Imitation Learning from Operator Experiences for a Real time CNC Machine Controller

Hoa Thi Nguyen[1, 2], Øystein Haugen[1], and Roland Olsson[2]

[1] *Faculty of Computer Sciences, Østfold University College,* Halden, Norway
[2] *Department of Informatics, University of Oslo,* Oslo, Norway
Email: {hoa.nguyen,oystein.haugen,roland.olsson}@hiof.no

***Abstract*—Controlling complex industrial systems can be a challenging task as it requires extensive knowledge and skills that are usually acquired through years of experience. This makes it difficult to program such expertise into machine algorithms. In this paper, we present a use case that demonstrates how we built control algorithms for a CNC machine using historical logging of observations from experts.**

With the advent of digital technologies, machining parts are now controlled by computer programs that offer high precision and speed. However, unforeseen scenarios can still arise, which demand operators' attention and intervention, even with finely crafted machine programs. For our experiment, we collected data from a 5-axis Mazak Integrex i500-series CNC machine over a month manufacturing multiple instances of the same part. We collected observational states, which are sensor data that match the information operators receive and output engagement feed rates following the operator's trajectories. Using behavioral cloning, we built an initial control policy from this data, testing three families of machine learning models: regression models, ensemble methods, and deep neural networks. The results showed that ensemble methods outperformed the baseline model significantly, proving that they have learned useful control patterns. The policies also demonstrated that ML models could eliminate noisy behaviors from operators' actions. We believe that with interactive demonstrations in the future, these models have the potential to fully mature.

Overall, our study demonstrates the feasibility of building control algorithms for complex industrial systems using historical expert demonstrations and machine learning techniques.

I. Introduction

Industry 4.0 is an integral component of the Fourth Industrial Revolution, with a focus on intelligent manufacturing that improves the relationship between digital technologies and production systems [1]. The concept originated from Germany and is expected to revolutionize manufacturing by increasing productivity and reducing labor and material costs. Industry 4.0 has delivered significant successes, particularly in the semiconductor industry and the oil and gas sector [2]. However, Dalenogare et al. [3] pointed out that while IoT technologies have accelerated product innovation, such as in additive manufacturing, Big Data and data analysis are still lagging in generating benefits. This is primarily due to the lack of knowledge of data analysis in most industrial sectors.

Machining parts is a subtractive manufacturing process of cutting raw materials into desired shapes and sizes. It is also a beneficiary of Industry 4.0. Blueprints of machined parts are created in Computer Aided-Design and Manufacturing (CAD/CAM) software and generated into Computer Numerical Control (CNC) programs that guide machine tools in producing machine parts with high-precision and high-speed. Many manufacturing sectors, such as the aerospace and medical industries, require machining parts rather than mass production because of the need for high-quality components [4]. Computer aid allows CNC machining of more complex customized designs than other mass production methods. The procedure of CNC machining is illustrated in Figure 1. Engineers or machinists create the design of machine parts from users' requirements. The generated CNC programs are tested thoroughly in simulation to optimize the toolpath and avoid any possible conflicts. Operators are then responsible for transferring the CNC programs to the CNC machine, loading raw materials; configuring the machine; maintaining and inspecting the production. Even though a CNC machine is a well-controlled environment, no raw materials are identical, nor do cutting tools stay the same throughout the process. These uncertainty factors are the main reasons why operators are still needed to monitor machining processes.

Research from Deloitte and The Manufacturing Institute projects that the US will experience a severe shortage of trained machine operators in 2025 [5]. This results from workers retiring due to old age and the cost of training replacement candidates. The knowledge of how to control industrial machines' cutting speed and feed rate is mainly obtained through experience. When workers are asked to describe how they intervene during machining processes, they often say it is due to a feeling that it is appropriate for the task [6]. Training a CNC machinist takes four to five years of technical education and on-site training [7]. While it is challenging for workers to give specific reasoning for their actions that can be coded into machine programs, machine learning (ML) can recognize patterns in how expert machinists operate their machinery from historical logging data.

This paper proposes using imitation learning to learn policies that represent operator experiences in controlling engagement linear feed rate and engagement spindle velocity speed rate by training them on historical data of the operator's actions. The rest of the paper is structured as follows. Section II briefs on the background of our project. Section III proposes using imitation learning for CNC control. Section IV details

