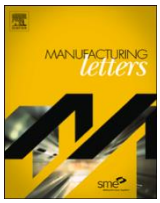
Available online at www.sciencedirect.com

Manufacturing Letters

Manufacturing Letters 00 (2023) 000–000



52nd SME North American Manufacturing Research Conference (NAMRCs 52, 2024)

Online real-time machining chatter sound detection using convolutional neural network by adopting expert knowledge

Eunseob Kim[a,b], Thu Bui[c], Junyi Yuan[a], S Chandra Mouli[c], Bruno Ribeiro[c], Raymond A. Yeh[c], Michael P. Fassnacht[b], Martin B.G. Jun[a,b,*]

[a]*School of Mechanical Engineering, Purdue University, West Lafayette, IN 47907, USA*
[b]*Indiana Next Negeration Manufacturing Competitiveness Center (IN-MaC), Purdue University, West Lafayette, IN 47906, USA*
[c]*Department of Computer Science, Purdue University, West Lafayette, IN 47907, USA*

* Corresponding author. Tel.: +1-765-494-3376; *E-mail address:* mbgjun@purdue.edu

Abstract

In machining processes, chatter detection remains a pivotal challenge, impacting both the quality and efficiency of manufacturing operations. This study introduces an approach that synergizes expert knowledge with the capabilities of advanced convolutional neural networks (CNNs) to enhance chatter detection. A comprehensive monitoring framework is proposed to adopt expert knowledge that digitizes machine tool and sound data, effectively labeling chatter events. This study merges human expertise in identifying milling tool chatter sounds with CNN architecture, marking a notable advancement in blending machining insights of experts with modern artificial intelligence (AI) technologies. The proposed chatter prediction architecture is distinguished by its incorporation of an attention block, fusing outputs from the AlexNet model with cutting parameters. This model outshines baseline models in both in-distribution (ID) and out-of-distribution (OOD) testing datasets. In OOD testing, the proposed model achieved an impressive accuracy of 94.51%, markedly surpassing the standalone CNN model's accuracy of 88.66%. Real-time 3D visualization of machining operations is demonstrated through the successful implementation of the trained model on a Raspberry Pi.

Keywords: Sound monitoring; Chatter detection; Convolutional neural network; Expert knowledge; MTConnect

1. Introduction

Monitoring machining operations is vital for maintaining quality and efficiency, particularly when production issues arise. Online and real-time monitoring enables quick responses to problems, enhancing productivity and reducing downtime. This capability is crucial in the progression towards intelligent unmanned manufacturing systems, where instant detection and resolution of machining issues are key to maximizing efficiency and advancing automation [1]. In this progression, the emphasis on knowledge and learning becomes increasingly critical [2]. As evolving, monitoring systems respond to immediate challenges while simultaneously gathering data and learning from each operation.

Machining chatter, caused by unstable vibrations, can adversely affect the final part quality. It not only compromises the integrity of the product but can also lead to tool damage and even reduce the lifespan of the machine tool [3]. Quick detection of chatter enables quick adjustments in spindle speed or an immediate halt of the operation, thereby preventing damage to tool, workpiece, and the equipment. Ideally, using stability lobes derived from frequency response functions during the planning stage can help establish cutting parameters and tool paths that prevent chatter [3, 4]. However, the structural dynamics of the tool and workpiece can change

during operation and vary with tool wear [5-7], leading to a lack of widespread adoption of online chatter detection methods in actual production settings.

In real-world shop floor environments, experienced operators utilize their finely honed auditory skills to identify chatter. This expert knowledge, developed through years of experience, enables them to discern the distinct sounds of chatter amidst various cutting conditions and with different tools and workpieces. Other than spindle speed and depth of cut, regenerative milling chatter is affected by various cutting parameters [8-11] such as width of cut, feed rate, and cutting directions. In situations where chatter is anticipated, operators are frequently observed with their hands on the spindle override knob, their attention finely tuned to the sounds of the machining operation. This human audible sensing is not just a skill but a critical asset in environments where experimental data collection on chatter can be challenging and costly. Considering expensive data collection and labels [12], digitizing chatter events with sound data in a systematic manner is more imperative. Their ability to detect chatter by ear provides a practical and valuable foundation for developing artificial intelligence (AI) model, demonstrating the importance of this expert knowledge in addressing and swiftly resolving potential issues in the machining process.

Chatter detection using sound in machining processes has received substantial attention, prompting recent investigations into diverse deep learning techniques aimed at addressing this critical challenge. In a notable study [13], the researchers harnessed audio data from CNC (Computer Numerical Control) machines, revealing promising findings. The study demonstrated that CNC audio data could be effectively reduced to a two-dimensional spectrogram representation with minimal information loss due to its inherent compressibility. This condensed data proved highly effective for chatter detection, exhibiting clear differentiation between chatter and non-chatter regions. These findings underscore the potential of employing deep learning models in audio-based chatter classification. Another intriguing approach [14] involves the amalgamation of multi-sensor data, acquired from accelerometers and microphones, as input to a Backpropagation neural network. This innovative approach yielded a significant enhancement in chatter classification accuracy. The proposed technique achieved an impressive accuracy of 97.56%, surpassing models relying solely on either vibration data (90.29%) or microphone-based audio signals (92.68%). This approach underscores the value of integrating multiple data sources and underscores the effectiveness of deep learning in bolstering chatter detection accuracy. In a distinct study [15], researchers delved into the integration of vibration data and key cutting parameters—depth of cut and spindle speed—with a deep learning model utilizing Convolutional Neural Networks (CNN). Notably, this research introduced the concatenation of cutting parameters with vibration data as input. The results showed remarkable accuracy in in-distribution testing, with an average accuracy of 99.82%. This high level of accuracy suggests that deep learning models can effectively utilize cutting parameters in conjunction with vibration signals to enhance chatter detection. A combining machine learning algorithms with a physics-based model is presented in [16]. This hybrid model significantly enhances the accuracy of chatter detection in milling processes by using CNN model with the energy-based model. Continuously adapting and learning during machine operations, the model improves its predictive capabilities over time. Demonstrating high accuracy rates in both machine state detection and chatter detection, this approach highlights the effectiveness of integrating machine learning with traditional physics-based methods. Collectively, these studies underscore the substantial potential of deep learning techniques in the realm of chatter detection. They emphasize the adaptability of deep learning models to a range of data sources, including audio, multi-sensor fusion, and manufacturing features, ultimately contributing to the advancement of precise and efficient chatter detection systems within machining processes.

