Probabilistic Models for Manufacturing Lead Times

Recep Yusuf Bekci
McGill University
Montreal, Canada
recep.bekci@mail.mcgill.ca

Yacine Mahdid
Axya
Montreal, Canada
yacine@axya.co

Jinling Xing
Axya
Montreal, Canada
jinling@axya.co

Nikita Letov
McGill University
Montreal, Canada
nikita.letov@mail.mcgill.ca

Ying Zhang
McGill University
Montreal, Canada
ying.zhang8@mail.mcgill.ca

Zahid Pasha
Axya
Montreal, Canada
pasha@axya.co

Abstract

In this study, we utilize Gaussian processes, probabilistic neural network, natural gradient boosting, and quantile regression augmented gradient boosting to model lead times of laser manufacturing processes. We introduce probabilistic modelling in the domain and compare the models in terms of different abilities. While providing a comparison between the models in real-life data, our work has many use cases and substantial business value. Our results indicate that all of the models beat the company estimation benchmark that uses domain experience and have good calibration with the empirical frequencies.

1 Introduction

Cost estimation has a fundamental function for manufacturing operations. Typically, the total cost of manufacturing a part is a sum of different elements such as material cost, labor cost, engineering cost, and operation costs. In this study, we collaborate with a metal manufacturing firm whose cost structure is the sum of the material cost and the cost of laser operations. In this case, the cost of laser operations is volatile and it is not easy to determine this element before completing the production process, nevertheless, the material costs have a deterministic structure. To determine the laser costs, production lead times (i.e. the compilation times of laser processes) are needed since machining and labor can be represented as time amounts. Moreover, once the lead time is known, it is just a simple calculation to determine the laser costs. Furthermore, production lead times are also critical for production planning and customer satisfaction. Correct information about lead times would increase the accuracy and efficiency of production planning, which naturally extend the benefits through the whole supply chain. As a result of correct production planning, manufacturers can give better deadline promises and build customer satisfaction.

Uncertainty quantification is an important research area in machine learning. In contrast with the deterministic view of machine learning, probabilistic models can be described by the phrase "the models that know what they don't know". Likewise in many processes in the business, laser production lead times also have an uncertain nature because of the complexity of the process. Therefore, the lead time modelling constitutes a natural yet not explored application area for probabilistic models. The variability in laser lead times can be explained up to some extent by the features of the part to be produced using pattern recognition techniques. Previous studies in this area focused on point estimates. However, quantifying the uncertainty of estimations is also valuable for practical reasons. The source of the uncertainty that we are interested in can be cast into two:

i. Epistemic uncertainty: uncertainty due to the model

Preprint. Under review.

ii. Aleatoric uncertainty: uncertainty that captures noise in the data.

Our model corpus varied in terms of the ability to capture these uncertainties (see Gal [2016] for an extensive discussion on this topic).

Probabilistic modelling of lead times also have substantial business value for pricing decisions. The prediction intervals can serve as functional decision tools when quoting for a product. For instance, taking a predetermined confidence level, the decision maker can quote for at the lower bound in order to be in front of the competition, or otherwise they can quote at the upper bound to make extra profit. Furthermore, for automatic quoting, higher uncertainty can be used as a signal for human intervention requirement.

In this paper, we present our methodology and results for the data-driven prediction of laser lead times using data from a metal manufacturing firm. This problem has drawn extensive attention in the literature due to its importance. While the nature of the problem is stochastic and probabilistic modelling of the problem has significant benefits, the majority of the literature is populated by point estimate methods. We introduce probabilistic modelling into the domain while exploring various probabilistic machine learning algorithms. Our results indicate that our models beat the company estimation benchmark that uses domain experience and have good calibration with the empirical frequencies.

2 Related Work

Predictive modelling of manufacturing lead times is a novel research stream aligned with the increased data availability and recent developments in artificial intelligence and machine learning. One of the most recent works in cost estimation is Loyer et al. [2016]. Their approach is a case study of a cost estimation problem in jet engine components. They applied various techniques including multiple linear regression and artificial neural networks, support vector regression and gradient boosted trees. They use six main features in their analysis: the part category, two features that describe the part geometry, two for material properties (machinability, cost rate) and the last one is the production volume. Note that their sample size in that study is relatively lower than the size that is generally required for artificial neural networks. Similarly, Lingitz et al. [2018] applied some machine learning algorithms to get point estimates of manufacturing lead times for semiconductor manufacturing.

Mori and Mahalec [2015] conducted a comparison between Bayesian networks, ANN, and SVM to predict probability distributions of production loads and production times for a plate mill process. They first classify production groups according to similar product features then computed the probability distribution for each production group. Öztürk et al. [2006] used a hypothetical manufacturing environment and synthetic data to test their decision tree approach augmented by an attribute selection scheme. Pfeiffer et al. [2016] used random forest for prediction and tested their approach on a synthetic data combined with a simulation based decision support system.

