Proceedings of the 2018 Winter Simulation Conference
M. Rabe, A.A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, eds.

COMPARISON OF DATA ANALYTICS APPROACHES USING SIMULATION

Sanjay Jain

George Washington University
Department of Decision Sciences
Funger Hall #415, 2201 G Street NW
Washington, DC 20052, USA

Anantha Narayanan

University of Maryland
Department of Mechanical Engineering
Glenn L. Martin Hall, 4298 Campus Dr.
College Park, MD 20742, USA

Yung-Tsun Tina Lee

National Institute of Standards and Technology
Engineering Laboratory
100 Bureau Drive
Gaithersburg, MD 20899, USA

ABSTRACT

Manufacturers need to quickly estimate cycle times for incoming orders for promising delivery dates. This can be achieved by using data analytics (DA) / machine learning (ML) approaches. Selecting the right DA/ML approach for an application is rather complex. Obtaining sufficient and right type of data for evaluating these approaches is a challenge. Simulation models can support this process by generating synthetic data. Simulation models can also be used to validate DA models by generating new data under varying conditions. This can help in the evaluation of alternative DA approaches across expected range of operational scenarios. This paper reports on use of simulation to select an approach to support the order promising function in manufacturing. Two DA approaches, Neural Networks and Gaussian Process Regression, are evaluated using data generated by a manufacturing simulation model. The applicability of the two approaches is discussed in the context of the selected application.

1 INTRODUCTION

Initiatives for advances in manufacturing, such as Smart Manufacturing, Industry 4.0, etc., call for exploiting the large volumes of data now available through ubiquitous sensors, to improve decision making by using Data Analytics (DA) and Machine Learning (ML) approaches. Though DA and ML are technically different, we use DA to cover both terms in this paper. The selection of the right DA approach is rather complex and requires significant expertise and effort. Manufacturing industry comprises of a wide range of configurations generally grouped into process and discrete production with multiple variations within each. Production of the same product by different companies may be set up using different philosophy. For example, a discrete product may be made using an assembly line setting, a batch production setting, or a job shop depending on the volumes and customization provided. Production systems may be configured differently even within the same production setting. For example, the division of work between sub-assembly lines and main line can vary for assembly lines for the same product. Beyond physical configuration, the operational policies can vary between push and pull, again with multiple further classifications within each. The level of technology employed to collect data and support real time decision making is another aspect with large variety. It is thus difficult to develop one approach that is applicable for manufacturing industry given the complexity due to the variety of product configurations, production system configurations, operation policies, and technology employed.

A number of DA approaches have been developed over the years, and new combinations continue to appear with ongoing research efforts. DA applications have been classified as descriptive, diagnostic, predictive, and prescriptive. DA approaches can fit multiple application classes. For example, Jain et al. (2017) discuss the use of simulation in diagnostic, predictive and prescriptive analysis roles and as a support application for other DA approaches. Recent efforts have focused on data-driven approaches to glean new learning from the data, and thus go beyond the modeling approaches based on known theoretical frameworks of systems. Such approaches include neural networks (NNs), Gaussian process regression (GPR), Bayesian networks, support vector machines (SVM), etc. It is rather complex to identify the approach that will work best for an application. In particular, the expertise required to identify the best approach for an application may not be generally available in manufacturing organizations.

This paper presents a simulation based approach for evaluating two DA approaches, NNs and GPRs, for a potential application in manufacturing. The two approaches were selected for this evaluation based on the reports in literature indicating these approaches generally worked better than other approaches (for example, see Scholz-Reiter et al. (2010)). The next section reviews relevant applications of NNs and GPRs in manufacturing and comparative evaluations of the two approaches in general. Section 3 discusses the use of simulation for generating the data for training and testing the two approaches. Section 4 describes our implementation of NN and GPR for predicting throughput in a small job shop environment, and Section 5 presents the experimental set-up used for this study. The results are presented and discussed in Section 6, and the last section concludes the paper.

2 RELATED WORK

This section identifies applications of NNs and GPR in manufacturing process and system control. Section 4 provides more technical details about NNs and GPR models and how we use them. For theoretical background on NNs and GPR, we refer the reader to Haykin (2004) and Rasmussen (2004). There are a number of applications of both approaches in maintenance area but they are not included due to space constraints. Comparative applications of the two approaches are included that are from outside of manufacturing due to lack of such works in manufacturing context.

2.1 Neural Networks for Manufacturing Applications

A number of applications of NNs have been proposed in the manufacturing domain at the process level with a few examples included here. Li et al. (2015) use a back propagation NN to optimize the cutting parameters in sculpted parts machining. The approach is shown to select process parameters leading to less machining time, less energy consumption, and better surface roughness compared to traditional approach for a test piece. Ding et al. (2016) use a NN for identifying the relationship between process parameters and aluminum bead geometry in arc-welding based additive manufacturing process. The authors use a Taguchi design for efficient data collection for training the NN. The weld settings are then selected based on the NN model. Khorasani and Yazdi (2017) similarly use a NN for identifying the relationship between milling process parameters and surface roughness of the product with a training data set collected using a full factorial design of experiment.