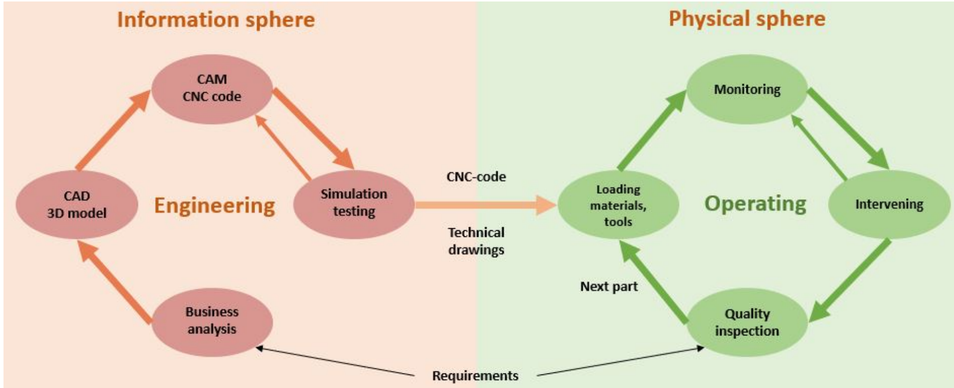


Fig. 1. Production procedure of a CNC machine

the data description and the preparation steps to create trainable datasets. Section V explores different ML techniques, and Section VI discusses the results of our study. Lastly, Section VII concludes our paper.

II. Background

As shown in Fig. 1, the information sphere for a production procedure involves designing blueprints and toolpath for machine parts. Research in this sphere addresses the challenges of producing the optimal toolpath in terms of energy consumption, speed, and accuracy ([8], [9], [10], [11]). Research in the physical sphere or the shop floor level mainly addresses machine monitoring which includes tool condition monitoring, tool wear monitoring, and quality control ([12], [13], [14]). Although the use of sensors is optional in the information sphere [11], sensors are essential for studies in this sphere as they deal with real-life properties that may not be presented in the information sphere, such as vibration and temperature. The value of sensors is, however, heavily dependent on the type of sensors and the location where they are installed [15]. Dealing with real-life properties also means the parameter space is much bigger. Consequently, although the digital twin is a promising technology for machine monitoring, its development is still immature as it struggles with creating a high-fidelity virtual representation of the CNC machine [16]. Thus there is an increasing trend of ML techniques combined with Big Data to tackle problems of machine monitoring [15].

Our industrial partner is Mekanisk Service Halden located in Norway. They are an industrial manufacturing business that specializes in machining parts using several multi-tasking Mazak CNC machines. These state-of-the-art machines are equipped with built-in sensors that generate real-time data, which the company is interested in exploring for potential business opportunities. Along with machine condition monitoring, they are also keen on optimizing human resources. Therefore, they intend to develop machine algorithms that can enhance operators' productivity by enabling them to monitor multiple CNC machines simultaneously.

Using real-time data for operational purposes has been investigated by some studies before. Moreira et al. [14] created a real-time monitoring and controlling system comprising of a surface roughness prediction model and a neuro-fuzzy inference system. Their experiment showed that their system could achieve better surface quality than the operators could. Sakarinto et al. [17] proposed a decision support system for sharing knowledge and expertise between operators. The core of their expert system lies within the knowledge base which is manually crafted. Both approaches required expert knowledge of CNC machinery and did not effectively utilize big data.

III. Imitation Learning For Industrial Machine

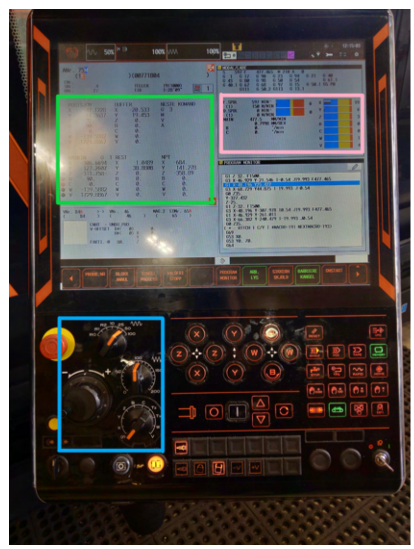


Fig. 2. Mazak Integrex i500 control interface. The green box shows information about the cutter's positions. The pink box shows information about different loads on the cutter. Inside the blue box are the knobs for controlling engagement rates.

Figure 2 shows the interface the operators use to manage the Mazak machine. The blue box at the bottom left encapsulates three knobs that can be used to control the engagement rates (%) for different feed rates. At 100%, the machine runs at the full speed dictated by the CNC program. There are two important configurations that operators need to control to ensure safety:

- Engagement rate of linear feed rate to override the linear feed rates of x, y, z (*Fovr*).
- Engagement rate of spindle velocity to override the rotary velocity of the milling spindle and the turning spindle (*Sovr*).

There are several reasons why the operators need to override them. One reason is surface roughness. If it is too bumpy, the chuck's grip might not be sturdy enough to hold the workpiece when the machine performs high-speed turning. In the worst case, the workpiece will fly out and damage itself and the machine. To determine the engagement rate for spindle velocity, the operators look at different loads on the cutter's motors shown in the pink box in Figure 2. Another reason for adjusting engagement rates is tool wear. In this case, the speed of the machine is adjusted to ensure the surface quality of the workpiece. The operators reported that they used the load on the cutter along the Z-axis to estimate the condition of the cutter. In addition, the operators also base their decision on the sound that the machine makes. However, this information is currently not in our dataset since the Mazak machine does not have built-in vibration sensors.

