Proceedings of the ASME 2024 19th International
Manufacturing Science and Engineering Conference
MSEC2024
June 17-21, 2024, Knoxville, Tennessee

MSEC2024-125435

BRIEF PAPER: MULTI-TASK BRAIN-INSPIRED LEARNING FOR INTERLINKING MACHINING DYNAMICS WITH PARTS GEOMETRICAL DEVIATIONS

Danny Hoang[1], Hanning Chen[2], Mohsen Imani[2], Ruimin Chen[1], Farhad Imani[1]*

[1]School of Mechanical, Aerospace, and Manufacturing Engineering, University of Connecticut, Storrs CT
[2]Department of Computer Science, University of California, Irvine, Irvine CA

ABSTRACT

Increasing complexity, and requirements for the precise creation of parts, necessitate the use of computer numerical control (CNC) manufacturing. This process involves programmed instructions to remove material from a workpiece through operations such as milling, turning, and drilling. This manufacturing technique incorporates various process parameters (e.g., tools, spindle speed, feed rate, cut depth), leading to a highly complex operation. Additionally, interacting phenomena between the workpiece, tools, and environmental conditions further add to complexity which can lead to defects and poor product quality.

Two main areas are of focus for an efficient automated system: monitoring and swift quality assessment. Within these areas, the critical aspects ascertaining the quality of a CNC manufacturing operation are: 1) Tool wear: the inherent deterioration of machine components caused by prolonged utilization, 2) Chatter: vibration that occurs during the machining process, and 3) Surface finish: the final product's surface roughness. Many research domains tend to focus on just one of these areas while neglecting the interconnected influences of all three. Therefore, to capture a more holistic and comprehensive assessment of a manufacturing process, the overall product quality should be considered, as that's what ultimately counts.

The integration of CNC systems with in-situ monitoring devices such as acoustic sensors, high-speed cameras, and thermal cameras is aimed at understanding the underlying physical aspects of the CNC machining process, including tool wear, chatter, and surface roughness. The incorporation of these monitoring devices has allowed the use of artificial intelligence and machine learning (ML) in smart CNC systems with hopes of increasing productivity, minimizing downtime, and ensuring product quality. By capturing the underlying phenomena that occur during the manufacturing process, users hope to understand the interlinking dynamics for zero-defect automated manufacturing. However, even though the use of ML methods has yielded noteworthy results in analyzing in-situ process data for CNC manufacturing, the black-box nature of these models and their tendency to focus predominantly on single-task objectives rather than multi-task scenarios pose challenges. In real-world part creation and manufacturing scenarios, there is often a need to address multiple interconnected tasks simultaneously which demands models that can multitask effectively. Yet, many ML models designed and trained for singular objectives are limited in their applicability and efficiency in more complex multi-faceted environments.

Addressing these challenges, we introduce MTaskHD*, a novel multi-task framework, that leverages hyperdimensional computing (HDC) to effortlessly fuse data from various channels and process signals while characterizing quality within a multi-task manufacturing operation. Moreover, it yields interpretable outcomes, allowing users to understand the process behind predictions. In a real-world experiment conducted on a hybrid 5-axis CNC Deckel-Maho-Gildemeister,* MTaskHD *was implemented to forecast the quality of three distinct features: left 25.4 mm counterbore diameter, right 25.4 mm counterbore diameter, and 2.54 mm milled radius. Demonstrating remarkable performance, the model excelled in predicting the quality levels of all three features in its multi-task configuration with an F1-Score of 95.3%, outperforming alternative machine learning approaches, including support vector machines, Naïve Bayes, multi-layer perceptron, convolutional neural network, and time-LeNet. The inherent multi-task capability, robustness, and interpretability of HDC collectively offer a solution for comprehending intricate manufacturing dynamics and operations.*

Keywords: Computer Numerical Control, Multi-task Learning, Quality Monitoring

1. INTRODUCTION

CNC subtractive manufacturing, vital for crafting complex parts, relies on programmed instructions for precise material removal via milling, turning, and/or drilling. Factors like tool choice, spindle speed, and feed rate play significant roles as their