Our work has substantial differences with the prior work. Firstly, our uncertainly quantification is a novel approach. Mori and Mahalec [2015] used Bayesian networks before, however, they first grouped parts then fitted distributions to these groups. Whereas we output a distribution to each sample and employ probabilistic models in more granular sense. Additionally, we propose and compare three state-of-the-art methods that have never been utilized for this problem and apply a post-hoc calibration method[Kuleshov et al., 2018] to increase the accuracy of probabilistic outputs.

3 Data

We exhibit our dataset in this section. We use a production dataset of a metal manufacturer. The dataset consists material, thickness, cutting gas, part dimensions, sheet dimensions, cut dimensions, the number of holes, operation type, and rule-based lead time estimate of the manufacturer as input features. The rule-based lead time estimate is a natural baseline for regression metrics since the model we develop is going to replace that in production. In addition, we use it as an input since it can carry valuable information from the manufacturer's experience. By this way, we integrate domain knowledge into our models, which is known to improve models [Nabi et al., 2020]. For instance, according to the manufacturer bigger parts take more time or more holes delay the compilation time. However, the exact relationship between the features and production time is unknown a priori and it has non-linearity and feature interactions. Moreover, production times have idiosyncratic variation,

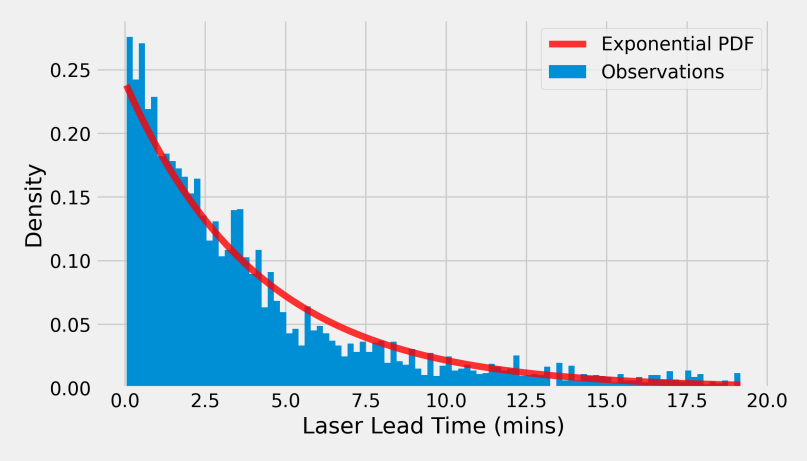

Figure 1: The target variable follows an exponential distribution with rate $\mu = 4.16$

as we explained in Section 1. To extract this relationship and quantify the uncertainty, we propose probabilistic models in this paper.

The output of the model is the realized production duration in the manufacturing facility. After cleaning data errors and samples with missing values, the size of the dataset is 7841. 20% of the dataset is used as the test set and 3-fold cross validation is used for the hyperparameter selection. Unless otherwise stated, all the results given in this paper are from the test set. It can be seen from Figure 1 that the laser lead times follow a typical exponential distribution.

4 Methods

In this section, we present the algorithms that we used for lead time prediction. We develop four methods to have predictive intervals for the output instead of a point estimate. Three of our models are probabilistic models that are fitting a probability distribution as the output. The fourth model uses quantile regression to form predictive intervals.

i. **Probabilistic neural network (P-NN):** We build a fully connected neural network with a probabilistic output layer. The output of this model is Gaussian distribution and the training is done by maximizing the log-likelihood.

ii. **Natural gradient boosting model (NGBoost):** NGBoost uses the natural gradient[Duan et al., 2020] to leverage a distribution as the output of the model in contrast with other tree-based boosting models. We optimized log-likelihood using exponential and Gaussian output distributions.

iii. **Gaussian process regression (GP) model:** Using a usual Gaussian process model is not feasible for large datasets since it has $O(N^3)$ data complexity. One way to deal with this complexity is variational inference [Hensman et al., 2013]. In this setting, the optimization is done by minimizing the evidence lower bound (ELBO).

iv. **Gradient boosted trees augmented by quantile regression (Q-GB):** Although this model does not output a distribution, it can be used to generate predictive intervals via quantile loss. According to our results, it works well in practice and the estimated prediction intervals are well calibrated.

The consistency between predicted and empirical probabilities is essential for probabilistic models [Simchi-Levi et al., 2020]. We report calibration performances of the models in Section 5. We used isotonic regression technique proposed in Kuleshov et al. [2018] to carry out post-hoc calibrations for probabilistic outputs.