Operators can demonstrate how to control complex dynamic systems such as a CNC machine, but they cannot fully articulate the specific skills involved. In the context of Industry 4.0, there is abundant logging of observations from operators making imitation learning best suited for this purpose. Behavioral cloning is the simplest imitation learning strategy directly mapping between states and experts' trajectories [18] $f : X_t \to y_t$. The observational states X_t are hypothesized as the same information that operators have available on the control interface during the machining process (Fig. 2). A time delay g is introduced between X_t and y_t

1) Assuming that the actions are operators' response to some previous state due to recognition and physical manipulation of the controls [19].

$$y_t = f(X_{(t-g)})$$

2) Our ML models aim to be a decision support system for operators or a training tool for novice operators. The models should be able to look far enough ahead in the future to allow the operators time to react to the instructions from the system.

Behavioral cloning can be formulated into a supervised learning problem by using state-action pairs to train policies. The mean squared error (MSE) is used as the loss function and evaluation metric.

$$MSE = \frac{1}{n}\Sigma_{i=1}^{n}(\hat{y_i} - y_i)^2$$

in which $\hat{y_i}$ is the output from ML model and y_i is the ground truth. To ensure that the models do not rely on a simple correlation, a naive forecasting model that predicts the previous engagement rate value $(t-1)$ is used as a baseline for comparison. It always uses the last observed value as its output.

$$\hat{y_t} = y_{t-1}$$

The policies are applied in the physical sphere of the production procedure in Fig. 1. Behavioral cloning is effective under the condition that collected observations fully cover all experts' trajectories. In practice, this is rarely the case, thus making it brittle and prone to distributional shifts between training and real-life scenarios. In this study, we present the building of initial behavioral cloning policies as seeds that can be grown into maturity via a direct expert demonstration approach in the future. There are two methods: data aggregation and policy aggregation. The learning algorithm will keep collecting observations of expert demonstrations from the operating cycle until it reaches maturity. A mixture of exploration into experiences can be beneficial as imitation learning rarely exceed the expert capacity that it learns from.

IV. Dataset Preparation

A. Data Collection

Mazak Integrex i500 is a 5-axis CNC machine, whose coordinates are depicted in Fig. 3. The C-axis allows the chuck to angle the workpiece. The B-axis allows the cutter to be angled. In a turning process, the value of the C angle is zero, while the B angle is zero in a milling process. The (x, y, z) axes allow linear movement of the cutter.

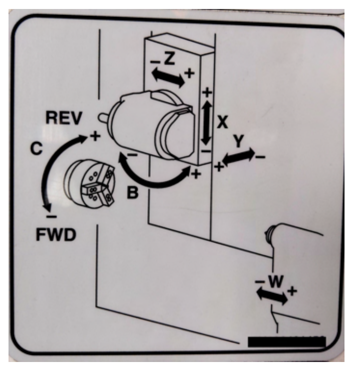


Fig. 3. 5-axis CNC machine. The five-axis include: linear axes (x, y, z), C-axis for workpiece angle, and B-axis for cutter angle

Data was collected for one month. During that time, the CNC machine mass-produced instances of the same machine part. Data is retrieved via the MTConnect protocol in XML format. The data extracting rate from Mazak is 1 Hz when the machine is idle and a maximum of 4Hz when the machine is running. Each field in the XML query has its timestamp, and they are synchronized before being converted into CSV format for further usage. Missing data are handled by forward-fill followed by backward-fill.

B. Data Description

There are 127 signals extracted from the XML query in total. They can be grouped into five categories

- Axes: this category contains information on the servo motors and spindles, such as position and workload.
- Controller: general information that controls the manufacturing is gathered into this category. It includes tool numbers and execution status.

- Door: normally, the machine is not allowed to run at full speed while the chamber door is open for safety reasons. There are two possible values for the door status: CLOSE and UNLATCHED.
- Systems: this category contains information about electricity, hydraulic condition, coolant, pneumatic, and lubrication.
- Auxiliaries: this category contains information about the surrounding environment such as room temperature.
- Resources: this category has information about materials and the operators.

Several signals are either unavailable or irrelevant. Some signals are directly correlated, such as the relative and absolute position of x. We have selected a subset of useful signals listed in Table I. They match the information that operators can observe on the control interface: *Axes-Position* signals are inside the green box; *Axes-Feed rates* and *Axes-Loads* signals are inside the pink box.