*Corresponding author: farhad.imani@uconn.edu complex interactions can lead to defects and lower quality, especially with intricate commands [1]. Integrating CNC with in-situ monitoring such as acoustic sensors and high-speed cameras aims to capture the dynamics of the process, including tool wear, chatter, and surface roughness issues that affect product quality [2, 3]. Analyzing these dynamics individually offers insights into quality but overlooks the interconnected relationships, where tool wear leads to faulty machining and defective components, and chatter causes uncontrolled interactions between the tool and workpiece, collectively affecting the final surface roughness of the product. A holistic analysis of these dynamics, rather than focusing on them individually, provides a deeper understanding of interconnected relationships and potential in-process defects.

Traditional machine learning (ML) in manufacturing analysis such as recurrent neural networks [4], and support vector machines [5] often focuses on single operations resulting in their limitations by the data collected solely during these specific processes. These methods often struggle with analyzing complex parts that require multiple operations. Integrating various processes, or tasks, into a single model enables a more comprehensive and varied approach to manufacturing analysis, facilitating a deeper investigation into the factors influencing the final product quality. Given the limitations outlined above, there is an intrinsic need to develop new models that provide multi-task capabilities for increasing complex manufacturing operations.

Hyperdimensional computing (HDC) is a computational paradigm designed to overcome the limitations of traditional ML, particularly for efficient learning and analyzing out-of-distribution data. HDC is driven by the insight that the cerebellum cortex represents data using high-dimensional spaces [6]. Within HDC, hypervectors are used to represent data and model human memory through defined vector operations, namely bundling, binding, and permutation. Learning from hypervectors is achieved for a wide range of applications, such as qualification of additive manufacturing [7], quality monitoring [8, 9], friction stir welding [10], data selection [11], and graph generation [12]. Despite these applications, exploration of hyperdimensional computing's multi-task learning capabilities has received limited attention.

We propose MTaskHD, a hyperdimensional computing framework for multi-task learning. Using a 5-axis hybrid CNC Deckel-Maho-Gildemeister (DMG) machine, MTaskHD fuses data from various in-situ channels and process signals, characterizing the geometric quality of three distinct features within a multi-task manufacturing operation. The remainder of this paper is as follows: Section 2 introduces the HDC framework for model implementation and training. Section 3 discusses the detailed setup for the experiments and results. Finally, the conclusion of this proposed study is presented in Section 4.

2. RESEARCH METHODOLOGY

2.1 Vector Operations

In this model, two principal operations are employed: bundling and binding. Bundling, denoted by (+), involves the element-wise addition of hypervectors and produces a resultant hypervector that retains similarity to its components. This process serves as a memory mechanism in computations. Binding, represented by (∗), is the element-wise multiplication or the Hadamard product of hypervectors. The hypervector generated from binding is distinct from the original hypervectors, making this operation suitable for associating hypervectors together. Additionally, the similarity of hypervectors is computed using cosine similarity δ defined as:

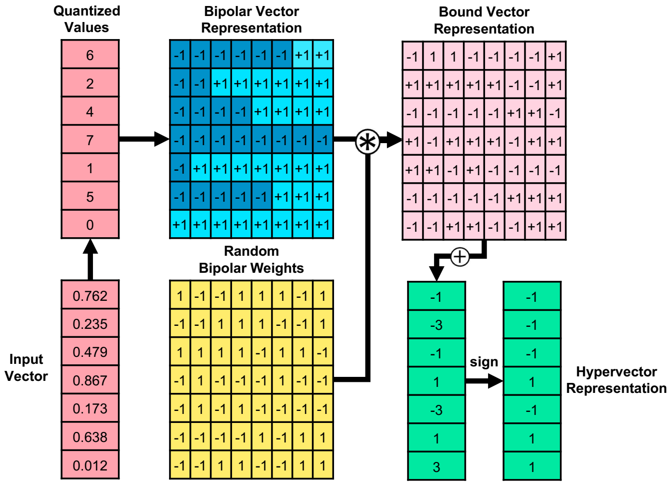
Quantized Values
Bipolar Vector Representation
Bound Vector Representation
Random Bipolar Weights
Input Vector
sign
Hypervector Representation