On top of that, in [17], to digitize operator's expertise and cognitive capabilities, the study focuses on developing an advanced chatter detection system, leveraging human-machine interaction. It utilizes operator reaction times to computer-generated chatter sounds for establishing a detection threshold, balancing the trade-off between early detection and false positives. The research demonstrates the system's effectiveness in accurately detecting chatter with limited operator input, highlighting the significance of operator experience in enhancing detection accuracy.

Building on previous efforts to harness human expertise for enhancing chatter detection, our study introduces a comprehensive framework that digitizes both machining and sound data for real-time monitoring. Our research advances the field by directly integrating human auditory judgments with advanced machine learning techniques. We leverage the MTConnect data integration standard for CNC machine tools, a strategy not previously reported, to systematically capture cutting parameters and sounds. This integration reflects a leap towards digitizing the nuanced auditory skills of experienced operators, translating their tacit knowledge into a format that deep learning models can utilize effectively.

Our approach is distinguished by incorporating sound signal and machining context, inspired by the cognitive process of human listening and contextual analysis. This is designed to significantly improve the model's predictive accuracy by considering both the machining parameters and the context, thereby closely emulating how human experts identify chatter. Through this comprehensive framework employing CNC machine tool data integration standard, we aim to not only capture, and label chatter events accurately but also bridge the gap between traditional machining expertise and contemporary AI technologies, marking a step forward in the application of AI-driven chatter detection systems. This paper details the development and application of this novel architecture utilizing CNN and cutting parameters, underscoring our contribution to the digitization of human expert knowledge for enhanced chatter detection in machining processes.

2. Monitoring system

2.1. Monitoring framework

The monitoring system for this study aims to digitize human knowledge data for chatter detection labeling with

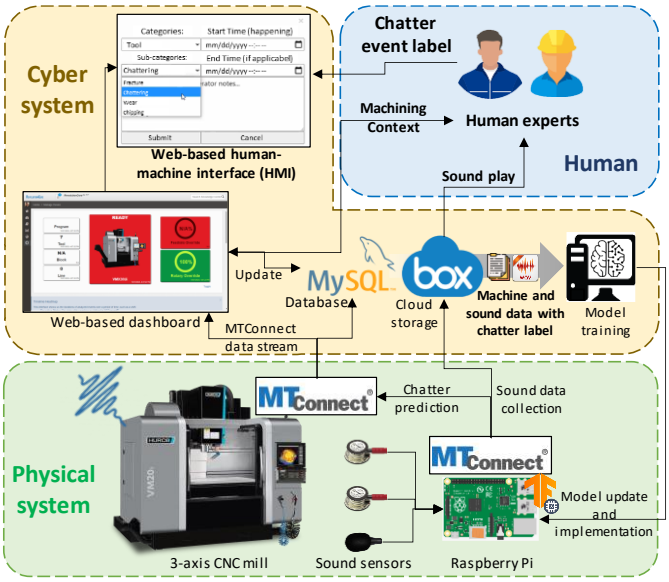


Fig. 1 Monitoring framework adopting expert knowledge

machining context. As shown in Fig. 1, the schematic of chatter monitoring framework is divided into three parts: “Physical system”, “Cyber system”, and “Human”. The “Physical system” where machining and sensing happen consists of the CNC machine, sound sensors, and edge computer. The edge computer plays a role of streaming the sound sensor signals as well as implementing the AI models. To make real-time data stream and enable integrated monitoring from the machine and sensors, MTConnect [18] as a middleware was applied for this system. The MTConnect standard offers a semantic vocabulary for manufacturing equipment to provide structured data with no proprietary format.

The target machine of this study was a vertical 3-axis CNC mill (VM20i, Hurco). The machine is equipped with the Hurco WinMax® MTconnect adapter which runs on the machine controller. This adapter provides real-time access to basic operational information as well as NC code program name, NC code block number, and movement information, such as spindle speed, axis positions, and so on. The component structure and data items of the machine’s MTConnect is illustrated in Fig. 2 (A). MTConnect adapter of the machine samples data at approximately 0.1-second intervals. The MTConnect adapter for sound sensor was written in Python using Advanced Linux Sound Architecture (ALSA) and PyAudio module. This MTConnect adapter was specifically designed for the sound sensors and was executed on a Raspberry Pi [19]. In this study, three sound sensors were used. The MTConnect structure for the sound sensors is shown in Fig. 2 (B). Each data stream, represented by the *Displacement* type and *Time Series* representation according to MTConnect standard, contains sound data in 48 kHz sampling rate with 2^{11} chunk size and 16-bit depth resolution of the mono-channel. A single sample data stream of MTConnect includes 2048 data points, which have a length of approximately 42.67 msec. The timestamp of each sample data stream in MTConnect represents the exact time of the last observation. The machine controller and the Raspberry Pi were in the same router network. In addition, both used the

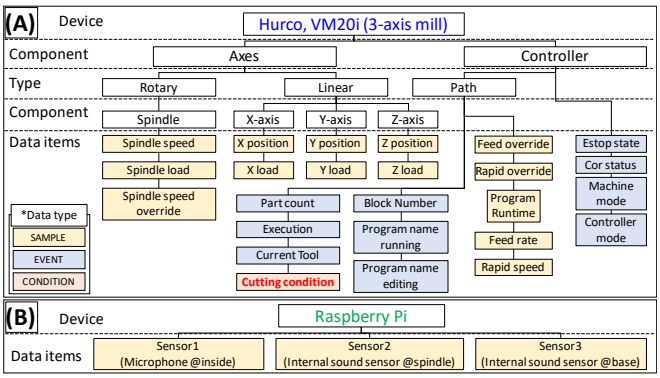


Fig. 2 Tree diagram of MTConnect data items: (A) Hurco, VM20i, and (B) Raspberry Pi

NIST Internet time server. This time synchronization enables matching timestamps of machine data and sensor data.