A chuck is a device that holds the workpiece in place and allows it to be rotated. When raw material is launched into the CNC machine, the chuck state transits from OPEN or UNLATCH to CLOSED. After the machine part is finished, the chuck is unlatched to release the workpiece. Therefore, the *Chuck state* signal is used to split the data stream into sessions for each machine part. The *Execution* signal is used to determine the operating state of the Mazak machine. Currently, we are only interested in when *Execution* is ACTIVE. To build initial policies, simple scenarios were considered. The narrowing conditions are

- The ML model is for only one tool that performs one machinery action. We choose tool ID number 1, which performs turning. This is the first tool used for each session. From our observations, we saw that the operators had to pay the most attention during the initial use of this tool. This is due to the roughness of the surface of the raw materials at the start of the machining process.
- We eliminate the sequences in which the tools move to the next ready position and only use the sequences for cutting paths. These sequences are transient and could increase the variance of our dataset. The process is detailed in Section IV-C

C. Cutting paths

A cutting path is defined as a period starting when the cutting tools first cut into the workpiece and ending when the cutting tools lift up and rapidly move to the next cutting position. When the cutter moves in the air, the load on the turning spindle significantly reduces toward zero. The task of identifying abrupt changes in the behavior of a time series is called change point detection. We used an implementation of the PELT (Pruned Exact Linear Time) method from the ruptures library [20]. The PELT algorithm identifies an unknown number of changing points with the assumption that the number of change points is linear [21]. The abrupt changes in turning spindle load can also be caused by arbitrary surfaces. To distinguish between cutting paths and transient, periods between change points with a quantile smaller than ten are classified as transient and are eliminated (Figure 4)

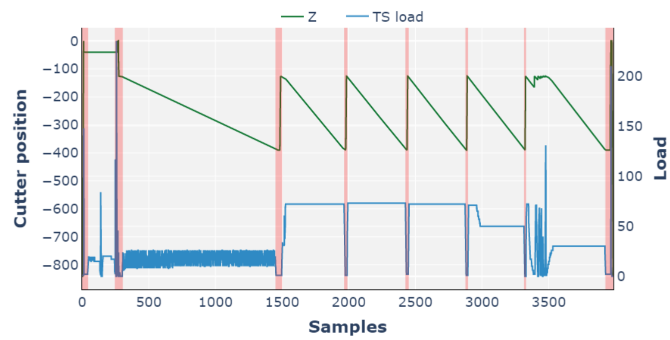


Fig. 4. The green line visualizes the toolpath on Z-axis. Transient periods are highlighted as light pink when the cutter quickly moves to the next position for the cutting path

D. Pre-processing

To get consistent frequency, data is interpolated to a maximum 4Hz. Data sequences in which the CNC machine is not active are dropped out. Experts' trajectories $y_t \in \mathbb{R}^2$ include the engagement rate of linear feed rate *Fovr* and the engagement rate of rotary velocity *Sovr*. In Table I, the signals used as inputs are denoted with a symbol. Preliminary tests showed that the rate of change ($\Delta s_t = s_t - s_{t-1}$) in signals adds more information improving the performance of the ML model. This finding is in line with comments from operators saying they often observe the changes in the signals. Thirteen rate-of-change signals were added to the input signals. They are denoted with an additional symbol. Preliminary tests also showed that a window size of 40 data points gives enough information for good prediction without sacrificing the model's performance. The observational states $X_t \in \mathbb{R}^{N \times M}$ is a vector $X_t = (x_t, x_{t-1}, ..., x_{t-n})$, x_t is a vector with $M = 33$ features and $n = 40$. g is set at one second or four data points.

Data is standardized by centering around the mean and scaling with standard deviation.

$$s_{scaled} = \frac{s - \mu}{\sigma}$$

in which μ is mean and σ is standard deviation. Standardization is a common practice in regression problems. It helps the models to learn quicker and easier. The effects of standardization vary depending on the learning models. In the case of neural networks, standardization positions input into the same scale, thus preventing saturation and speeding up convergence.

We divided the dataset into training, validation, and test sets with a ratio of 80-10-10. Cutting paths, in general, are rougher at the beginning of each session and become smoother later. The dataset is in chronological order before the split to guarantee that the rough and smooth cutting paths are all presented in the three sets.

TABLE I

LIST OF IMPORTANT SENSOR SIGNALS

Category	Signal	Unit	Description
Axes - Positions	**X	mm	Linear positions of the cutting tools
	**Y	mm	
	**Z	mm	
	*B	degree	B angle of the cutting tools
	*C	degree	C angle of the workpiece
Axes - Feed rates	**X frt	mm/s	Linear feed rate of the cutting tools
	**Y frt	mm/s	
	**Z frt	mm/s	
	**MS rpm	rpm	Milling spindle's rotary velocity
	**TS rpm	rpm	Turning spindle's rotary velocity
Axes - Loads	**X load	%	Linear load on the cutting tools
	**Y load	%	
	**Z load	%	
	**MS load	%	Milling spindle's load
	**TS load	%	Turning spindle's load
Axes - Temperature	*X temp	celsius	Temperature of X servo motor
	*Y temp	celsius	Temperature of Y servo motor
	*Z temp	celsius	Temperature of Z servo motor
	*MS temp	celsius	Temperature of milling spindle motor
	*TS temp	celsius	Temperature of turning spindle motor
Controller	**Fovr**	%	Feed rate Override - engagement rate of linear feed rate
	Sovr	%	Spindle speed Override - engagement rate of rotary velocity
	Tools number	-	ID number of tools in used
	Chuck state	-	Status of the chuck. It has three values CLOSED, UNLATCHED, OPEN
	Execution	-	Status of the machine. It has six values ACTIVE, STOPPED, READY, FEED HOLD, INTERRUPTED, PROGRAM STOPPED