Recent reported applications of NN at levels higher than processes in the manufacturing management hierarchy are harder to find. Scholz-Reiter et al. (2010) cite an example of use of NN for dispatching rule selection from 2006 and a few other similar papers from the last millennium. Recently, Miller et al. (2014) train a NN for estimating the assembly times of vehicle sub-assemblies using geometric part information as inputs. The authors evaluate more than a hundred architectures of NN and use the top five to generate probability density plots of estimated assembly times. They achieved moderate success using this approach with predicted values falling within a +/-15% range of the associated target value. Rehman et al. (2016) develop a NN for estimating organization performance measures based on green manufacturing data on design initiatives, standards adaptation, purchasing, disposal, etc. for companies in the Indian industry. The developed NN is applied to a steel company and was found to be useful in verifying company's initiatives and for providing further guidance.

2.2 Gaussian Process Regression for Manufacturing Applications

GPR applications in manufacturing focus at process level similar to NN applications and again a couple representative examples are provided here. Bhinge et al. (2014) use GPR methods to predict the energy used to machine a part based on machine monitoring data. They process the raw machine monitoring data into derived data that is then further process through a cutting simulator to generate operation parameters. GPR is applied with operation parameters as inputs and power consumption as output to develop an energy prediction model. Liu et al (2015) utilize an ensemble GPR approach to predict a measure of viscosity of the product of an industrial rubber mixing process based on parameters and recipes. The authors also include the ability of GPR based approach to generate uncertainty measures for the predictions as an advantage over other available approaches for the purpose.

2.3 Comparison of NNs and GPR

There are a handful of efforts available in literature that compare NNs and GPR in application settings. Goebel et al. (2008) compared three DA approaches, NNs, GPR, and Relevance Vector Machines (RVM), for prediction of remaining useful life based on damage for a rotating aerospace equipment on test stand. The set up was limited to a small damage data set given the expensive set up. The authors found NNs performance was dependent on choice of data and the design of its architecture. Performance of GPR was similarly dependent of the choice of the covariance function used but it offered the advantage of providing confidence bounds around mean predictions. GPR was identified as scaling typically as $O(n^3)$ with the number of training data points and thus would demand high computation power and time. The authors also indicated the need for domain specific measures to compare performance beyond accuracy.

Scholz-Reiter et al. (2010) compare GPR with NN and other approaches for dynamically selecting dispatching rules in production scheduling. They train the approaches using utilization and due date tightness as inputs and tardiness as outputs. They found that GPR generally outperformed other approaches including NN. Interestingly, their results differ from all other comparative studies as they indicate that GPR is outperformed by NN for smaller learning data sets. They identify some issue with hyperparameter setting of GPR for this anomaly and propose to further investigate it.

Ahmed et al. (2010) compared eight machine learning approaches including multiple variations of NNs and GPR for business-type time series. They used monthly time series of the benchmark M3 competition data (IIF 2017) for the study. The multilayer perceptron (MLP), referred to as NN in this paper, and GPR were found to be the top performing approaches. The performance was found dependent on the preprocessing of the data with MLP performing best for the commonly used lagged-value and moving average techniques. GPR was found to have second best performance with the two preprocessing techniques but was found to be robust for the difference technique also.

Chen et al. (2014) compare GPR with NN for wind power forecasting and find that GPR provided 9-14% improvement over NN for two large data sets. They note the advantage of GPR improved to 17% for the third data set with limited amount of training data.

Kamath et al. (2018) compare the two for representing potential energy surfaces in molecules with more than 3 atoms. They found the GPR outperformed NNs, that is, achieved higher accuracy with smaller data sets used for fitting. However, they point out that GPR is slower compared to NNs and needs training data points that are sufficiently far apart. NNs can work with overlapping data points, but can suffer from overfitting. NNs were considered easier to build and recommended when the cost of data collection is low. The authors suggested that both approaches can gain from optimized sampling of the training data points.

The comparative studies report above do draw some common conclusions across the widely different application areas. GPR appears to have an accuracy advantage over NNs when training data size is limited. Also, GPR is identified as computationally expensive approach compared to NNs. Only one comparative study by Scholz-Reiter et al. (2010) mentioned above used a manufacturing application scenario and interestingly it differed in its assessment of GPR performance for smaller data sets from all the other studies cited above. The different results suggest that the performance of the DA approaches may depend on the application. With the push towards smart manufacturing there is a need to evaluate the potential manufacturing system level planning and control applications of DA approaches.

3 DATA GENERATION USING SIMULATION

DA approaches such as NN and GPR require data for training the respective models that can be used in a predictive capacity. While the models can provide predicted outcomes for input data for situations that were out of the coverage range in the training data, such predictions may have low accuracy. It is best to have such predictive models be trained across the range of situations that they will then be required to assess. For manufacturing applications, it can take a long time and lot of effort to collect such data from the real manufacturing system. Simulation models of manufacturing allow the option of creating a range of situations in the virtual representation and collecting the necessary data for training the predictive models.

Also, it is generally difficult to access data from manufacturing organizations for researchers. This paper, hence, uses a virtual factory prototype, essentially a multi-resolution simulation model of manufacturing, for generating the data. This section provide brief information on the use case, the virtual factory prototype and the set-up for data generation. The reader is referred to Jain et al. (2017) for more details.

3.1 Use Case

The use case for this work is based on the order promising scenario for a small job shop that produces three part types that are essentially the same part but produced using different materials: aluminum, titanium, and steel. The small job shop consists of a turning cell with 4 machines and a milling cell with 2 machines. All parts go through two operations, the first one being in the turning cell and the second one in the milling cell. Each arriving batch goes from a raw material stock to a finished good area after being successively processed in the two cells. Each batch is composed of ten parts. Each type of part requires different value ranges of machine parameters leading to different processing times. Orders are received at the source and specify the part type and the number of parts to be processed in the shop. In the base scenario, the order frequency follows a normal distribution with a mean of 60 minutes. The order frequency is varied to mimic different load levels in the shop.