	Mean Predictions		Median Predictions	
	R^2	MAPE	R^2	MAPE
Probabilistic Neural Network	0.35	14.52	**0.35**	14.52
Natural Gradient Boosting (Exponential Output)	**0.40**	**11.71**	0.32	**8.07**
Natural Gradient Boosting (Normal Output)	0.31	14.45	0.31	14.45
Gaussian Process	0.33	13.63	0.33	13.63
Gradient Boosting with Quantile Regression	0.39	16.11	0.28	8.94
Status Quo Predictions	0.22	19.61	N/A	N/A

Table 1: Model performance measures. Mean and median predictions differ for Natural Gradient Boosting and Gradient Boosting with Quantile Regression. Status quo predictions come from company estimates based on domain experience.

5 Results

In this section, we compare the performances of the algorithms and the status quo benchmark. We use the coefficient of determination (R^2) and mean absolute percentage error (MAPE) as performance measures and calibration curves to evaluate probability calibrations. Using these metrics, we are able to compare all four algorithms and the status quo benchmark.

It can be seen from Table 1 that all of the models improve the company benchmark. NGBoost with exponential output is better than others in both mean and median predictions. Moreover, median predictions are better off in terms of MAPE and worse off in R^2. We note that the models whose output is normal distribution have the same mean and median. Since the target variable has an exponential distribution, the left skewness of exponential distribution makes an advantage for median predictions for MAPE. This has a reverse effect for R^2 since it normalizes the scale, therefore, predicting means is more meaningful for normalized values.

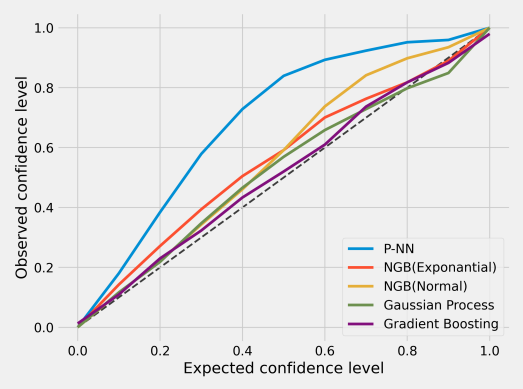

Figure 2: Calibration curves of models. Perfect calibration is represented by the grey dashed line.

Figure 2 shows the alignment between predicted probabilities and empirical counterparts. We give the results without a post-hoc calibration for brevity, post-hoc calibration results can be found in the Appendix. All models have a tendency to have slight underconfidence. Although it is at negligible levels for the other models, one can observe that the P-NN is significantly underconfident. On the other hand, we can conclude that quantile regression augmented Gradient Boosting has the best calibration.

6 Concluding remarks

In this study, we applied probabilistic modelling techniques for the lead time prediction of metal manufacturing. Lead time prediction is essential for several business decisions and probabilistic modelling of this task is almost entirely ignored in practice. We fill this gap with four powerful methods and present our results in this paper. Our results show evident improvement over the company's current rule-based estimation along with well probability calibration.

References

Tony Duan, Avati Anand, Daisy Yi Ding, Khanh K Thai, Sanjay Basu, Andrew Ng, and Alejandro Schuler. Ngboost: Natural gradient boosting for probabilistic prediction. In *International Conference on Machine Learning*, pages 2690–2700. PMLR, 2020.

Jerome Friedman, Trevor Hastie, Robert Tibshirani, et al. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

Yarin Gal. Uncertainty in deep learning. 2016.

James Hensman, Nicolo Fusi, and Neil D Lawrence. Gaussian processes for big data. *arXiv preprint arXiv:1309.6835*, 2013.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. Accurate uncertainties for deep learning using calibrated regression. In *International Conference on Machine Learning*, pages 2796–2804. PMLR, 2018.

Zhong Heng Lim, Umi Kalsom Yusof, and Haziqah Shamsudin. Manufacturing lead time classification using support vector machine. In *International Visual Informatics Conference*, pages 268–278. Springer, 2019.

Lukas Lingitz, Viola Gallina, Fazel Ansari, Dávid Gyulai, András Pfeiffer, Wilfried Sihn, and László Monostori. Lead time prediction using machine learning algorithms: A case study by a semiconductor manufacturer. *Procedia Cirp*, 72:1051–1056, 2018.

Jean-Loup Loyer, Elsa Henriques, Mihail Fontul, and Steve Wiseall. Comparison of machine learning methods applied to the estimation of manufacturing cost of jet engine components. *International Journal of Production Economics*, 178:109–119, 2016.

Andrey Malinin, Liudmila Prokhorenkova, and Aleksei Ustimenko. Uncertainty in gradient boosting via ensembles. *arXiv preprint arXiv:2006.10562*, 2020.