* *Input signals*

** *Input signals with added rate of change* $\Delta s_t = s_t - s_{t-1}$

V. MACHINE LEARNING MODELING

A. ML models

We have investigated three families of ML models:

- Regression model with Support vector machine (SVM) [22]. SVM is a classical ML method that is still widely used thanks to its simplicity, reliability, and flexibility. We use the implementation of SVM from the Sklearn library [23] that performs a kernel trick with radial basis function kernel (rbf).
- Ensemble methods are based on the idea that one regressor is weak as it can only capture an aspect of the problem. Combining many weak regressors makes a good model that can generalize well. Boosting methods create sub-models successively, one after another while bagging methods create sub-models independently. Both approaches take a majority vote, in the end, [24]. Gradient boost regressor (GBR) and random forest regressor (RFR) are two popular representatives of boosting and bagging methods, respectively. We also use the implementations from the Sklearn library for the ensemble methods. To avoid overfitting, both GBR and RFR are limited to depth ten.
- The family of deep neural networks has gained popularity in the last decade along with the exponential Big Data and increasing computational power. Two variations of deep neural networks are investigated: fully connected deep networks (DNN) and long-short-term-memory (LSTM). LSTM is a variation of recurrent neural network [25] - a variation of deep neural network that works effectively with time series data. We use the implementation of neural networks from the Tensorflow framework [26]. Hyperparameter search is done with the Hyperband method from Keras tuner [27]. Hyperband is a random search method that randomly tries out different configurations on a specific schedule of iterations per configuration, then proceeds with the selected candidates for longer runs.

Training and validation sets were used for training deep neural networks. For the other two families, we applied cross-validation with nine folds on the combined training and validation sets. It ensured that each ML model was trained and validated with 80-10% of the dataset. Finally, performance on the test set was used as an additional benchmark to evaluate the generality of the ML models.

B. Results

Fig. 5 shows the results from ML methods with y-axis in logarithmic scale. The boxes visualize cross-validation results for SVM, GBR, and RFR models. From top to bottom, the edges are maximum, 3-quantile, median, 1-quantile, and minimum MSE values of the 9-fold. For DNN and LSTM models, the dots represent MSE values on the validation set. The red lines illustrate the baseline values. Detailed measurements of MSE values are listed in Table II.

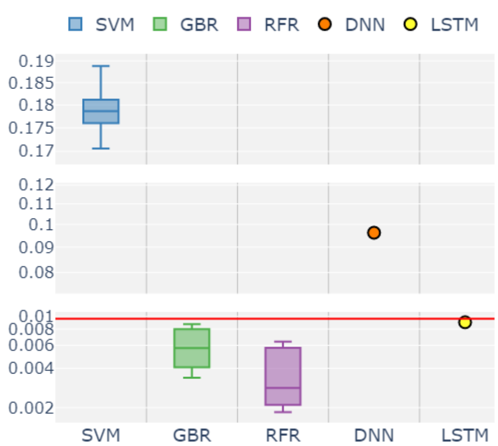
Overall, the engagement rate of spindle velocity *Sovr* is easier to fit than the engagement rate of linear feed rate *Fovr* as its MSE values tend to be lower than *Fovr*'s across all methods. As the problem is non-linear, a linear model like

TABLE II
MSE VALUES OBTAINED FROM DIFFERENT MACHINE LEARNING METHODS. FOR CROSS-VALIDATION, THE VALUES ARE MEDIAN (2-QUANTILE).

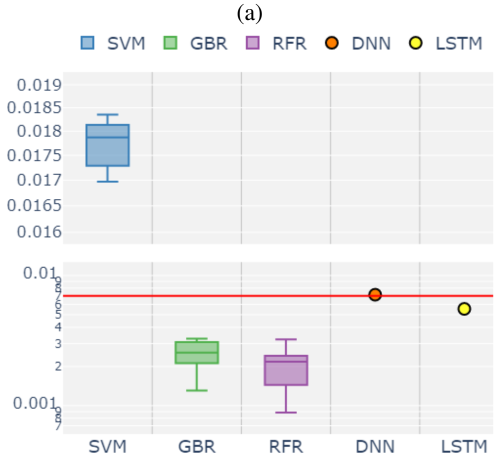
Models	Val MSE		Test MSE	
	Fovr	Sovr	Fovr	Sovr
Baseline	0.0096	0.0070	-	-
*SVM	0.179	0.018	0.134	0.011
*GBR	0.0057	0.0026	0.0056	0.0024
*RFR	0.0028	0.0022	0.0037	0.0021
DNN	0.096	0.0071	0.098	0.0086
LSTM	0.0090	0.0055	0.0042	0.0045

* *Cross validation results*

SVM performs poorly. According to Aytekin et al. [28], a fully connected network can be equivalently represented as a decision tree. Therefore, the DNN model is outperformed by the GBR and RFR as they are ensemble methods with a collection of several sub-decision trees. Only ensemble methods are able to achieve better performances than the baseline performances. Among the ensemble methods, RFR performs better in cross-validation and the test set.



