dimensionality than that of x^, which is further used for the reconstruction of x. The reconstruc-tion of x from x^ forces the network to learn the geometric "manifold" [28] of x. See Vincent et al. [27] and Erhan et al. [17] for an in depth discussion on this phenomenon.

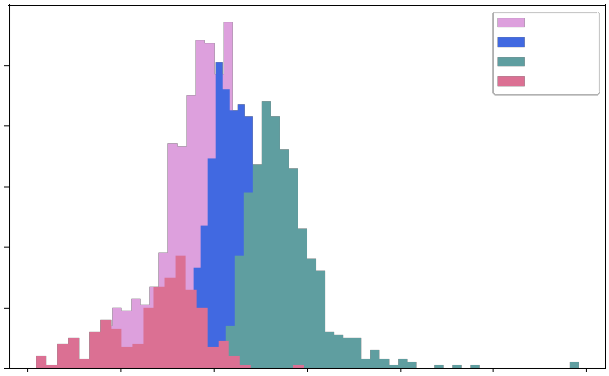

Fig. 3: Histogram of K-means clustered target values (in hours)

Regression methods attempt to predict a continuous unknown value (in our case, cycle time) given a series of other known values (independent variables). The most commonly used regression method is linear regression, specifically multiple linear regression due to it simplicity and ease of understanding. This assumes a linear relationship between inputs and outputs and cannot be used ef-fectively to capture non-linear characteristics. Modelling non-linear relationships can be achieved using intrinsically non-linear techniques such as deep neural networks (DNN), quantile regression and ensemble decision tree methods such as gradient boosting machines (GBM) and random forests (RF). Quantile re-gression attempts to map non linear relationships and predict specific quantiles (intervals of the target variable) rather than a single value [8]. Decision trees are easy to interpret and implement predictive methods based on conditional logic. Although these can handle various kinds of predictors without pre-processing, they are unstable, highly sensitive to changing data and offer less than optimal predictive performance. A solution to this is using tree ensemble techniques such as Random Forests and GBMs that output a combination of a number of pre-dictions from different trees as the final prediction. GBMs are methods in which a number of weak predictors are aggregated (or boosted) to produce an ensem-ble predictor with superior performance. RF is a tree ensemble technique which randomly selects predictors to use for construction of trees in the ensemble in an attempt reduce correlations among the predicting variables.

Modelling and evaluation Six different regression techniques were trained in conjunction with two different unsupervised methods on the data set using Python 3 in Jupyter Notebooks [29]. Five evaluation metrics were chosen for the models – mean squared error (MSE), root mean squared error (RMSE), normalised root mean squared error (NRMSE), mean absolute error (MAE) and mean absolute percentage error (MAPE). The objective function used for all the regression tasks was MSE, as shown by Equation 1. The evaluation metric for quantile regression was chosen to be percentage accuracy of prediction(testing what percentage of actual test values lied in the predicted intervals).

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(\hat{y}_t - y_t)^2 \quad (1)$$

where $\hat{y}_t$ is the set of predictions at timestep t and y_t is the set of target values.

Table 2: Evaluation of tested methods using various metrics

Unsupervised Method	Supervised Method	MSE	RMSE	NRMSE	MAE	MAPE
Autoencoder Pretraining	DNN	1292.71	35.95	6.16	26.64	3.77
	avg(DNN,Resnet18)	071.34	32.73	5.75	23.96	3.41
	Resnet18	1315.11	36.26	6.37	26.09	3.69
K-means preprocessing	LR	3743.67	61.18	10.49	45.45	6.37
	RF	2367.84	48.66	8.55	35.29	4.95
	GBM	4836.21	69.54	11.92	53.60	7.57

Table 3: Evaluation of quantile regression using prediction accuracy

	Q_1	Q_2	Q_3	Q_4
	(y 683:95)	(683.95 < y 713:75)	(713.75 < y 747:092)	(713.75 < y)
RF Quantile Regression	83.6	94.8	86.4	82.0