The “Cyber system” consists of database, web-based dashboard, and model training client. The MTConnect data from the machine is continuously transmitted to the database and web-based dashboard. The web-based dashboard which is developed by ASP.NET Core, which is an open-source web framework, visualizes machine data connected with MySQL database. The web-based dashboard was designed to have a human-machine interface (HMI) in which operators make a log or event when chatter happens. The HMI connects between the “Cyber system” and “Human”. The web-based dashboard, featuring timestamps aligned with machining contexts such as NC code lines and cutting parameters, allows for the precise tagging of chatter events as well as human experts figure out the machining context such as cutting parameters, tool, workpiece information, and so on. This means each chatter label can be specifically matched to a corresponding tool path, enhancing the accuracy and relevance of the data. The log from HMI is also stored in the database. Sound data is continuously collected in 1-second length of each in WAV format stored in a cloud storage, BOX. The filename of each WAV file contains timestamp information. By synchronizing the timestamps between the machine and the sensor, operators can listen to the recorded sound corresponding to a specific tool path in an NC code. Although the web-based interface does not yet support fully automated sound playback, it is still feasible to access the collected WAV files by filtering based on timestamp. Furthermore, these files can be integrated into the web-based dashboard for playback and visualization purposes.

The machine and sound data, along with chatter labels, are automatically uploaded for training the chatter detection model. Once trained, this model is transferred to a Raspberry Pi for real-time chatter prediction. The Raspberry Pi, utilizing the MTConnect sound data stream, simultaneously manages data collection and model execution. The output of the real-time model, which predicts chatter, is then relayed to the machine’s MTConnect, specifically to the ‘Cutting condition’ data item as illustrated in the bottom of Fig. 2 (A). This process ensures seamless integration of machine data with AI model predictions for enhanced operational efficiency. Although certain features are not yet fully automated, we have successfully tested this monitoring framework for both the development of an online chatter prediction model and its real-time deployment.

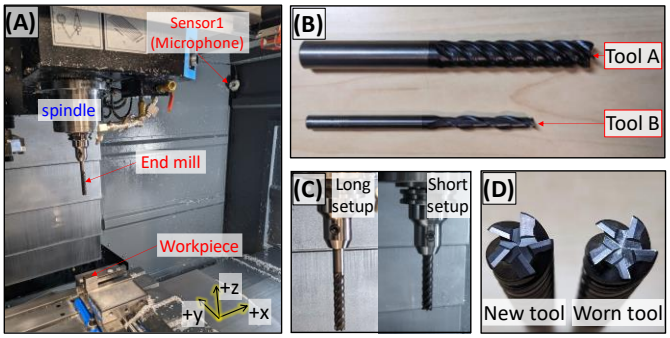


Fig. 3 Experimental setup and variations of tool type, setup, and condition

Table 1 In-Distribution (ID) and Out-Of-Distribution (OOD) datasets

Category	Workpiece	Tool type	Tool setup	Tool condition	Table
ID 1	Steel 4140	A	Long	Worn	A. 1,2
ID 2	Steel 4140	A	Long	New	A. 3
ID 3	Steel 1018	A	Long	Worn	A. 4
ID 4	AL 6061	B	Long	Worn	A. 5-7
OOD 1	Steel 4140	A	Long	Worn	A. 8
OOD 1	Steel 4140	A	Short	New	A. 9
OOD 2	Steel 4140	B	Short	New	A. 10
OOD 3	AL 6061	B	Short	New	A. 11-13

2.2. Experiment and dataset

Various milling experiments were conducted using the 3-axis mill in order to train and evaluate the model within the monitoring framework. Three sound sensors on different placements were utilized. The milling experimental setup is shown in Fig. 3 (A). In addition to cutting parameters, various tool-related conditions such as tool type, setup, and wear condition were considered. Two types of endmills were used, as shown in Fig. 3 (B): Tool A is ½" diameter 5-flute square endmill (UGDE0500J5BRB, Kennametal), and Tool B is ¼" diameter 2-flute square endmill (2SE0250IX150A, Kennametal). Different tool setups, long versus short, were employed for the cutting experiments, as illustrated in Fig. 3 (C). Various tool conditions, new or moderately worn but functional, were also taken into account, as shown in Fig. 3 (D). The workpieces used in the experiments were Steel 4140, 1018, and Aluminum 6061. 527 different combinations of cutting parameters, tool, workpiece conditions were tested. Details of sensor deployments and experiments are described in Appendix A.1.

Table 1 summarizes the training and testing datasets, using in-distribution datasets for training to cover a wide range of scenarios reflective of typical conditions, and out-of-distribution (OOD) datasets for testing to access the model's generalization to unseen conditions. This approach highlights our focus on evaluating the model's robustness in the variable, complex CNC machining environments. Details on the cutting parameters, essential for contextualizing the model's operation, are provided in Appendix A.2. The last column of Table 1 directs to the appendix for comprehensive cutting parameters and label information. By adopting OOD testing, we aim to demonstrate the model's ability to navigate diverse cutting conditions, ensuring a robust validation of its practical utility and reliability in real-world applications.

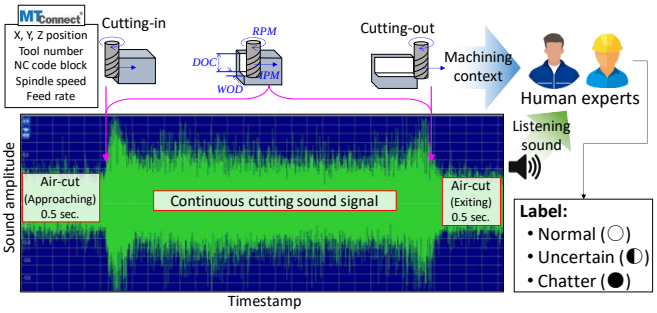


Fig. 4 Chatter labelling adopting expert knowledge

Table 2 Label for machining chatter

Label	Symbol	Definition
Normal	○	It is stable cutting sound.
Uncertain chatter	◐	Questionable. It is not sure whether it is chatter or not, but an operator may listen carefully when this sound is generated to see if cutting is okay.
Chatter	●	An operator should interrupt the operation or change the cutting parameters if sound continues.