(a)



(b)

Fig. 5. Cross validation performances for *Fovr* signal (a) and *Sovr* signal (b). The y-axis is the MSE. Smaller values give better performances. The red line is the baseline performance from naive forecasting.

VI. DISCUSSION

The linear feed rates (X frt, Y frt, Z frt) and the spindle velocity (MS rpm TS rpm) are directly affected by *Fovr* and the *Sovr* respectively. This increases the risk of causal confusion in which the policies misidentify the effects as the causes for experts' action [29]. The delay g we introduced helps mitigate that risk as the effects of actions at y_t would appear in observational states X_t instead of input $X_{(t-g)}$. Ensemble methods beat naive forecasting with a considerable gap, proving that these models have learned more useful patterns than the naive baseline.

A residual plot is a tool to evaluate regression models. Fig. 6's left subplot, and right subplot shows a standardized residual plot of the RFR's performance on the test set and the marginal distribution of the residuals, respectively. In common cases, the residuals should be scattered around randomly. In this case, the residuals forming parallel lines can be explained by the operator's tendency to use rounded values at multiples of 10 rather than the whole (0-100%) range. As a result, few unique values of the observations lead to the parallel feature of the residual plot [30]. The expected regression problem has been transformed into an ordinal regression problem. This highlights one characteristic where human experience and biases can affect how a problem is solved. The marginal distribution shows that the residuals were condensed around the zero value. This can be attributed to the high correlation between the ground truth and the output predictions. In manufacturing

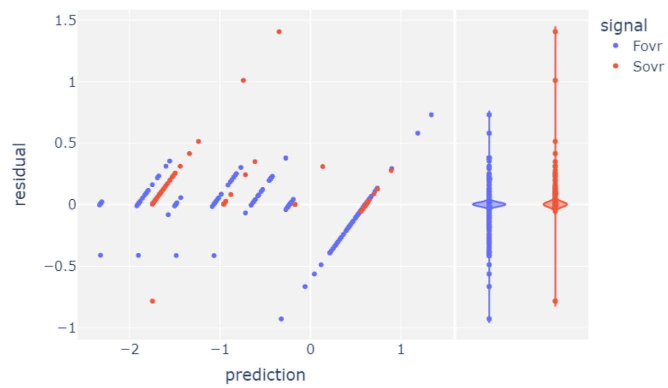
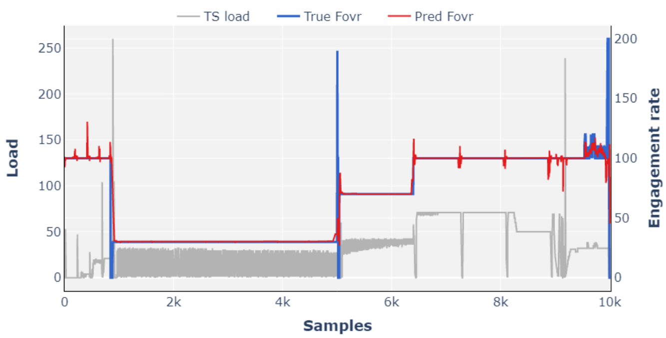
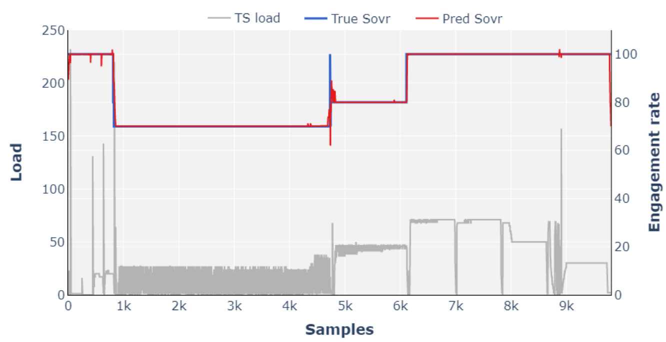


Fig. 6. Residual plot for RFR prediction

settings, the interpretability of models is important to build trust in the algorithms. This is a weakness for ML models, as most are still considered black boxes. Although a decision tree is essentially a set of human-readable rules, it is still tricky for humans to reason about them. Another way to validate the ML models is to observe the behavior on the test set. Fig. 7 investigates the performance of the RFR model in two scenarios where tool number 1 is used in the turning process. The operators use the turning spindle load signal to justify their process interventions. The blue lines in the figure correspond to the operator's interventions, while the red lines are the outputs of the RFR model. It can be seen that the model has learned to correlate the turning spindle patterns to appropriate values for the engagement rates. However, *Fovr* prediction is more sensitive to changes in the turning spindle than *Sovr*. The sudden spikes in the operator's interventions can be explained by the operator turning the override knob past the desired value. The operator then quickly adjusts the knob to compensate and reach the intended value. The interventions where overshooting occurs are considered outliers by the RFR model. As a result, the model tries to smooth the predicted values. Fig. 7 shows that interventions of the engagement rates