Customers place orders for defined quantities of one of the three part types. The planner performing the order promising function has to provide an expected shipment date for the order. It is complex to estimate the shipment date as it can vary based on several factors including the ordered quantity, current load on the shop, and machine failure characteristics. Simulation models that use the current situation on the shop floor as the starting condition can be used for estimating the shipment date, but it can take time to generate such estimates and may require more expertise than a typical planner may have. Also, the company wouldn't want to have customers wait for several minutes for finding out the shipment date. The planner needs to be supported by an application that generates shipment date estimates for proposed orders within seconds. The two DA approaches are being evaluated to meet the need for estimating the cycle time for an incoming order based on the current conditions on the shop floor.

3.2 Virtual Factory Prototype

The virtual factory prototype is an initial effort towards implementing the virtual factory concept described in Jain et al. (2017). The prototype allows development of integrated multi-resolution model that can be executed at the selected resolution. The levels include a physics-based process representation, an agent simulation based machine representation, and a discrete event simulation based cell or factory representation. It includes the capability of reading in data files describing a small manufacturing system, generating a discrete event simulation model using a library of machine level models, executing the model together with a basic animation, and generating selected output graphs.

3.3 Data Generation Set-up

The data generation set-up is shown in Figure 1. The virtual factory model has been set-up with standards based interfaces. The input files include manufacturing configuration data in Core Manufacturing Simulation Data (CMSD) standard format (SISO 2012), machine instructions in STEP-NC format (ISO 2007), and some custom formats for machine data. The virtual factory model is generated mostly automatically using the input data files together with the library. Following the execution of the model, in addition to the standard results generated by the simulation software, output files are generated using standard formats as shown in the figure. The factory data files in Business To Manufacturing Markup Language (B2MML; MESA 2013) are used for training the DA approaches for this study.

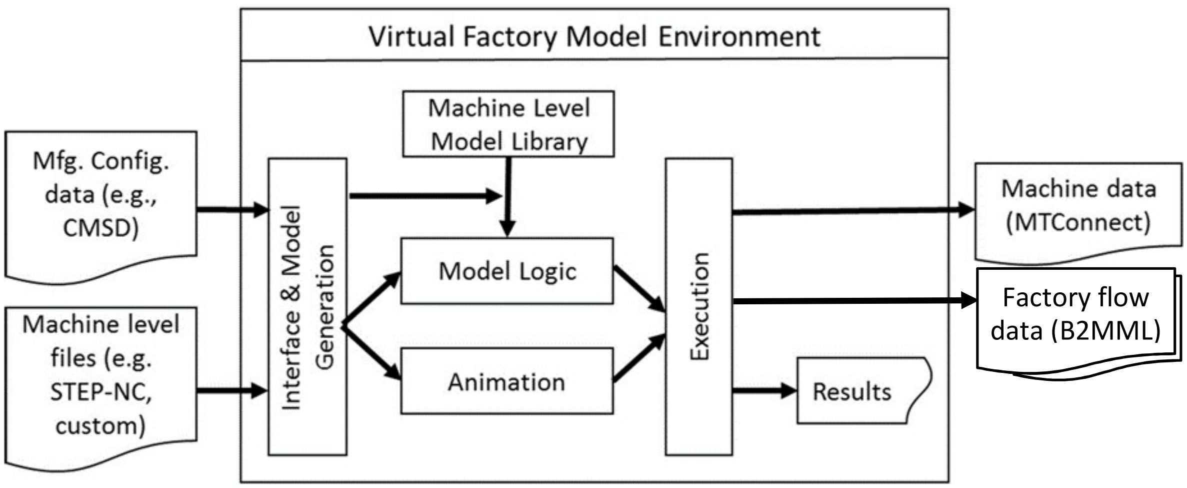

Figure 1: Data generation using virtual factory prototype.

4 METHOD IMPLEMENTATIONS

4.1 Neural Network Implementation

The NN implementation was described in Jain et al. (2017) and is briefly summarized here. Artificial Neural Networks (ANNs) are a type of predictive model that consist of an input layer, a number of hidden layers, and an output layer. Each layer consists of one or more neurons, and the neurons between layers are connected by weighted edges. The neurons of the input layer correspond to the known system variables, and the neuron in the output layer corresponds to the metric to be predicted. The ANN consists of a series of linear and non-linear transformations that turn the input variables to the output metric. During training, the weights on the edges of the NN are adjusted to produce the correct output value for each combination of input values from a training data set.

In this work, the output value to be predicted is the duration expected to complete an incoming job. The input values are the number of parts in the job, the material type, and the current load on the system. The current load on the system is modeled by a triplet (n_A, n_S, n_T), which denotes the parts of material type aluminum, steel, and titanium currently being processed in the system. The NN had one hidden layer with ten neurons. We generated a large data set from simulations. The data set was split into two parts, one for training the NN, and the other for validating the trained NN model. The details of the NN implementation and the results of the validation are described in detail in Jain et al. (2017).

4.2 Gaussian Process Regression Implementation

Gaussian process regression (GPR) is a probabilistic method of interpolation to determine a target value from given inputs. Instead of computing a single polynomial with a fixed number of parameters to fit the training data, GPR determines a distribution of random functions that best fit the data. The distribution of random functions that fits the data (called the posterior) is determined from a prior distribution of random functions, which is defined by a covariance and mean. GPR uses a Gaussian distribution for the priors.