FIGURE 1: DEPICTION OF ENCODING PROCEDURE.

$$\delta(\vec{H}_1, \vec{H}_2) = \frac{\vec{H}_1 \cdot \vec{H}_2}{\|\vec{H}_1\|\|\vec{H}_2\|} \qquad (1)$$

where the numerator is the dot product between $\vec{H}_1$ and $\vec{H}_2$. By employing these vector operations and comparison metrics, the model can be created to memorize important features from the hypervectors and updated according to the degree of similarity between hypervectors. Additionally, classification can be performed utilizing the highest similarities.

2.2 Encoding

The transformation of data into hypervectors is done utilizing the density encoding framework introduced in [13]. We define input vectors as $\vec{x}_i = \{\vec{x}_{i,1}, \vec{x}_{i,2}, ..., \vec{x}_{i,n}\}$ where there are n input vectors. Each value in the input vectors is then quantized to the nearest integers scaled by a selected dimension size d. Each integer is then represented by a bipolar vector of dimension d with elements of {-1, +1}, where the count of -1 elements matches the integer value, and the remaining elements are +1. These bipolar vectors are subsequently multiplied with randomly generated bipolar weight hypervectors of dimension d for each corresponding bipolar vector and then bundled together. Lastly, the hypervector is passed through a sign function to create the final hypervector representation of the input vector. The full procedure is depicted in Figure 1.

2.3 Multi-task Learning

The multi-task learning involves a hierarchical binding and bundling procedure that produces a final hypervector corresponding to a task and label. Given a task T with n samples, c time-series channels that are of parameter type p, and a corresponding label y, the channel data of each sample is first encoded utilizing the procedure described in 2.2. Next, for each channel, there is a corresponding randomly generated bipolar hypervector bound to the encoded channel hypervectors to associate the hypervector with its specific channel. Each bound hypervector is then bundled with its corresponding parameter type p and bound using a randomly generated bipolar hypervector, establishing an association between the hypervector and the specific parameter type. Lastly, parameter hypervectors for each respective sample are bundled together to create the final hypervector representation corresponding to the samples from task T and label y.

After construction of hypervectors for each input sample and task, those with the same label y are bundled together to create the representative label hypervectors $\mathbf{L}$. Subsequently, a similarity check is conducted between the sample and label hypervectors to identify which samples are incorrectly predicted. Given a query hypervector $\mathbf{Q}$ with label y that is mispredicted by the current model as y', we iteratively update both the correct and mispredicted label hypervectors $\mathbf{L}_y$ and $\mathbf{L}_{y'}$ as:

$$\begin{aligned} \mathbf{L}_y &\leftarrow \mathbf{L}_y + \eta[1 - \delta(\mathbf{L}_y, \mathbf{Q})] \times \mathbf{Q} \\ \mathbf{L}_{y'} &\leftarrow \mathbf{L}_{y'} - \eta[1 - \delta(\mathbf{L}_{y'}, \mathbf{Q})] \times \mathbf{Q} \end{aligned} \tag{2}$$

where η indicates the learning rate of model. When $\delta(\mathbf{L}_y, \mathbf{Q}) \simeq \delta(\mathbf{L}_{y'}, \mathbf{Q})$, the model only updates a small portion since the similarities will be close to each other. On the other hand, when the prediction is far from the true label the retraining algorithm will update the model significantly because the value of $\delta(\mathbf{L}_y, \mathbf{Q})$ is smaller than $\delta(\mathbf{L}_{y'}, \mathbf{Q})$. This iterative procedure exploits the similarities between hypervectors to efficiently update and retrain the model while taking into account all different tasks and corresponding labels.