2.3. Labeling by expert knowledge

The process of labeling by human experts is depicted in Fig. 4. Following the experiments, the sound data was segmented for each tool path. The timestamps for cutting-in and cutting-out for each tool path were calculated using the machine's MTConnect data. These timestamps were then used to split the sound signals, including a margin of 0.5 seconds before and after the cutting period. Two experienced operators were enlisted to classify each tool path as either 'chatter' or 'normal'. Their assessment was based on listening to the sound recordings and reviewing the cutting parameters (such as tool and workpiece information, Depth of Cut (DOC), Width of Cut (WOC), spindle speed, and feed rate), all of which were accessible via MTConnect. Machining experience of the two operators is summarized as follows:

- Operator 1 has 13 years of experience as a CNC machine operator, specialized in CAM.
- Operator 2 has 15 years of experience as a CNC machine operator and tool maker.

Both operators have rich experience with machining chatter and were engaged in designing experiments to make various chatter conditions. The operators were requested to determine chatter labels as described in Table 2. For the model training, numeric values were assigned to the chatter labels as follows: Chatter (1), Uncertain (0.5), and Normal (0).

Fig. 5 illustrates the spectrogram representation of sound signals in chatter and stable cases using the short-time Fourier transform (STFT). These signals correspond to the case of ID 1, which involves Steel 4140 and Tool A under a Long-Worn condition, with cutting parameters set at a 3000 RPM spindle speed, 60 IPM feed rate, 0.75 in. depth of cut (DOC), and 0.025 in. width of cut (WOC). It's noteworthy that, despite having identical tool and workpiece setups as well as cutting parameters, varying the milling directions (up-mill or down-mill) resulted in differences in cutting stability.

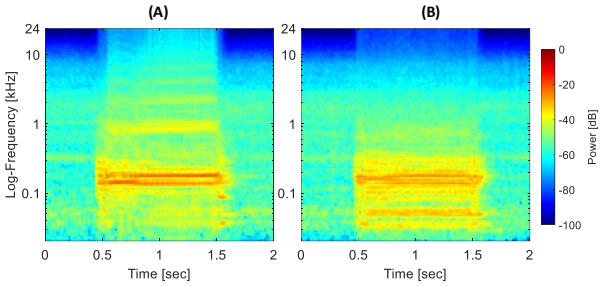


Fig. 5 Sound signal comparison: (A) Chatter (Up-mill) and (B) Stable (Down-mill)

3. Chatter sound classifier

3.1. Data Preprocessing

In the data preprocessing phase, our initial objective is to manage audio files acquired from diverse experiments, each with varying durations. To standardize the data and meet real-time requirements, we segment these audio files into 1-second intervals. This consistent segmentation ensures uniformity in subsequent analytical and processing steps. The Mel-spectrogram was adopted to extract audio features because the purpose of the classifier is to mimic auditory sensing ability of the experts for chatter detection. The Mel scale indicates how humans perceive the frequency of a pure tone compared to its objectively measured frequency. The Mel-spectrogram proves to be a valuable representation in audio data analysis, enabling to capture the spectral characteristics of the audio signals [19-21]. We proceed to convert each 1-second audio segment into a Mel-spectrogram representation. A Mel-spectrogram is configured with dimensions (3, 64, 94), where '3' signifies the number of channels recorded by three sound sensors, '64' represents the predetermined count of Mel filter banks applied during the transformation, and '94' is determined by the length of the audio sample and a predefined window size (n_fft=1024). This makes the data suitable for further processing, feature extraction, and utilization in a wide range of machine learning and audio processing tasks.

3.2. Proposed model architecture

The model architecture was designed to detect chatter using sound signals and machining context, framed as a Binary Classification task. This approach leverages numeric values assigned through human labeling. In the proposed architectural design, a multi-stage approach focused on audio data processing has been built.

At the heart of this framework is the Audio Layer, powered by the well-known AlexNet [22] - a convolutional neural network (CNN) primarily designed for image classification. Notably, AlexNet has demonstrated its effectiveness in transfer learning for Chatter Classification, as shown in paper [23]. Key features of AlexNet include its five CNNs layers, which learn hierarchical image features, ReLU activations for non-linearity, and max-pooling layers for down sampling. To enhance generalization, local response normalization is employed. The classification task is facilitated through three fully connected layers, with dropout used to mitigate overfitting.

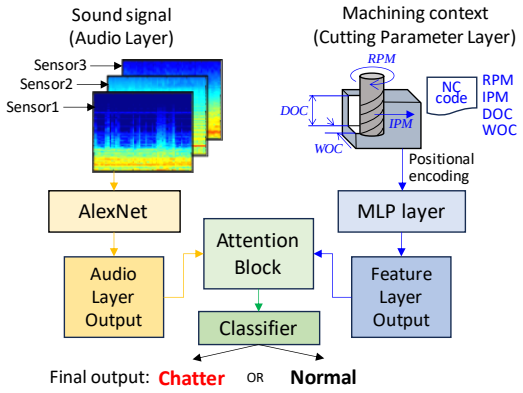


Fig. 6 Proposed architecture for chatter prediction

In our setup, we reinterpret the audio input, already transformed into a three-channel spectrogram, treating it as if it were an RGB image. It's crucial to highlight that we employ a pretrained version of AlexNet, trained on a subset of the ImageNet dataset [24], which comprises 1,281,167 images spanning 1,000 distinct object classes. While our dataset may diverge significantly from the ImageNet classes, the prior training on such a diverse and extensive dataset equips the model with a versatile repertoire of features and hierarchical representations. By combining this pretrained model with the strategy of reinterpreting and fine-tuning only the weights of layers in AlexNet during training on our specific dataset, the fine-tuned AlexNet demonstrates remarkable proficiency in spectrogram classification tasks.

Running in parallel to the Audio Layer, we introduce the Cutting Parameters Layer. The input of this layer is spindle rotational speed in revolution per minute (RPM), feed rate in inch per minute (IPM), depth of cut (DOC), and width of cut (WOC) in inch unit. Here, a combination of Positional Encoding and a Multilayer Perceptron (MLP) serves as the encoding mechanism, facilitating the extraction of vital features from the machining setup, aligning with the audio samples. The MLP layer is composed of a single layer with 10 nodes, an input size of 4 from positional encoding, and employs ReLU activation.