The goal of GPR is to determine an unknown target function *f(x)* from the prior distribution and some known data points. To define the prior distribution, we use a kernel function that approximates the covariance, which is a measure of the geometrical distance of closely located input points and their corresponding function values. The chosen kernel function determines the geometrical shape of the target function, and depends on the scenario and data we are interested in. There are many choices for the kernel function. In this study, we chose the radial basis function (RBF), also called the squared exponential, which is a commonly used in many situations.

The squared exponential covariance function is defined as below:

$$K(x, x') = \sigma^2 exp\left[-\frac{1}{2}\left(\frac{x - x'}{l}\right)^2\right] \qquad (1)$$

where, $K(x, x')$ represents the covariance function for the pair of inputs x and x', and σ and l are the hyperparameters that represent the amplitude and length scale, respectively. The amplitude signifies the overall magnitude of the covariance value, and the length scale indicates the relevance of the input features to the response y. In simple terms, adjusting σ changes the overall magnitude of the covariance, and adjusting l changes the smoothness of the target function curve. These hyper parameters must be tuned appropriately to obtain a good target function that matches the data.

5 EXPERIMENTAL SET-UP

The data generation using the virtual factory prototype is described in Jain et al. (2017), and summarized briefly in Section 5.1. Section 5.2 describes the data generated for training and validation for this study.

5.1 Data Generated using the Virtual Factory Prototype

The virtual factory prototype was used to mostly auto-generate a model of the small job shop described in Sections 3.2 and 3.3. The execution was set up to run primarily at the cell level of detail using the discrete event simulation model in the hierarchy of models generated in the prototype for the scenario. Corresponding to the execution at the cell level of detail, the model was set up to generate the factory flow data files in B2MML format (MESA 2013) while the generation of machine event data stream was disabled. In the simulated scenario, raw material is processed through a turning cell, and then a milling cell, and then ends in the finish goods area. The simulation of failure and repair time of machines is included. The time to failure follows an exponential distribution with a mean time between failures (MTBF) defined in the virtual factory model. The machine model remains in the failure state for a sampled value of repair time. The repair time follows an exponential distribution with a mean time to repair (MTTR) also defined in the virtual factory model.

Two different B2MML files were generated. First file recorded the total number of parts produced for each part type and the total duration to produce those parts. The second file recorded the following information for each order completed: ID, start and end times, part type and number of parts ordered, and the load of the factory at the time the order was released captured as number of parts of each of the three part types in process in the job shop at the time. The B2MML files are converted to comma-separated value (CSV) format to facilitate further processing by the DA approaches.

6 RESULTS AND DISCUSSIONS

In this section, we discuss the results of the tests. First we look at the overall prediction by the GPR model, and compare it with the performance of the NN. Figure 2 shows the graph of the predictions for the test data set. The predicted values from the GPR model are represented by a blue line with diamond markers, the predictions of the NN model are represented by a green line with '+' markers, and the corresponding test ground truth values are represented by an orange line with dot markers. The gray area represents the confidence bounds of the GPR prediction, up to one standard deviation above and below the mean. Note that the *x*-axis is simply a sequence of incoming orders – and does not show the other parameters associated with that order. Each point on the *x*-axis corresponds to a point in the test data set that includes the following values: the number of parts in that order, the material type for that order, and the current load on the factory denoted as a triplet (n_A, n_S, n_T). It is expected that in a realistic system, the parameter representing load on the system will be a continuously monitored variable of the system, rather than a part of the order specification. For the purpose of our study, we denote this parameter explicitly as an input parameter.

In the case of the GPR model, the root mean squared error (RMSE) was 1626 seconds, and the mean absolute error (MAE) was 1241 seconds for the test data set. The hyperparameters of the model were empirically chosen. The mean duration in the test data set was 61426 seconds. This represents an error of about 2% on average in the predicted order-completion duration. The NN model had a poorer performance, with an RMSE of 4013 and MAE of 2414. As can be seen in Figure 2, the NN model performance suffers in the range of orders between Order Number 200 and 500, which corresponds to a high load on the system in the test set. The GPR performs better in this region, and not only produces more accurate average predictions, but also provides confidence bounds that reflect the confidence in the prediction, which can