3. EXPERIMENTAL DESIGN & RESULTS

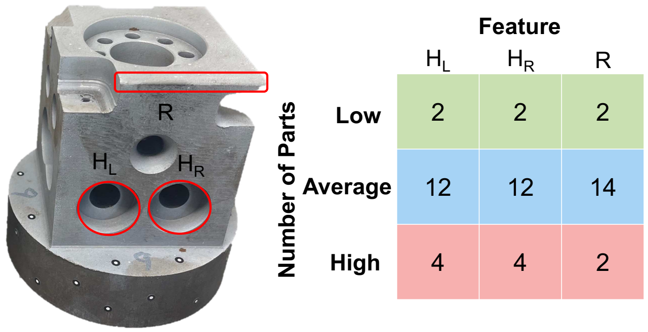

FIGURE 2: FINAL BUILD OF THE DESIGNED PART WITH FEATURES 25.4 MM LEFT COUNTERBORE DIAMETER, 25.4 MM RIGHT COUNTERBORE DIAMETER, AND 2.54 MM MILLED RADIUS HIGHLIGHTED. THE TOTAL NUMBER OF PARTS FOR EACH GEOMETRIC DEVIATION LEVEL IS SHOWN ON THE RIGHT.

MTaskHD was evaluated utilizing data collected from a LASERTEC 65 DED hybrid CNC DMG machine at the Connecticut Center for Advanced Technology (CCAT). A total of 18 parts were fabricated using the subtractive CNC portion of the machine from 1040 steel blocks each with original dimensions 76.2 mm x 76.2 mm x 76.2 mm. Each part was created incorporating 42 manufacturing operations, or tasks, to create 47 features such as chamfers, holes, rounded corners, and pockets. During the manufacturing operations, over 91 time series channels were captured at 500 Hz by a Siemens Simatic IPC227E corresponding to the 5 axes (i.e., three linear axes, and rotary A and C axes), and spindle of the machine. Each of the channels had a corresponding type of process parameter such as current, load, power, and torque. After creation of the parts, a GOM ATOS ScanBox was used to measure each feature which was then compared against the average respective feature measurements. Taking these measurements, labels corresponding to the geometric deviation of each feature were created utilizing computed Z-scores where scores below -1 indicate low, scores between -1 and 1 indicate average, and scores above 1 indicate high levels of deviation. Each geometric deviation level also had a corresponding task label to show which task created which feature.

From these task and geometric deviation labels, two homogeneous tasks used to create 25.4 mm diameter counterbore holes and one heterogeneous task used to create a 2.54 mm milled radius feature were chosen for analysis. The purpose of choosing these three tasks was to analyze how homogeneous or heterogeneous tasks can be learned from each other within the same model. The left counterbore hole, right counterbore hole, and milled radius are labeled as H_L, H_R, and R, respectively, and are depicted in Figure 2. A total of nine labels were used as there were three features and three geometric deviations; each feature is referred to as a task and used interchangeably.

The type of process parameters chosen as inputs were axis position, command speed, control differential I, control differential II, contour deviation, current, encoder position I, encoder position II, load, power, torque, torque feed forward, and velocity feed forward. There were a total of 64 channels from these process parameters and which were then normalized using a min-max scaler. Due to the low number of parts but an abundance of time series data, more samples corresponding to each label were created using non-overlapping n-gram windows over the time-series data. This technique ensures an adequate number of samples to train on rather than utilizing the full length of the data. From these generated samples, the number of samples corresponding to each label was balanced ensuring the unbalanced labels did not influence the results during training.

MTaskHD was implemented in Python using PyTorch. Sensitivity analysis was then conducted to determine the influence of the amount of data during training on the performance of the model. A window size of 10, dimension size of 5000, a learning rate of 1.7, and 181 epochs, or update iterations, were used as these values achieved stabilized results. Figure 3 depicts the performance of each individual feature along with various combinations of the three features. The legend specifies the proportion of data allocated for training, with 0.1 representing 10% of the data, 0.3 denoting 30%, and so forth. The general trend of all combinations is performance increases as the proportion of training data increases. This can be explained due to MTaskHD acquiring a more diverse and varied data set to train on, enabling it to update the label hypervectors more effectively. An interesting observation is the performance of the model when just using feature R. This model achieves metrics above 0.95 with only 10 percent of the data and improves to maximum of 0.985 with 90 percent of