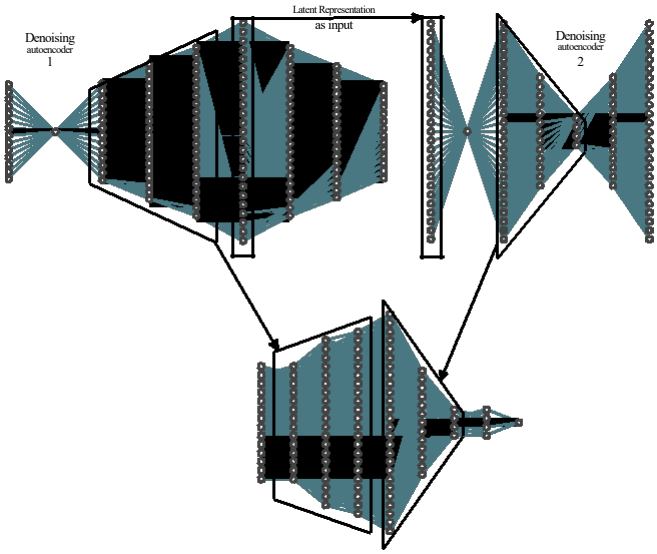

Fig. 4: Stacking of pretrained denoising autoencoders to form DNN architecture: Two denoising autoencoders were trained on unlabelled data, attempting to learn latent structures and relationships (data manifold [28]) among the predictor features. The encoders from these were stacked together as shown, and further trained on labelled data to enhance prediction performance.

Maintaining simplicity of implementation and an unbiased analysis strategy, all of the 285 features were included in the evaluation of the different models. K-means clustering was used for implementation of an inductive unsupervised preprocessing strategy. The model was trained on the labelled data set and was then used to learn the natural groupings in the labelled samples and further training a classifier for the unlabelled samples for various values of K. K=4 was selected as the optimal number of clusters for the given data points. The clusters for labelled values of cycle time are visualized as a histogram in Fig 3(b). Cluster separated data points were then used to train four different LR, RF and GBM models. Two autoencoders were pre-trained on unlabelled data set and then stacked (See fig. 4) to form a DNN with seven hidden layers which was trained on labelled data set for the task of regression. This DNN was further co-trained [30] using the unlabelled data set in conjunction with a CNN model with the architecture of Resnet18 [18].

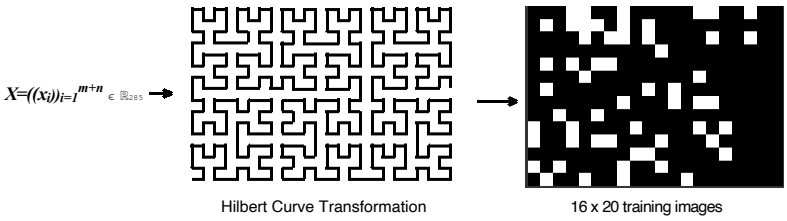

Fig. 5: Hilbert curve transformation of 1D data to images while preserving spatial information

The Resnet18 CNN architecture was trained on image samples generated from the data set where each data point was used to generate an image of size 16 pixels x 20 pixels based on a Hilbert curve transformation (See Fig.7) from one-dimensional to two-dimensional space. The Hilbert curve is a space filling curve which has been used extensively for preservation of spatial information during inter-dimensional conversions [31, 32]. The evaluation results for the different models are summarised in Tables 3 and 4. The optimal number of estimators in ensemble for RF was found to be 1200, for GBM classifiers it was found to be 80 or 90 estimators, varying by cluster and for GBM regressors it was found to be 120. The best prediction performance and lowest error rates were acheived by averaging the test set predictions of the seven-hidden-layer DNN and Resnet18 CNN models. Individually as well, the DNN and CNN regression models outperformed the linear and tree based methods by a significant margin. The architectures of the two pre-trained autoencoders and the resulting stacked DNN are represented in Fig. 4. The autoencoder models were trained for 400 epochs each with learning rates in the range (3 10 3; 3 10 9) which took a total of 6 minutes and 31 seconds. The stacked DNN was further trained for regression using the labelled data set for 90 epochs with learning rates in the range (3 10 3;3 10 7) for a total of 2 minutes and 25 seconds. Resnet18 was 4 for 21 minutes trained for 53 epochs with a constant learning rate at 3 10 and 3 seconds. This model training was done using the ADAM [33] optimization algorithm with a mini-batch size of 16 on an Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz. Data I/O and preprocessing was carried out using the Pandas [34], Numpy [35] and Scikit-learn preprocessing [36] libraries for Python while LR, DNN and CNN methods were implemented using the Pytorch [37] and FastAI [38] libraries. RF tree ensembles and K-means algorithms were implemented using the Scikit-learn [36] library for Python. Our final recommendation for predicting cycle time this specfic use case is averaging individual predictions from a seven layer DNN