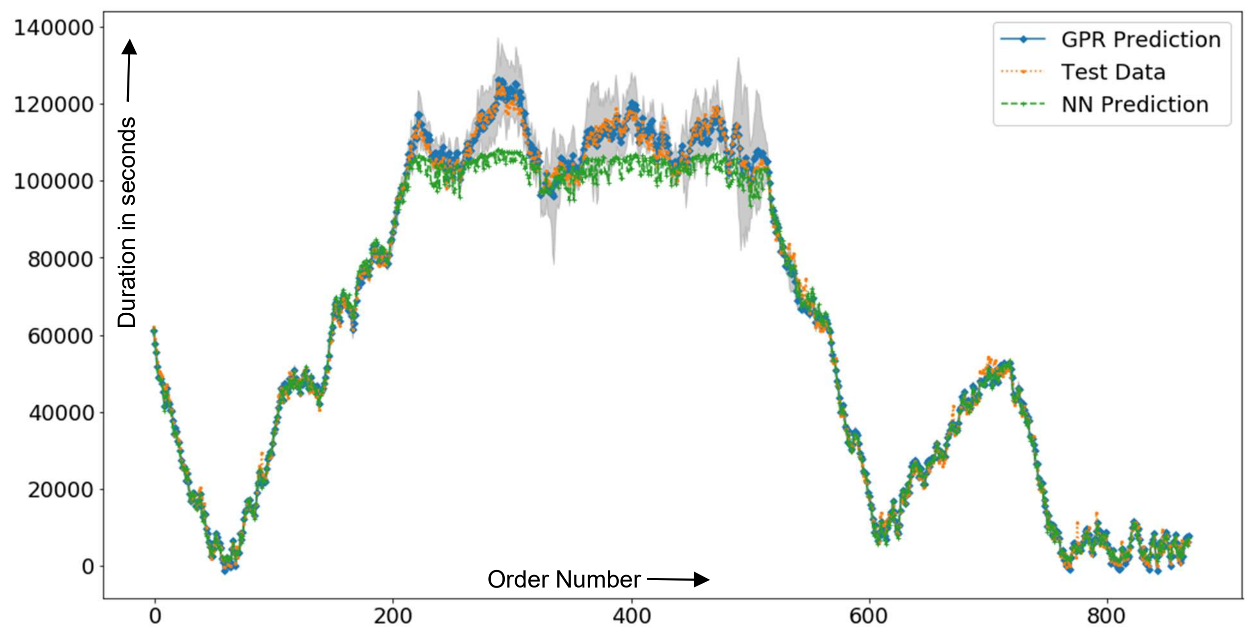

Figure 2: Order-completion duration predictions for test data.

help decision making.

It may be possible to tune the models to further reduce the errors. From the test results, it can be observed that the predictions of both models are less accurate in the range of orders corresponding to the factory being under heavy load in the simulation, as seen by the orders that require a long duration to complete from the moment of order arrival, due to the number of jobs already running on the machines and waiting in the queues. For clarity and comparison, Figures 3 and 4 show expanded views of different portions of the same test data in Figure 2. Notice the difference in order duration on the *y*-axis of the two figures. For the high duration orders in Figure 4, it can be seen that the predictions are less accurate. Some additional model tuning may be performed to address this portion, and is left as future work.

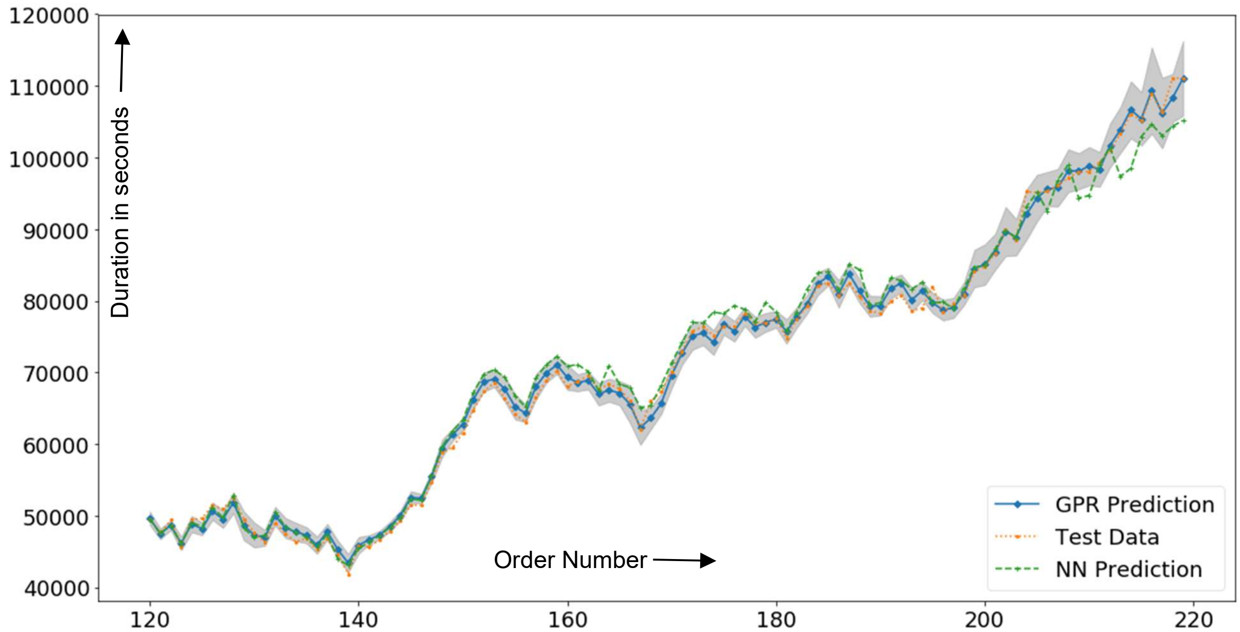

Figure 3: Order-completion duration prediction for test data at low load levels.

Generating data from simulations allows us to build and test these predictive models, and pay close attention to different system conditions, such as light load versus heavy load. Such comparisons are harder to make if we relied only on real factory data.