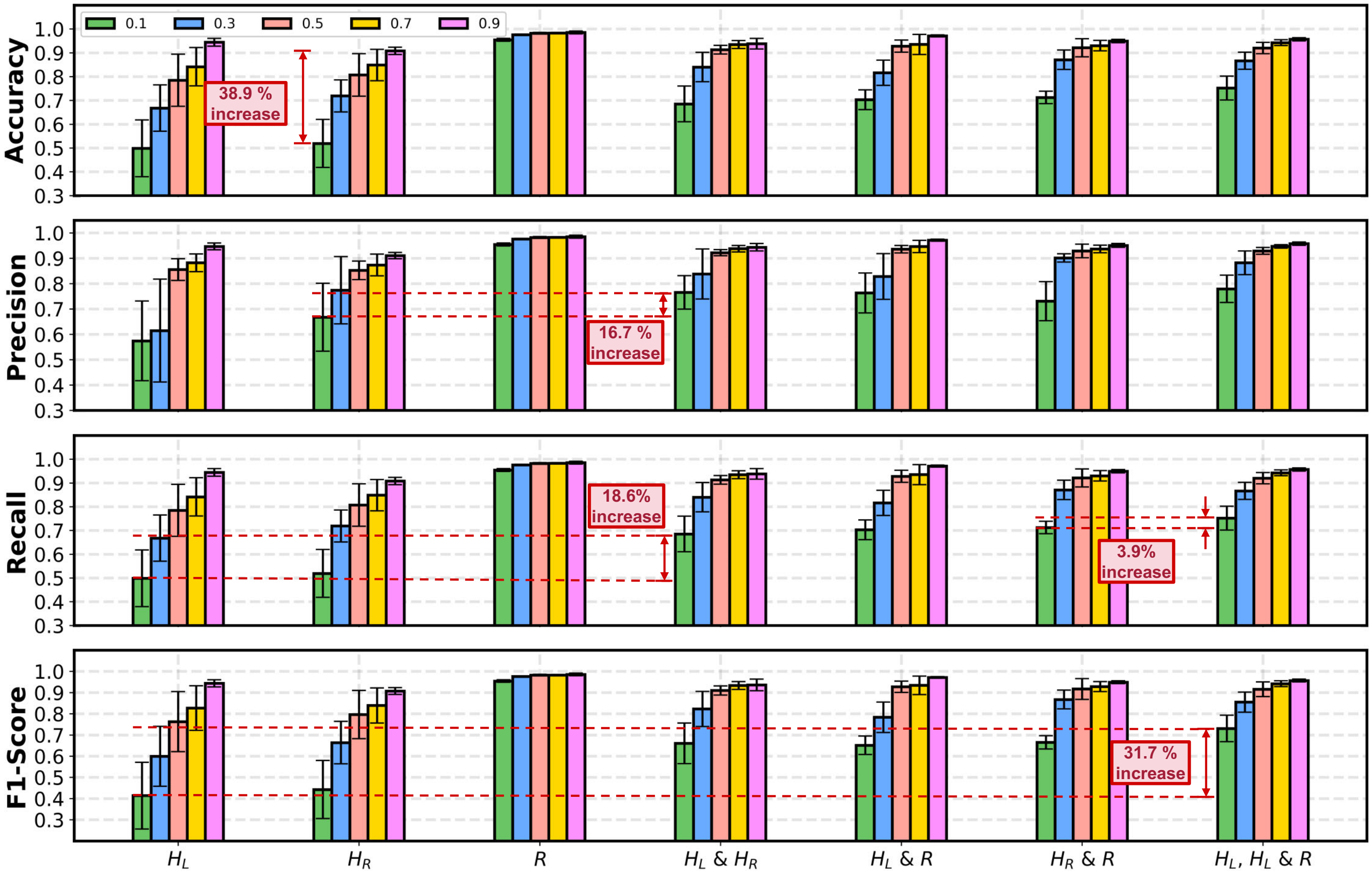

FIGURE 3: SENSITIVITY OF MODELS CONCERNING COMBINATIONS OF THE PROPORTION OF TRAINING DATA USED FOR THE THREE FEATURES AND THEIR ACCURACY, PRECISION, RECALL, AND F1-SCORE.

the data. This observation underscores the robustness and effectiveness of the model in capturing critical information from this feature even with limited samples.