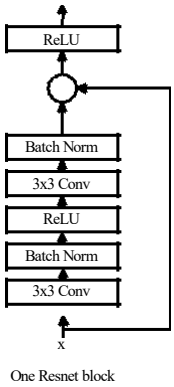

Fig. 6: Single Resnet block

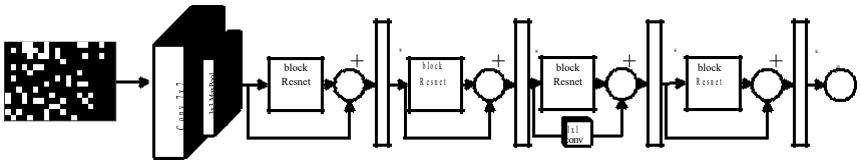

Fig. 7: Resnet18 architecture

IV CONCLUSION

The current research explored production cycle time prediction using a data set collected from a real medical device manufacturing system consisting of 2703 labelled and 5497 unlabelled samples through semi supervised methods and demonstrated the combined applicability of pre-trained Autoencoders, stacked and co-trained DNN and CNN models for the same. Our final recommendation for predicting cycle time this specfic use case is averaging individual predic-tions from a seven layer stacked DNN developed from Autoencoders trained on unlabelled data including all available predictors and further fine tuned on la-belled data, in conjunction with a Resnet18 CNN model trained on Hilbert curve transformation generated images also using all the available predictor features. Individually, predictions by the seven layer DNN achieved NRMSE at 6.16 and MAPE at 3.77, the CNN achieved an NRMSE at 6.37 and MAPE at 3.69, while when the predictions from these two models were averaged, the best average NRMSE and MAPE of 5.75 and 3.41 were achieved. These promising outcomes exhibit the potential for application of the developed semi supervised regression

REFERENCES

[1] Daniel T Larose. Data mining and predictive analytics. John Wiley & Sons, 2015.
[2] Fei Tao, Qinglin Qi, Ang Liu, and Andrew Kusiak. Data-driven smart manufac-turing. Journal of Manufacturing Systems, 48:157–169, 2018.
[3] László Monostori, András Márkus, Hendrik Van Brussel, and E Westkämpfer. Ma-chine learning approaches to manufacturing. CIRP annals, 45(2):675–712, 1996.
[4] Thorsten Wuest, Daniel Weimer, Christopher Irgens, and Klaus-Dieter Thoben. Machine learning in manufacturing: advantages, challenges, and applications. Pro-duction & Manufacturing Research, 4(1):23–45, 2016.
[5] Shai Shalev-Shwartz and Shai Ben-David. Understanding machine learning: From theory to algorithms. Cambridge university press, 2014.
[6] Y Narahari and LM Khan. Modeling the effect of hot lots in semiconductor manufacturing systems. IEEE Transactions on Semiconductor Manufacturing, 10(1):185–188, 1997.
[7] Chen-Fu Chien, Chih-Wei Hsiao, Cheng Meng, Kuo-Tong Hong, and Szu-Tsung Wang. Cycle time prediction and control based on production line status and manufacturing data mining. In ISSM 2005, IEEE International Symposium on Semiconductor Manufacturing, 2005., pages 327–330. IEEE, 2005.
[8] Nan Chen and Shiyu Zhou. Simulation-based estimation of cycle time using quan-tile regression. IIE Transactions, 43(3):176–191,2010
[9] Yair Meidan, Boaz Lerner, Gad Rabinowitz, and Michael Hassoun. Cycle-time key factor identification and prediction in semiconductor manufacturing using machine learning and data mining. IEEE transactions on semiconductor manufacturing, 24(2):237–248, 2011