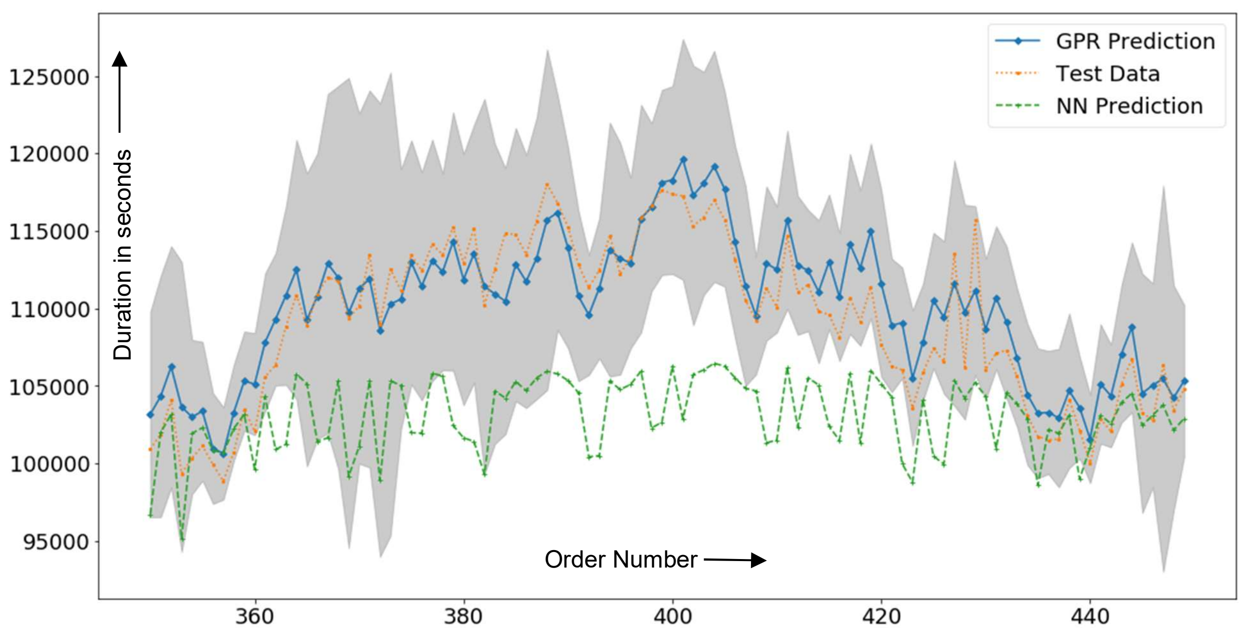

Figure 4: Order-completion duration prediction for test data at high load levels.

Our training data in the above example consisted of 800 training points from simulated data. As an experiment, we performed the training on a highly reduced training set of only 40 points (choosing every 20^{th} in the original training set). Surprisingly, this reduced training set produced good predictions on the same data set, with only a slightly larger error. The GPR model had RMSE of 2938 seconds, while the NN model had an RMSE of 8670 seconds. The GPR model performed significantly better on the reduced training set. The error in the predicted duration is still quite small, compared to the average order-completion duration of 61426 seconds. Figure 5 shows the reduced training set. The important point to note is that the reduced training set still has good coverage over the range of duration values observed. The predictions for corresponding models are shown in Figure 6. In real factory situations, getting quality training data may often prove difficult. Simulated data such as this can be used to quickly evaluate the performance of different models, and make an appropriate choice of model to train on the limited factory data. The performance of the models has been summarized in Table 1.

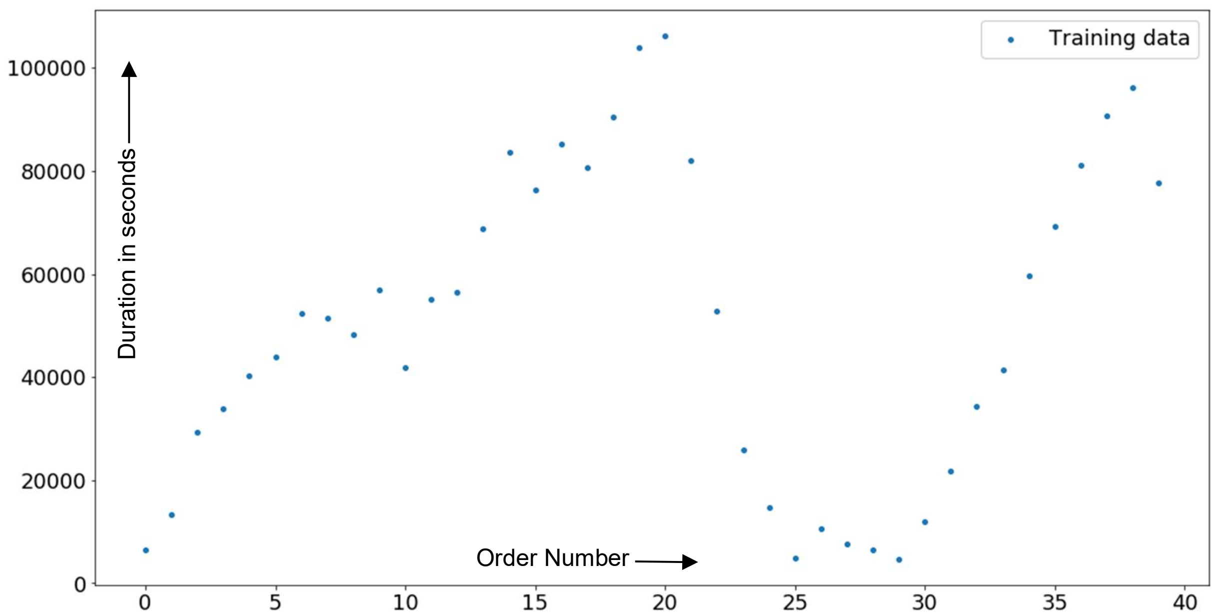

Figure 5: Reduced training data set.