The distinctive feature of our architectural framework resides in the Combination Layer, incorporating an attention mechanism [25]. This mechanism enhances the model's capacity to selectively concentrate on the most pertinent information, facilitating the discernment of intricate patterns within the data. This strategic fusion of features and attention allocation reinforces the overall effectiveness and adaptability of our architecture in managing intricate tasks related to the analysis of complex audio data. The attention layer, a pivotal component in neural network architectures, has been proven to augment focus on specific input data segments by dynamically assigning varying levels of importance to different elements. Widely employed in machine learning, the attention layer improves the model's performance across diverse tasks, including image recognition, natural language processing, and sequence-to-sequence operations [26]. Significantly, in scenarios where specific data segments hold greater relevance, the attention layer enables adaptive concentration on the most crucial information, thereby enhancing learning and decision-

making processes. Additional mathematical details regarding the attention mechanism are available in Appendix B. Consequently, this layer plays a crucial role by assigning weights to and combining features from both the Audio Layer and the Cutting Parameter Layer. The attention mechanism includes three MLP layers, each with 128 nodes.

In the Classifier stage, our architecture incorporates another MLP to predict the likelihood of chatter events, utilizing the comprehensive feature set derived from the Attention layer. This enables the MLP-based classifier to make informed decisions, enhancing the robustness and adaptability of our model, particularly in varied manufacturing setups encountered across different experiments. The Classifier block consists of two dense layers, with the first layer containing 128 nodes and the second layer 64 nodes, both utilizing ReLU activation. Following each dense layer, a dropout layer with a rate of 0.5 is applied to prevent overfitting. To ensure high confidence in identifying "Chatter" and "Normal" conditions, we applied a binary classification threshold of 0.5: predictions equal to or above this threshold are classified as "1" (indicating "Chatter"), while those below are classified as "0" (indicating "Normal"). This careful selection of the threshold aims to minimize the risk of false positives in uncertain scenarios, ensuring that our model's predictions are both reliable and precise.

4. Result and discussion

4.1. Model training setup and dataset splitting

The effectiveness of our proposed model was assessed using a dataset encompassing experiments with varying values across four critical machining parameters: tool type, tool setup, tool condition, and workpiece material. To gauge the model's Out-Of-Distribution (OOD) performance, its robustness against unseen samples was evaluated during training. Consequently, the data splitting procedure deviates from the standard practice of random splitting in a specific ratio.

The training and in-distribution (ID) testing dataset include all possible combinations of the four cutting parameters. However, we intentionally reserve the "Short-New Tool" category, encompassing variable tool condition and workpiece values, for OOD testing. While this category is specifically earmarked for OOD testing, we maintain the flexibility to introduce other manufacturing feature combinations, even if the model has encountered similar combinations during training, into the OOD testing phase when required. A comprehensive breakdown of the dataset and its allocation can be found in Table 1. This strategic approach allows for a comprehensive evaluation of the model's performance across experiments with both known and combinations of cutting parameters, offering significant insights into its ability to generalize.

4.2. Training with the proposed model

To address labeling conflicts in Binary Classification tasks, where two experts have different opinions, a practical solution is to average their labels. The theoretical proof validating this approach will be detailed in Appendix C. This average label can subsequently be utilized in conjunction with the Binary Cross-Entropy loss function to guide model training.

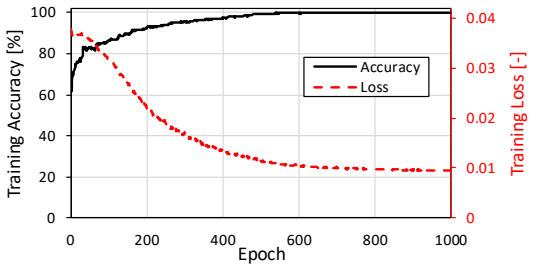


Fig. 7 Training curve of proposed architecture

In the training phase, we primarily utilized the default hyperparameters such as kernel size, dropout and so on associated with the pretrained AlexNet model. This decision was based on the proven effectiveness and general applicability of AlexNet's established configurations across various tasks, allowing us to streamline our fine-tuning process without delving into extensive adjustments of additional hyperparameters. A learning rate of 10^{-6} was chosen, and training was conducted with a batch size of 8. This parameter configuration yielded a training curve that exhibited distinct characteristics. Over the course of training, the curve in Fig. 7 displayed a monotonically decreasing trend, signifying a gradual reduction in the average training loss. After approximately 800 epochs, the curve ultimately converged, indicating that the model had reached a stable and optimized state. This observed convergence suggests that the chosen learning rate and batch size were effective in guiding the training process to minimize the training loss and enhance the model's overall performance.

4.3. Performance evaluation

Evaluation of the trained models was performed exclusively on samples with consistent labels to ensure reliability. The baseline models employed the AlexNet architecture, processing spectrogram data as both a standalone model and in conjunction with the concatenation of cutting parameters, mirroring our architectural setup. However, our approach departed significantly by incorporating an attention block to dynamically combine the outputs from the AlexNet model and cutting parameters. The detailed results of this evaluation are presented in Table 3, showing mean and standard deviation between runs. The proposed method demonstrated superior performance in both ID and OOD testing datasets. Notably, the model using AlexNet alone demonstrated the least effective

Table 3 Overall model prediction accuracy comparison in percentage unit

Case	AlexNet (Standalone)	AlexNet + Concat	Proposed Architecture
Training Accuracy	99.77 ± 0.08	99.85 ± 0.02	**99.85 ± 0.00**
ID Testing	94.33 ± 0.02	95.74 ± 0.01	**98.93 ± 0.35**
OOD Testing	88.66 ± 1.52	93.46 ± 0.01	**94.51 ± 0.03**

Table 4 Individual prediction accuracy comparison in percentage unit

Case	AlexNet (Standalone)	AlexNet + Concat	Proposed Architecture
OOD 1 Testing	78.95 ± 0.04	88.60 ± 0.06	**95.61 ± 0.03**
OOD 2 Testing	90.83 ± 0.01	91.67 ± 0.02	**93.33 ± 0.00**
OOD 3 Testing	90.00 ± 0.03	**96.33 ± 0.01**	96.00 ± 0.01
OOD 4 Testing	90.68 ± 0.01	93.17 ± 0.01	**93.17 ± 0.01**

performance across the three datasets, suggesting that the contribution of cutting parameters has a more significant impact on the prediction. All three models exhibited robust training, consistently achieving accuracy levels above 99%. However, in ID testing, the standalone AlexNet model experienced a drop to 94.33%, and the concatenation AlexNet dropped to 95.74%, while our model maintained the highest accuracy at 98.93%. The model continues to perform well on the training and in-distribution datasets, indicating that the pretrained model with fine-tuning techniques is sufficient for capturing the correlations between the three spectrograms. Regarding OOD testing Accuracy, our model dropped to 94.51% but remained superior to the other models. Table 1 highlights the efficacy of our approach in addressing labeling challenges and enhancing the model's generalization across diverse data distributions, especially when encountering samples with cutting parameters not seen during the training process.