(a)



(b)

Fig. 7. Ground truth and prediction from RFR model for *Fovr* signal (a) and *Sovr* signal (b)

are sparse. This is because operators do not need to adjust the engagement rate frequently. However, the CNC machine still requires the full attention of operators. The use of ML decision support systems can bring value by freeing operators from constant monitoring, thereby increasing their productivity as they can work on other tasks. As a first step, the ML models are treated as a decision support system. Thus they must look into the future and make a forecasting decision. There is a trade-off between the time gap and the performances of the models. Fig. 8 examines the trade-off. *Fovr* does not get affected much by the gap compared to *Sovr*. The practicability of delay time for operators to react is an engineering question that needs to be addressed in further research. Ideally, the reaction gap can be reduced significantly when the ML models are integrated directly into production systems. To achieve it, a certain degree of trust is required. This trust should be built through careful risk analysis. Through interactive and

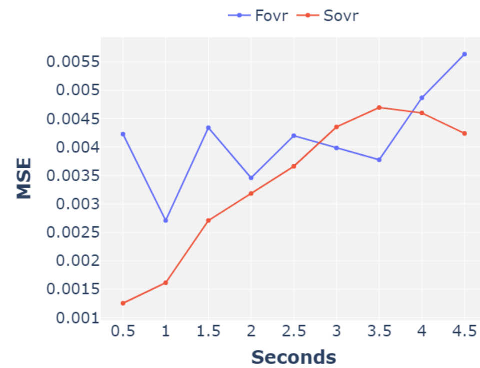


Fig. 8. Cross validation examination of different values for delay g. The ML model in use is RFR. The entire chart is below the baseline values for both *Fovr* and *Sovr*

successful use, trust in the model can be increased.

VII. Conclusions

This study proposes the application of Big Data in imitation learning for expert control in the manufacturing sector, which offers several benefits, including shorter training time, reduced risk for the learning algorithm, and access to abundant amounts of expert demonstrations. The paper presents initial machine learning (ML) models trained from historical data that can evolve into mature control policies through direct policy learning. The ML models control two important configurations during a CNC machine operation for engagement feed rates. They learn initial policies that can map out from the same information operators receive via the control interface to their actions. The best-performing ML model, the Random Forest Regressor, achieved better MSE values than the baseline models. The paper also examined different time gaps, and a small delay was applied between state and action to mitigate the effects of causal confusion and accommodate the design of the decision support system. The control algorithm learned through imitation learning holds significant potential as a decision support system, aiding in workload reduction for operators and providing assistance to novice operators in system control.

However, synthesizing operators' experiences, like handcrafted expert systems, may introduce biases from human knowledge. In this case, bias results from the operator's tendency to truncate the engagement rates to rounded numbers. Ideally, the model should fine-tune the output to optimize the quality of the product surface, production time, and machining tool lifetime. However, it is challenging to enhance a model's performance beyond the operators' experiences in the supervised learning paradigm. To break this limitation, exploration will be added to the learning algorithm in the future. Nevertheless, the ML models show that they can overcome human's faults as they detect and smooth operators' actions to remove sudden spikes.

Although the ML models perform well in theory, implementing them on the shop floor requires further work. Moreira et al.'s supervision controller operated directly on the simulated CNC machine, while this system is meant to function as a decision support system [14]. Therefore, there will be an additional requirement for operators to have enough time to react to the recommendation. A too-short reaction time window can damage the product or machine. One way to meet this requirement is to forecast the need for interventions. As the system's goal is to increase productivity, it's important not to overload the operator with recommendations. Lastly, to transition from a decision support system to an active control system, a level of trust needs to be built via intense verification and validation of the system over time. To make the procedure practical, the system will need parallel watch-dogs to eliminate critical actions.

ACKNOWLEDGMENT

The research activities were carried out with the data and information received from Mekanisk Service Halden. We are grateful for their support and constructive discussions.

REFERENCES

[1] T. Philbeck and N. Davis, "The fourth industrial revolution: Shaping a new era," *Journal of International Affairs*, vol. 72, no. 1, pp. 17–22, 2019.

[2] C. Boles. (2022) Smart manufacturing will lead us to industry 4.0. Forbes. https://www.forbes.com/sites/forbestechcouncil/2022/08/17/smart-manufacturing-will-lead-us-to-industry-40/?sh=6798defd2dc3, visited 2022-08-17.

[3] L. Dalenogare, G. Benitez, and N. Ayala, "The expected contribution of industry 4.0 technologies for industrial performance," *International Journal of Production Economics*, vol. 204, pp. 383–394, 2018.

[4] A. M. Industry. (2022) What industries use cnc machinery. American Micro Industry. https://www.americanmicroinc.com/resources/industries-use-cnc-machinery/, visited 2022-02-02.