Analyzing the impact of adding homogeneous features together, one can see that performance improves across all metrics. For example, when utilizing 10% of the data for training and using only H_L, or H_R, performance of the models was around 0.4 for F1-Score. By utilizing both features together in a multi-task configuration, performance increased by around 0.18 showing a dramatic uplift. This emphasizes that by including data from similar tasks, the model is able to learn more nuances from the data compared to utilizing a singular objective. This also underscores the capability of the model in analyzing features even with a limited amount of data as long as one is able to introduce data of a similar nature. The inclusion of all three features also shows an increase in performance across various proportions of training data compared to the singular features with the exception of just using feature R. Overall, by utilizing information from a diverse selection of tasks, the model enhances its capability to isolate and comprehend information from the input, thereby enhancing its proficiency in identifying crucial signals, even in the presence of noise or contradictory patterns. The integration of multiple tasks contributes to a broader and more adaptable learning approach, ultimately resulting in a more effective model.

Finally, the multi-task capability of MTaskHD for three tasks is compared against conventional machine learning models such as support vector machines (SVM), Naïve Bayes (NB), multilayer perceptron (MLP), convolutional neural network (CNN), and time-LeNet (t-LeNet). These models were implemented using Python, specifically using scikit-learn and TensorFlow. All models are run 50 times and the average performance and standard deviations are reported. As shown in Table 1, regarding F1-Score, MTaskHD achieves a score of 0.953 and outperforms SVM, NB, MLP, CNN, and t-LeNet by 41.9%, 51.6%, 47.2%, 29.3%, and 15.8%, respectively. The performance of MTaskHD is consistent across all metrics and shows its incredible robustness with the inclusion of varying tasks. This cannot be said of the other models as their performance is much lower which is explained by the models being confused with the addition of more out-of-distribution data across different tasks. Overall, MTaskHD showcases its incredible capability in analyzing and predicting the various geometric deviations across all three features and is a model that is more holistic and nuanced.

4. CONCLUSION

This paper introduces MTaskHD for multi-task learning in CNC subtractive manufacturing. Outperforming traditional machine learning models, this study showcases MTaskHD's effec-

TABLE 1: PERFORMANCE METRICS OF THE MTaskHD COMPARED TO OTHER CHARACTERIZATION ALGORITHMS FOR THREE TASKS.

Model	Accuracy	Precision	Recall	F1-Score
MTaskHD	0.953 ± 0.011	0.955 ± 0.008	0.953 ± 0.011	0.953 ± 0.012
SVM	0.552 ± 0.021	0.572 ± 0.026	0.552 ± 0.021	0.534 ± 0.024
NB	0.467 ± 0.018	0.425 ± 0.018	0.467 ± 0.018	0.437 ± 0.018
MLP	0.532 ± 0.013	0.609 ± 0.030	0.532 ± 0.013	0.481 ± 0.030
CNN	0.681 ± 0.036	0.691 ± 0.046	0.681 ± 0.036	0.660 ± 0.044
t-LeNet	0.806 ± 0.181	0.812 ± 0.199	0.806 ± 0.181	0.795 ± 0.201

tiveness in complex manufacturing, highlighting its robustness and adaptability in understanding and predicting product quality.

ACKNOWLEDGMENTS

This work was supported in part by the DARPA Young Faculty Award, the National Science Foundation (NSF) under Grants 2127780, 2319198, 2321840, 2312517, and 2235472, the Semiconductor Research Corporation (SRC), the Office of Naval Research through the Young Investigator Program Award, and Grants N00014-21-1-2225 and N00014-22-1-2067. Additionally, support was provided by the Air Force Office of Scientific Research under Award FA9550-22-1-0253, along with generous gifts from Xilinx and Cisco. The authors gratefully acknowledge the valuable contributions from the Connecticut Center for Advanced Technology (CCAT), particularly Nasir Mannan and Ruby ElKharboutly, for this research.