In addition, we conducted a detailed analysis of the performance of our model and the two baseline models, focusing specifically on the OOD testing dataset for each case individually. The outcomes are detailed in Table 4. The results clearly demonstrate that our architecture consistently maintains a high level of performance across different experiments, consistently exceeding 93% accuracy, even when dealing with previously unseen manufacturing features. In contrast, there is notable variability in the performance of AlexNet and AlexNet combined with manufacturing features.

4.4. Real-time implementation

The trained model, when deployed on the Raspberry Pi using TensorFlow Lite, maintained prediction accuracies closely aligned with the original expectations, demonstrating 97.16% for ID dataset and 92.76% for OOD dataset. Despite experiencing a slight decrease in accuracy compared to the original model by approximately 1.8% for ID and 1.7% for OOD, the model's performance within the TensorFlow Lite environment showed a commendable level of predictive capability. Additionally, the model's operational efficiency on the Raspberry Pi, with signal processing and model computing times measured at approximately 68±6 milliseconds and 85±10 milliseconds respectively, not only highlights the practicality of deploying our model on edge computing devices but also emphasizes its suitability for real-time applications. This ensures that the continuous chatter prediction output is promptly converted to MTConnect data item 'Cutting condition', which is classified as either 'Normal' or 'Chatter'. It makes condition warning when chatter happens in the monitoring system. To facilitate the real-time visualization of the CNC machining operations and chatter sound prediction, an integrative approach was employed. The data flow for the real-time chatter prediction and visualization is illustrated in Fig. 8.

MTConnect of the machine is utilized to enable web-based visualization. In tandem, React.js was utilized for its modular architecture and its proficiency in rendering dynamic user interfaces. Furthermore, THREE.js library within the React.js ecosystem was incorporated to imbue our application with three-dimensional visualization capabilities.

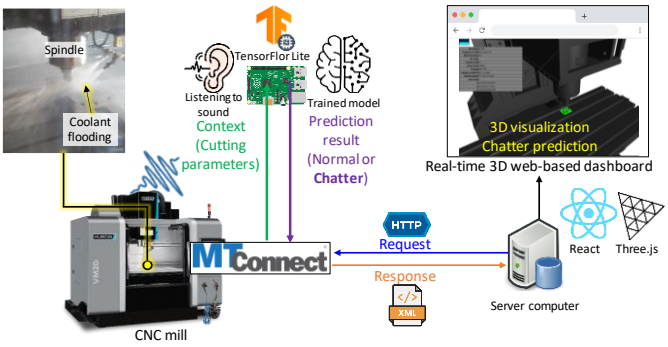


Fig. 8 Data flow of real-time chatter detection

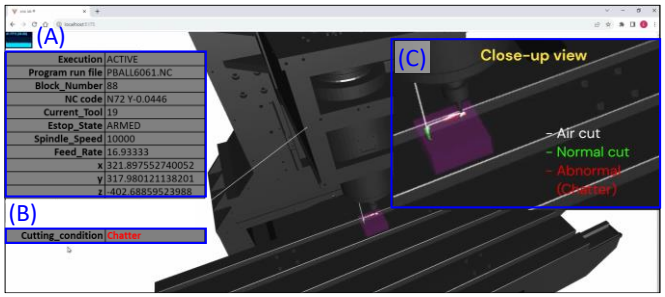


Fig. 9 Capture of web-based cutting monitoring dashboard: (A) MTConnect data, (B) Chatter prediction model result, and (C) Close-up view near tool

The system architecture is demarcated into three distinct phases: initialization, continuous data acquisition and analysis, and the final rendering stage. During the initial phase, the system loads a bespoke configuration file for the machine. This file is pivotal as it delineates critical details about the machine components, including the storage paths for their 3D models, initial positions, motion constraints, and transformation algorithms. Subsequent to this, we establish a 3D scene that is populated with the machine's components, each rendered with appropriate materials and positioned accurately using specialized algorithms that reflect their unique functions within the machine assembly.

The phase dedicated to data acquisition and analysis operates cyclically, refreshing the 3D visualization in real-time. It harnesses the HTTP protocol to fetch positional data from CNC machine via MTConnect. Following the retrieval of data, the system parses this information to discern vital parameters, such as the current positions of machine components, the active NC code, and the detection of any operational anomalies such as chatter. The updated positions of the components are then calculated through their respective transformation equations. Concurrently, additional data are systematically organized into a tabular format for display alongside the 3D visualization.

In the rendering phase, the application concentrates on the user interface, displaying the 3D model of the CNC machine in a canvas element. Additionally, it presents detailed tables that materialize conditionally based on the data procured in the analysis phase. These tables offer a granular view of the CNC machine's performance metrics, enriching the visual representation with precise and actionable data. The web-based real-time 3D dashboard to monitor cutting is in Fig. 9. The real-time MTConnect data is shown on a table in the left top side of the dashboard (Fig. 9 (A)) and the chatter prediction result from sound is shown on the left middle side (Fig. 9 (B)). By utilizing

color coding of each tool path with the model prediction result, the tool path which makes chatter is shown in red color as shown in the close-up view of the cutting in Fig. 9 (C).

5. Conclusion

This study introduced a comprehensive framework that employs a deep learning architecture, integrating the AlexNet model and machining context, for the detection of chatter in milling processes. Central to our approach is the digitization of machine tool and sound data by employing MTConnect, coupled with expert auditory judgments, to facilitate accurate event labeling. This research demonstrates the potent application of deep learning in achieving precise chatter detection, significantly enhancing both accuracy and reliability. The strategic combination of human expert insights on milling tool chatter sounds with our CNN-based model, which uses AlexNet as a comparative baseline, signifies an advancement in fusing traditional machining expertise with contemporary AI technologies.