Alexander G de G Matthews, Mark Van Der Wilk, Tom Nickson, Keisuke Fujii, Alexis Boukouvalas, Pablo León-Villagrá, Zoubin Ghahramani, and James Hensman. Gpflow: A gaussian process library using tensorflow. *J. Mach. Learn. Res.*, 18(40):1–6, 2017.

Junichi Mori and Vladimir Mahalec. Planning and scheduling of steel plates production. part i: Estimation of production times via hybrid bayesian networks for large domain of discrete variables. *Computers & Chemical Engineering*, 79:113–134, 2015.

Sareh Nabi, Houssam Nassif, Joseph Hong, Hamed Mamani, and Guido Imbens. Bayesian meta-prior learning using empirical bayes. *arXiv preprint arXiv:2002.01129*, 2020.

Atakan Öztürk, Sinan Kayalıgil, and Nur E Özdemirel. Manufacturing lead time estimation using data mining. *European Journal of Operational Research*, 173(2):683–700, 2006.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

András Pfeiffer, Dávid Gyulai, Botond Kádár, and László Monostori. Manufacturing lead time estimation with the combination of simulation and statistical learning methods. *Procedia CIRP*, 41:75–80, 2016.

Amir Rahimi, Kartik Gupta, Thalaiyasingam Ajanthan, Thomas Mensink, Cristian Sminchisescu, and Richard Hartley. Post-hoc calibration of neural networks. *arXiv preprint arXiv:2006.12807*, 2020.

David Simchi-Levi, Rui Sun, Michelle X Wu, and Ruihao Zhu. Calibrating sales forecast in a pandemic using online non-parametric regression model. *Available at SSRN 3670264*, 2020.

M Welsing, J Maetschke, K Thomas, A Gützlaff, G Schuh, and S Meusert. Combining process mining and machine learning for lead time prediction in high variance processes. In *Congress of the German Academic Association for Production Technology*, pages 528–537. Springer, 2020.

A Post-hoc Calibration Results

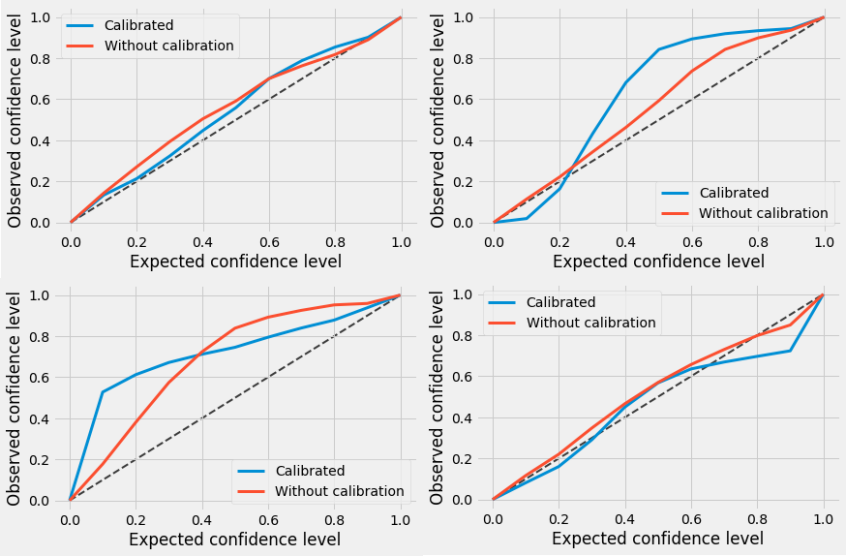

Figure 3: Calibration curves of the raw outputs and calibrated outputs. Top left: NGBoost with Gaussian output. Top right: NGBoost with exponential output. Bottom left: Probabilistic neural network. Bottom right: Gaussian Processes.

While some of the models do not require an ad-hoc calibration, the method we used [Kuleshov et al., 2018] failed to improve the models with bad calibration (e.g. NGBoost with Gaussian output and the probabilistic neural network).

B Model Parameters

Gaussian Processes. We use the model proposed in Hensman et al. [2013] and implemented using GPflow [Matthews et al., 2017]. We use ADAM[Kingma and Ba, 2014] to optimize the evidence lower bound with mini-batch size of 1000.

Probabilistic Neural Network. We use Gaussian distribution in the output layer and ADAM[Kingma and Ba, 2014] to optimize log-likelihood with learning rate 10^{-4}. Our architecture consists of two hidden layers each with 256 nodes and L2 regularization factor of 0.01.

NGBoost. We use default hyperparameters for Gaussian and exponential output distribution.

Quantile regression augmented Gradient Boosting. We use Scikit-learn[Pedregosa et al., 2011] implementation with the following hyperparameters:

Hyperparameter	Value
learning_rate	0.007
max_depth	13
max_features	3
min_samples_leaf	5
min_samples_split	10
n_estimators	200
subsample	0.65