The training time for GPR models increases as the data sets become large, and can become computationally expensive over large data sets. In our tests, the training times for the GPR model using the full training data and the limited training data were 75 milliseconds and 3 milliseconds, respectively. While the individual model training times may be improved by optimizing the code and improving system hardware, the relative times are indicative of the time savings when the model can be trained with less data. In our study, GPR proved to be a good model for training over smaller data sets, as long as the data sets had a good coverage over the expected operational range of the system parameters. In real factory scenarios, it may be difficult to quickly build a data set that has this kind of coverage. Our study using simulated data shows that GPR model performs very well with limited data. The simulations can be a guide to obtain good factory data over the required range of values, and train a good predictive model with limited data.

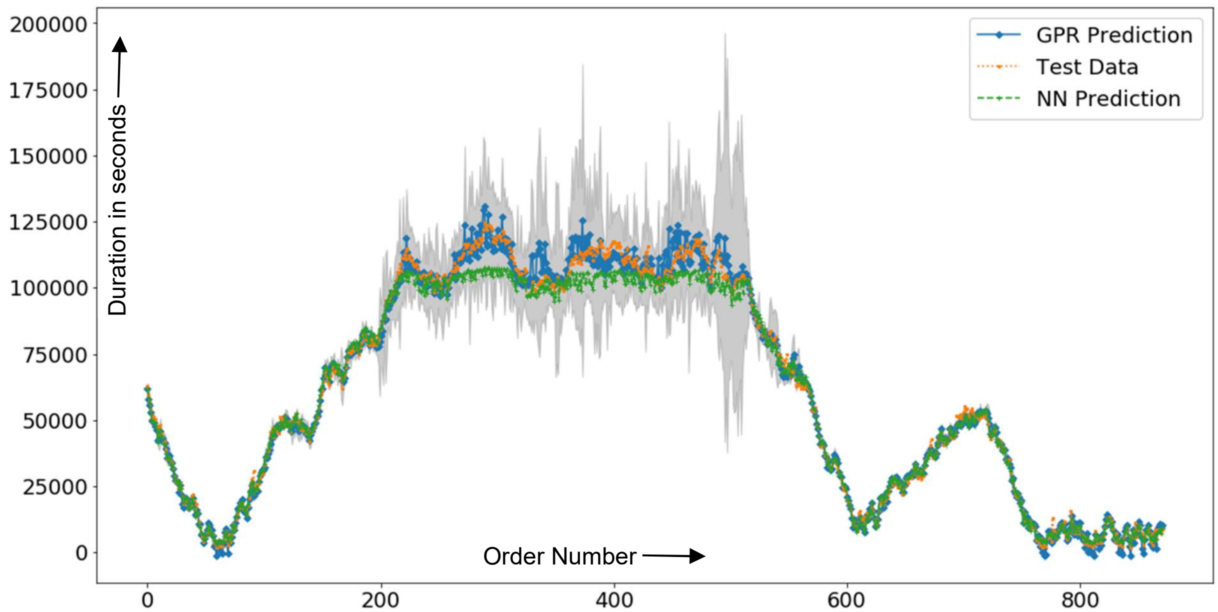

Figure 6: Predictions on reduced training data set.

Table 1: Performance of machine learning models.

Model	Data Set	RMSE	MAE
GPR	Full	1626	1241
NN	Full	4013	2414
GPR	Limited	2938	2008
NN	Limited	8670	6353

7 CONCLUSION

In this paper, we studied GPR as a machine learning model to predict throughput for a small job shop. We compared the GPR model with an NN model for the same purpose. A simulation of the job shop was used to generate data under varying load conditions, and the generated data was used to train and validate the machine learning models. Being able to generate synthetic data from simulation models allows us to build, test, and compare different machine learning models, which would be difficult to do if access to real factory data is limited. Our results showed that the GPR model performed better than the NN model, especially when the factory is operating under the high load condition. The GPR model also performed well when trained using limited data, while the NN model predictions were less accurate when trained with limited data. Testing these models using the simulation allows us to choose a machine learning model based on data availability and prediction accuracy. This allows manufacturers to build a data analytics and decision guidance system when real factory data is limited or not yet available. The GPR model also provides confidence bounds for its prediction which can help decision making.

Future directions under consideration include testing the DA and ML approaches for models of larger manufacturing systems, enhancing the virtual factory prototype for the purpose, generating a benchmark data set for comparing DA and ML approaches, and testing additional DA and ML approaches.

DISCLAIMER

No approval or endorsement of any commercial product by the National Institute of Standards and Technology (NIST) is intended or implied. Certain commercial software systems are identified in this paper to facilitate understanding. Such identification does not imply that these software systems are necessarily the best available for the purpose.

ACKNOWLEDGMENTS

The work of Sanjay Jain and Anantha Narayanan on this effort was supported by National Institute of Standards and Technology Cooperative Agreement Nos. 70NANB18H010 and 70NANB14H250, respectively.

REFERENCES

Ahmed, N. K., A. F. Atiya, N. E. Gayar, and H. El-Shishiny. 2010. "An Empirical Comparison of Machine Learning Models for Time Series Forecasting." *Econometric Reviews* 29(5-6):594-621.

Bhinge, R., N. Biswas, D. Dornfeld, J. Park, K. H. Law, M. Helu, and S. Rachuri. 2014. "An Intelligent Machine Monitoring System for Energy Prediction Using a Gaussian Process Regression." In *Proceedings of the 2014 IEEE International Conference on Big Data*, 978-986. Piscataway, NJ: IEEE.