The architecture proposed in the paper strategically integrated an attention block, combining outputs from both the AlexNet model and cutting parameters. This design choice resulted in improved performance compared to baseline models, seen in both in-distribution and out-of-distribution testing datasets. In out-of-distribution testing, the proposed model achieved an accuracy of 94.51%, surpassing the standalone CNN model's 88.66%. This emphasizes the robust generation capability of the proposed architecture, especially with samples featuring cutting setups and conditions unseen during the training. Moreover, the architecture's ability to process inputs with a mere 1-second duration renders it highly suitable for real-world applications, enabling the timely detection of chatter before human intervention. The practical application of this research was evidenced by the successful implementation of the trained model on a Raspberry Pi for real-time 3D visualization of machining processes. Consequently, this research not only highlights the model's accuracy but also emphasizes its practical feasibility, showcasing its potential to significantly enhance machining chatter monitoring.

Future research will not only aim to expand the monitoring to include tool condition monitoring and machining failures but also address the challenge of human bias in the labeling process. We plan to explore and implement measures to enhance the accuracy and consistency of true and false labels provided by human experts, thereby mitigating the impact of subjective judgment on dataset and findings.

Acknowledgements

This research was supported by Wabash Heartland Innovation Network (WHIN) and Indiana Next Generation Manufacturing Competitiveness Center (IN-MaC). This work is also funded by NSF Grant No. 2134667 "FMRG: Manufacturing USA: Cyber: Privacy-Preserving Tiny Machine Learning Edge Analytics to Enable AI-Commons for Secure Manufacturing," and AnalytiXIN through CICP. The authors appreciate the assistance provided by Jacob Coffing of Coffing CNC in the experimental design and chatter labeling.

Appendix A. Experimental details

A.1. Sensor deployment and tool path

Three sound sensors were deployed to the 3-axis CNC milling machine (Hurco, VM20i). Fig. A. 1 shows the locations of sound sensors. Sensor 1 is a USB microphone (Fifine technology, K053) located inside the machine at the corner of the working area (Fig. A. 1 (B)). The distance between spindle and Sensor 1 is approximately 6 feet. Sensor 2 and 3 are the internal sound sensor which consists of a stethoscope (3M, Littmann III) and that the USB microphone the same as Sensor 1. The configuration of the internal sound sensor is illustrated in Fig. A. 2. The microphone is inserted in the rubber hose of the stethoscope to capture sound from the stethoscope. Silicon gasket was applied around rim of the stethoscope head and then firmed attached using silicon tape when the internal sound sensors were deployed. Sensor 2 is located near the spindle, and Sensor 3 is on the base column, as shown in Fig. A. 1 (C) and (D), respectively.

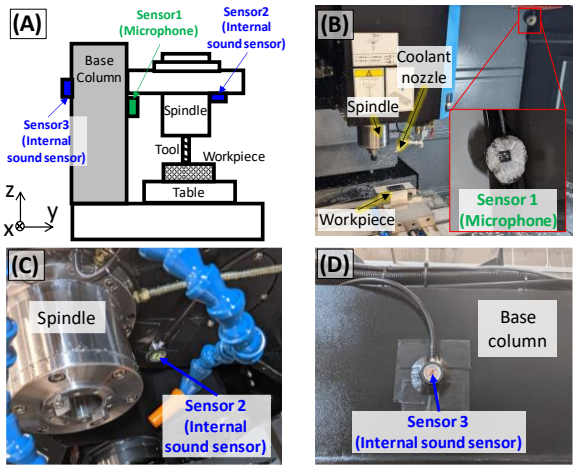


Fig. A. 1 Location of sound sensor: (A) Schematic of machine and sensor configuration, (B) Sensor 1 (Microphone) inside machine, (C) Sensor 2 (internal sound sensor) near spindle, and (D) Sensor 3 (internal sound sensor) on base column

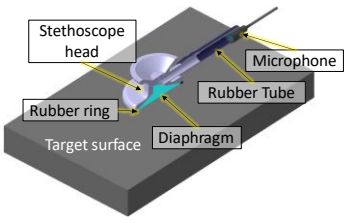


Fig. A. 2 Configuration and crosscut of internal sound sensor in 3D CAD

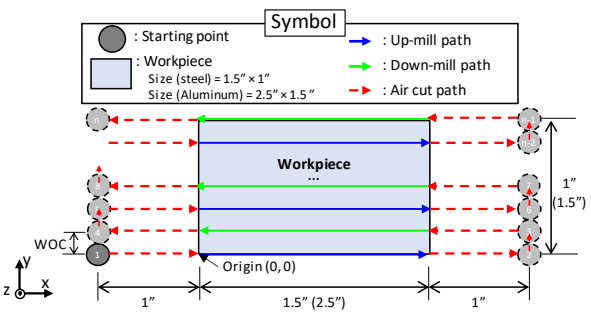


Fig. A. 3 Top schematic view of tool path for cutting experiment

The zig-zag cutting tool path was selected for all experiments to have both up- and down-milling. Top view of the schematic of cutting path is illustrated in Fig. A. 3. The size of the workpiece in the x and y direction was 1.5×1 inches for steel and 2.5×1.5 inches for aluminum. The spindle rotational direction was clockwise in Fig. A. 3. Work origin (e.g., G54) was set at the corner where the x and y directions are negative. To facilitate an air-cut before and after each tool path, a 1-inch distance margin was set in the x positive and negative directions. The feed rate for the air cut was set to match the cutting feed rate. Additionally, a 1-second pause was implemented in every single path. Cutting parameters, including spindle speed, feed rate, width of cut, and depth of cut, were repeated once in both up-milling and down milling throughout the experiments. The exact timestamp when the cutting is engaged in a specific NC code line is determined based on the work origin and positional information of the 3-axis via MTConnect, allowing for the retrieval of cutting parameters after performing the experiments.

A.2. Dataset

Cutting parameters and corresponding chatter labels, as determined by experts according to the datasets, are summarized in the following tables. 'ID' indicates in-distribution data used for training the model, while 'OOD' denotes out-of-distribution data used for testing the model. The label symbols are as follows: ○ for Normal (0), ◐ for Uncertain chatter (0.5), and ● for Chatter (2). For the ID1 dataset, experiments were conducted with all combinations of three DOCs – 1, 0.5, and 0.25 inches and three WOCs – 0.025, 0.02, and 0.015 inches. Apart from Table A. 2, eight additional tables are not included in this manuscript due to their excessive length.