REFERENCES

[1] Ntemi, Myrsini, Paraschos, Spyridon, Karakostas, Anastasios, Gialampoukidis, Ilias, Vrochidis, Stefanos and Kompatsiaris, Ioannis. “Infrastructure monitoring and quality diagnosis in CNC machining: A review.” *CIRP Journal of Manufacturing Science and Technology* Vol. 38 (2022): pp. 631–649.

[2] Zhu, Lida and Liu, Changfu. “Recent progress of chatter prediction, detection and suppression in milling.” *Mechanical Systems and Signal Processing* Vol. 143 (2020): p. 106840.

[3] Moreira, Lorena Caires, Li, WD, Lu, Xin and Fitzpatrick, Michael E. “Supervision controller for real-time surface quality assurance in CNC machining using artificial intelligence.” *Computers & Industrial Engineering* Vol. 127 (2019): pp. 158–168.

[4] An, Qinglong, Tao, Zhengrui, Xu, Xingwei, El Mansori, Mohamed and Chen, Ming. “A data-driven model for milling tool remaining useful life prediction with convolutional and stacked LSTM network.” *Measurement* Vol. 154 (2020): p. 107461.

[5] Liu, YC, Hu, XF and Sun, SX. “Remaining useful life prediction of cutting tools based on support vector regression.” *IOP Conference Series: Materials Science and Engineering*, Vol. 576. 1: p. 012021. 2019. IOP Publishing.

[6] Chen, Ruimin, Sodhi, Manbir, Imani, Mohsen, Khanzadeh, Mojtaba, Yadollahi, Aref and Imani, Farhad. “Brain-inspired computing for in-process melt pool characterization in additive manufacturing.” *CIRP Journal of Manufacturing Science and Technology* Vol. 41 (2023): pp. 380–390.

[7] Imani, Farhad and Chen, Ruimin. “Latent Representation and Characterization of Scanning Strategy on Laser Powder Bed Fusion Additive Manufacturing.” *ASME International Mechanical Engineering Congress and Exposition*, Vol. 86649: p. V02BT02A009. 2022. American Society of Mechanical Engineers.

[8] Hoang, Danny, Mannan, Nasir, ElKharboutly, Ruby, Chen, Ruimin and Imani, Farhad. “Edge Cognitive Data Fusion: From In-Situ Sensing to Quality Characterization in Hybrid Manufacturing Process.” *International Manufacturing Science and Engineering Conference*, Vol. 87240: p. V002T06A029. 2023. American Society of Mechanical Engineers.

[9] Hoang, Danny, Errahmouni, Hamza, Chen, Hanning, Rachuri, Sriniket, Mannan, Nasir, ElKharboutly, Ruby, Imani, Mohsen, Chen, Ruimin and Imani, Farhad. “Hierarchical representation and interpretable learning for accelerated quality monitoring in machining process.” *CIRP Journal of Manufacturing Science and Technology* Vol. 50 (2024): pp. 198–212.

[10] Hoang, Danny, Chen, Ruimin, Mishra, Debasish, K. Pal, Surjya and Imani, Farhad. “Data Fusion Cognitive Computing for Characterization of Mechanical Property in Friction Stir Welding Process.” *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Vol. 87295: p. V002T02A055. 2023. American Society of Mechanical Engineers.

[11] Chen, Ruimin, Imani, Mohsen and Imani, Farhad. “Joint active search and neuromorphic computing for efficient data exploitation and monitoring in additive manufacturing.” *Journal of manufacturing processes* Vol. 71 (2021): pp. 743–752.

[12] Chen, Hanning, Zakeri, Ali, Wen, Fei, Barkam, Hamza Errahmouni and Imani, Mohsen. “HyperGRAF: Hyperdimensional Graph-Based Reasoning Acceleration on FPGA.” *2023 33rd International Conference on Field-Programmable Logic and Applications (FPL)*: pp. 34–41. 2023. IEEE.

[13] Kleyko, Denis, Kheffache, Mansour, Frady, E Paxon, Wiklund, Urban and Osipov, Evgeny. “Density encoding enables resource-efficient randomly connected neural networks.” *IEEE Transactions on Neural Networks and Learning Systems* Vol. 32 No. 8 (2020): pp. 3777–3783.