Ding, D., C. Shen, Z. Pan, D. Cuiuri, H. Li, N. Larkin, and S. van Duin. "Towards an Automated Robotic Arc-Welding-Based Additive Manufacturing System from Cad to Finished Part." *Computer-Aided Design* 73(C):66-75.

Goebel, K., B. Saha, and A. Saxena. 2008. "A Comparison of Three Data-Driven Techniques for Prognostics." In *Failure Prevention for System Availability: Proceedings of 62nd Meeting of the Society for Machinery Failure Prevention Technology*, 119-131. Dayton, OH : Society for Machinery Failure Prevention Technology (MFPT).

Haykin, S. 2004. "NEURAL NETWORKS: A Comprehensive Foundation", Prentice Hall.

IIF. 2017. M3 Competition. International Institute of Forecasters. https://forecasters.org/resources/time-series-data/m3-competition/, accessed August 4th, 2018.

ISO. 2007. *ISO 10303-238:2007. Industrial Automation Systems And Integration.* Geneva, Switzerland: International Standards Organization.

Jain, S., G. Shao, and S.-J. Shin. 2017. "Manufacturing Data Analytics Using a Virtual Factory Representation". *International Journal of Production Research* 55(18):5450-5464.

Jain, S., D. Lechevalier, and A. Narayanan. 2017. "Towards Smart Manufacturing with Virtual Factory and Data Analytics." In *Proceedings of the 2017 Winter Simulation Conference*, edited by W. K. V. Chan et al., 3018-3029. Piscataway, NJ: IEEE.

Kamath, A., R. A. Vargas-Hernández, R. V. Krems, T. Carrington Jr, and S. Manzhos. 2018. "Neural Networks vs Gaussian Process Regression for Representing Potential Energy Surfaces: A Comparative Study of Fit Quality and Vibrational Spectrum Accuracy." *The Journal of Chemical Physics* 148(24): 241702.

Khorasani, A., and M. R. S. Yazdi. 2017. "Development of a Dynamic Surface Roughness Monitoring System Based on Artificial Neural Networks (ANN) in Milling Operation." *The International Journal of Advanced Manufacturing Technology* 93(1-4): 141-151.

Li, L., F. Liu, B. Chen, and C. B. Li. "Multi-Objective Optimization of Cutting Parameters in Sculptured Parts Machining Based on Neural Network." *Journal of Intelligent Manufacturing* 26(5): 891-898.

Liu, Y., and Z. Gao. 2015. "Real - Time Property Prediction for an Industrial Rubber - Mixing Process with Probabilistic Ensemble Gaussian Process Regression Models." *Journal of Applied Polymer Science* 132(6):41432.

MESA 2013. *Business To Manufacturing Markup Language Release Notes Version 6.0 – March 2013.* MESA International, Chandler AZ.

Miller, M. G., J. D. Summers, J. L. Mathieson, and G. M. Mocko. 2014. "Manufacturing Assembly Time Estimation Using Structural Complexity Metric Trained Artificial Neural Networks." *Journal of Computing and Information Science in Engineering* 14(1): 011005.

Rasmussen, C. E. 2004. "Gaussian Processes in Machine Learning." *Advanced Lectures on Machine Learning*. Springer, Berlin, Heidelberg, 63-71.

Rehman, M. A., D. Seth, and R. L. Shrivastava. "Impact of Green Manufacturing Practices on Organisational Performance in Indian Context: An Empirical Study." *Journal of Cleaner Production* 137 (2016): 427-448.

Scholz-Reiter, B., J. Heger, and T. Hildebrandt. 2010. "Gaussian Processes for Dispatching Rule Selection in Production Scheduling: Comparison of Learning Techniques." In *Proceedings of 2010 IEEE International Conference on Data Mining Workshops (ICDMW),* 631-638. Piscataway, NJ: IEEE.

SISO 2012. *SISO-STD-008-01-2012: Standard for Core Manufacturing Simulation Data – XML Representation.* Simulation Interoperability Standards Organization, Orlando, FL.

AUTHOR BIOGRAPHIES

SANJAY JAIN is an Associate Industry Professor in the Department of Decision Sciences, School of Business at the George Washington University. Before moving to academia, he accumulated over a dozen years of industrial R&D and consulting experience working at Accenture in Reston, VA, USA, Singapore Institute of Manufacturing Technology, Singapore and General Motors North American Operations Technical Center in Warren, MI, USA. His research interests are in application of modeling and simulation of complex scenarios including smart manufacturing systems and project management. His email address is jain@email.gwu.edu.

ANANTHA NARAYANAN is a Research Associate at the University of Maryland. He completed his PhD at Vanderbilt University, Nashville, TN, USA in 2008. His research interests are in data analytics and model based systems engineering. He is currently working as a guest associate in NIST’s Systems Integration Division of the Engineering Laboratory. His email address is anantha@umd.edu.

Y. TINA LEE is a computer scientist in the Systems Integration Division of the Engineering Laboratory at NIST. Currently, she leads the Data Analytics for Smart Manufacturing Systems Project of NIST’s Smart Manufacturing Systems Design and Analysis Program. She is the co-editor of the Simulation Interoperability Standards Organization (SISO) Standards of Core Manufacturing Simulation Data (CMSD). Her email address is yung-tsun.lee@nist.gov.