[5] T. M. Institute. (2022) Training to win - talent solutions for the new economy. The Manufacturing Institute. https://www.themanufacturinginstitute.org/wp-content/uploads/2020/03/MI-Training-to-Win.pdf, visited 2022-14-10.

[6] J. Espinel. (2021) Introduction to cnc milling cuts and toolpaths. fictiv. https://www.fictiv.com/articles/cnc-milling-cuts-and-toolpaths, visited 2022-21-09.

[7] ——. (2019) How to become a cnc machinist. Intergrative Staffing Group. https://www.integrativestaffing.com/2019/06/11/how-to-become-a-cnc-machinist/, visited 2022-21-09.

[8] S. Newman, A. Nassehi, R. ImaniAsrai, and V. Dhokia, "Energy efficient process planning for cnc machining," *CIRP Journal of Manufacturing Science and Technology*, vol. 5, no. 2, pp. 127–136, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1755581712000223

[9] S. Tajima and B. Sencer, "Global toolpath smoothing for cnc machine tools with uninterrupted acceleration," *International Journal of Machine Tools & Manufacture*, vol. 121, pp. 81–95, 2017.

[10] B. Li, H. Zhang, P. Ye, and J. Wang, "Trajectory smoothing method using reinforcement learning for computer numerical control machine tools," *Robotics and ComputerIntegrated Manufacturing*, vol. 61, p. 101847, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0736584519300419

[11] L. WANG, X. YUAN, H. SI, and F. DUAN, "Feedrate scheduling method for constant peak cutting force in five-axis flank milling process," *Chinese Journal of Aeronautics*, vol. 33, no. 7, pp. 2055–2069, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1000936119303449

[12] T. Mohanraj, J. Yerchuru, H. Krishnan, R. Nithin Aravind, and R. Yameni, "Development of tool condition monitoring system in end milling process using wavelet features and hoelder's exponent with machine learning algorithms," *Measurement*, vol. 173, p. 108671, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0263224120311830

[13] D. F. Hesser and B. Markert, "Tool wear monitoring of a retrofitted cnc milling machine using artificial neural networks," *Manufacturing Letters*, vol. 19, pp. 1–4, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2213846318301524

[14] L. Moreira, W. Li, X. Lu, and M. Fitzpatrick, "Supervision controller for real-time surface quality assurance in cnc machining using artificial intelligence," *Computers & Industrial Engineering*, vol. 127, pp. 158–168, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360835218306211

[15] M. I. Ahmad, Y. Yusof, M. E. Daud, K. Latiff, A. Z. A. Kadir, and Y. Saif, "Machine monitoring system: a decade in review," *The International Journal of Advanced Manufacturing Technology*, vol. 108, pp. 3645–3659, 2020.

[16] W. Luo, T. Hu, C. Zhang, and Y. Wei, "Digital twin for cnc machine tool: modeling and using strategy," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, pp. 1–4, 2019.

[17] W. Sakarinto, H. Narazaki, , and K. Shirase, "A decision support system for capturing cnc operator knowledge," *International Journal of Automation Technology*, vol. 5, no. 5, pp. 655–662, 2011.

[18] T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, and J. Peters, "An algorithmic perspective on imitation learning," *CoRR*, vol. abs/1811.06711, 2018. [Online]. Available: http://arxiv.org/abs/1811.06711

[19] I. Bratko, T. Urbančič, and C. Sammut, "Behavioural cloning: Phenomena, results and problems," *IFAC Proceedings Volumes*, vol. 28, no. 21, pp. 143–149, 1995, 5th IFAC Symposium on Automated Systems Based on Human Skill (Joint Design of Technology and Organisation), Berlin, Germany, 26-28 September. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474667017467164

[20] C. Truong, L. Oudre, and N. Vayatis, "Selective review of offline change point detection methods," *Signal Processing*, vol. 167, p. 107299, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165168419303494

[21] R. Killick, P. Fearnhead, and I. Eckley, "Optimal detection of changepoints with a linear computational cost," *Journal of the American Statistical Association*, vol. 107, pp. 1590–1598, 2012.

[22] C. Cortest and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.

[23] 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*, vol. 12, pp. 2825–2830, 2011.

[24] A. Liaw and M. Wiener, "Classification and regression by randomforest," *JR News*, vol. 2, pp. 18–22, 2002.

[25] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.

[26] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[27] T. O'Malley, E. Bursztein, J. Long, F. Chollet, H. Jin, L. Invernizzi *et al.*, "Kerastuner," https://github.com/keras-team/keras-tuner, 2019.

[28] C. Aytekin, "Neural networks are decision trees," 2022. [Online]. Available: https://arxiv.org/abs/2210.05189

[29] P. de Haan, D. Jayaraman, and S. Levine, "Causal confusion in imitation learning," *CoRR*, vol. abs/1905.11979, 2019. [Online]. Available: http://arxiv.org/abs/1905.11979

[30] S. R. Searle, "Parallel lines in residual plots," *The American Statistician*, vol. 42, p. 211, 1988.