[10] András Pfeiffer, Dávid Gyulai, Botond Kádár, and László Monostori. Manufactur-ing lead time estimation with the combination of simulation and statistical learning methods. PROCEDIA CIRP, 41:75–80, 2016
[11] Minqi Li, Feng Yang, Hong Wan, and John W Fowler. Simulation-based experi-mental design and statistical modeling for lead time quotation. Journal of Manu-facturing Systems, 37:362–374, 2015
[12] Wenny HM Raaymakers and AJMM Weijters. Makespan estimation in batch pro-cess industries: A comparison between regression analysis and neural networks. European Journal of Operational Research, 145(1):14–30, 2003
[13] Abdulrahman Alenezi, Scott A Moses, and Theodore B Trafalis. Real-time predic-tion of order flowtimes using support vector regression. Computers & Operations Research, 35(11):3489–3503,2008
[14] Lukas Lingitz, Viola Gallina, Fazel Ansari, Dávid Gyulai, András Pfeiffer, and László Monostori. Lead time prediction using machine learning algorithms: A case study by a semiconductor manufacturer. PROCEDIA CIRP, 72:1051–1056, 2018
[15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. nature, 521(7553):436–444, 2015
[16] DE Rumelhart, GE Hinton, and RJ Williams. Learning internal representations by error propagation, parallel distributed processing, vol. 1. Foundations. MIT Press, Cambridge, 1986.
[17] Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pas-cal Vincent, and Samy Bengio. Why does unsupervised pre-training help deep learning? Journal of Machine Learning Research, 11(Feb):625–660, 2010.
[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.
[19] Francisco Javier De Cos Juez, Fernando Sánchez Lasheras, Ana Suárez Sánchez, Pedro Riesgo Fernández, and Paulino Jose Garcia Nieto. Determination and study of lead times of metallic components in the aerospace industry through a cox-type hazard model. International Journal of Computer Mathematics, 89(13-14):1901– 1913, 2012.
[20] David R Cox. Two further applications of a model for binary regression. Biometrika, 45(3/4):562–565, 1958.
[21] Zhilu Chen, Jing Wang, Haibo He, and Xinming Huang. A fast deep learning system using gpu. In 2014 IEEE International Symposium on Circuits and Systems (ISCAS), pages 1552–1555. IEEE, 2014.
[22] Samuel L Smith, Pieter-Jan Kindermans, Chris Ying, and Quoc V Le. Don't decay the learning rate, increase the batch size. arXiv preprint arXiv:1711.00489, 2017.
[23] Richard Morris. The fundamentals of product design. Bloomsbury Publishing, 2016.
[24] Nelson P Repenning. Understanding fire fighting in new product development. Journal of Product Innovation Management: An International Publica-Tion Of The Product Development & Management Associa-Tion, 18(5):285–300, 2001.
[25] Amiya K Chakravarty. Overlapping design and build cycles in product develop-ment. European Journal of Operational Research, 134(2):392–424, 2001.
[26] SP Lloyd. Least square quantization in pcm. bell telephone laboratories paper. published in journal much later: Lloyd, sp: Least squares quantization in pcm. IEEE Trans. Inform. Theor.(1957/1982), 18, 1957.
[27] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful represen-tations in a deep network with a local denoising criterion. Journal of machine learning research, 11(Dec):3371–3408, 2010.
[28] Olivier Chapelle and Alexander Zien. Semi-supervised classification by low density separation. In AISTATS, volume 2005, pages 57–64. Citeseer, 2005.
[29] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Jason Grout, Sylvain Corlay, Paul Ivanov, Damián Avila, Safia Abdalla, and Carol Will-ing. Jupyter notebooks – a publishing format for reproducible computational work-flows. In F. Loizides and B. Schmidt, editors, Positioning and Power in Academic Publishing: Players, Agents and Agendas, pages 87 – 90. IOS Press, 2016.
[30] Zhi-Hua Zhou and Ming Li. Semi-supervised regression with co-training. In IJCAI, volume 5, pages 908–913, 2005.
[31] Jonathan K. Lawder and Peter J. H. King. Querying multi-dimensional data in-dexed using the hilbert space-filling curve. ACM Sigmod Record, 30(1):19–24, 2001.
[32] Baback Moghaddam, Kenneth J Hintz, and Clayton V Stewart. Space-filling curves for image compression. In Automatic object recognition, volume 1471, pages 414– 421. International Society for Optics and Photonics, 1991.
[33] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.
[34] Wes McKinney. Data structures for statistical computing in python. In Stéfan van der Walt and Jarrod Millman, editors, Proceedings of the 9th Python in Science Conference, pages 51 – 56, 2010.
[35] Travis Oliphant. NumPy: A guide to NumPy. USA: Trelgol Publishing, 2006
[36] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12:2825–2830, 2011.
[37] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Des-maison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, Advances in Neural Information Processing Systems 32, pages 8024–8035. Curran Associates, Inc., 2019.