Table A. 1 ID 1 (Steel 4140, Tool A:Long-Worn condition) cutting parameters; DOC=1 in., WOC=0.05 in.

Spindle speed (RPM)	Feed rate (IPM)	Cutting direction	Label	
			Expert 1	Expert 2
5625	50	Up	○	○
5625	50	Down	○	○
6750	50	Up	●	●
6750	50	Down	○	○

Table A. 2 ID 1 (Steel 4140, Tool A:Long-Worn condition) cutting parameters; DOC=1 in., WOC=0.025 in.

Spindle speed (RPM)	Feed rate (IPM)	Cutting direction	Label	
			Expert1	Expert 2
3000	60	Up	○	○
3000	60	Down	○	○
3000	90	Up	○	○
3000	90	Down	○	○
3000	120	Up	○	○
3000	120	Down	○	○
3000	150	Up	○	○
3000	150	Down	○	○
5000	60	Up	◐	◐
5000	60	Down	○	○
5000	90	Up	●	●
5000	90	Down	○	○
5000	120	Up	○	○
5000	120	Down	○	○
5000	150	Up	◐	○
5000	150	Down	○	○
7000	60	Up	●	●
7000	60	Down	○	○
7000	90	Up	●	●
7000	90	Down	○	○
7000	120	Up	●	●
7000	120	Down	○	○
7000	150	Up	●	●
7000	150	Down	○	○
9000	60	Up	●	●
9000	60	Down	○	○
9000	90	Up	●	●
9000	90	Down	○	○
9000	120	Up	●	●
9000	120	Down	○	○
9000	150	Up	●	●
9000	150	Down	○	○

Table A. 3 ID 2 (Steel 4140, Tool A:Long-New condition) cutting parameters; DOC=1 in., WOC=0.025 in.

Spindle speed (RPM)	Feed rate (IPM)	Cutting direction	Label	
			Expert 1	Expert 2
3000	60	Up	○	○
3000	60	Down	○	○
3000	90	Up	○	○
3000	90	Down	○	○
3000	120	Up	○	○
3000	120	Down	○	○
3000	150	Up	○	○
3000	150	Down	○	○
5000	60	Up	◐	○
5000	60	Down	○	○
5000	90	Up	◐	○
5000	90	Down	○	○
5000	120	Up	○	○
5000	120	Down	○	○
5000	150	Up	○	○
5000	150	Down	○	○
7000	60	Up	◐	◐
7000	60	Down	○	○
7000	90	Up	◐	◐
7000	90	Down	○	○
7000	120	Up	●	●
7000	120	Down	○	○
7000	150	Up	●	●
7000	150	Down	○	○
9000	60	Up	●	●
9000	60	Down	○	○
9000	90	Up	●	●
9000	90	Down	○	○
9000	120	Up	●	●
9000	120	Down	○	○
9000	150	Up	●	●
9000	150	Down	○	○

Table A. 4 ID 3 (Steel 1018, Tool A:Long-Worn condition) cutting parameters; DOC=1 in., WOC=0.02 in.

Spindle speed (RPM)	Feed rate (IPM)	Cutting direction	Label	
			Expert 1	Expert 2
5500	60	Up	◐	◐
5500	60	Down	○	○
5500	90	Up	●	◐
5500	90	Down	○	○
5500	120	Up	●	●
5500	120	Down	○	○
5500	150	Up	●	●
5500	150	Down	○	○
7500	60	Up	●	●
7500	60	Down	○	○
7500	90	Up	●	●
7500	90	Down	○	○
7500	120	Up	●	◐
7500	120	Down	○	○
7500	150	Up	●	●
7500	150	Down	○	○

9500	60	Up	●	●
9500	60	Down	○	○
9500	90	Up	●	●
9500	90	Down	○	○
9500	120	Up	●	●
9500	120	Down	○	○
9500	150	Up	●	●
9500	150	Down	○	○
11500	60	Up	◐	◐
11500	60	Down	○	○
11500	90	Up	●	●
11500	90	Down	○	○
11500	120	Up	●	●
11500	120	Down	○	○
11500	150	Up	●	●
11500	150	Down	○	○

Table A. 5 ID 4 (Aluminum 6061, Tool B:Long-Worn condition) cutting parameters; DOC=0.5 in., WOC=0.02 in.

Spindle speed (RPM)	Feed rate (IPM)	Cutting direction	Label	
			Expert 1	Expert 2
8000	40	Up	●	●
8000	40	Down	●	●
8000	50	Up	●	●
8000	50	Down	●	●
8000	60	Up	●	●
8000	60	Down	●	●
10000	40	Up	●	●
10000	40	Down	●	●
10000	50	Up	●	●
10000	50	Down	●	●
10000	60	Up	●	●
10000	60	Down	●	●
12000	40	Up	●	●
12000	40	Down	●	●
12000	50	Up	●	●
12000	50	Down	●	●
12000	60	Up	●	●
12000	60	Down	●	●

Table A. 6 ID 4 (Aluminum 6061, Tool B:Long-Worn condition) cutting parameters; DOC=0.5 in., WOC=0.025 in.

Spindle speed (RPM)	Feed rate (IPM)	Cutting direction	Label	
			Expert 1	Expert 2
8000	40	Up	●	●
8000	40	Down	●	●
8000	50	Up	●	●
8000	50	Down	●	●
8000	60	Up	●	●
8000	60	Down	●	●
10000	40	Up	●	●
10000	40	Down	●	●
10000	50	Up	●	●
10000	50	Down	●	●
10000	60	Up	●	●
10000	60	Down	●	●
12000	40	Up	●	●
12000	40	Down	●	●
12000	50	Up	●	●
12000	50	Down	●	●
12000	60	Up	●	●
12000	60	Down	●	●

Table A. 7 ID 4 (Aluminum 6061, Tool B:Long-Worn condition) cutting parameters; DOC=0.5 in., WOC=0.03 in.

Spindle speed (RPM)	Feed rate (IPM)	Cutting direction	Label	
			Expert 1	Expert 2
8000	40	Up	●	●
8000	40	Down	●	●
8000	50	Up	●	●
8000	50	Down	●	●
8000	60	Up	●	●
8000	60	Down	●	●
10000	40	Up	●	●
10000	40	Down	●	●
10000	50	Up	●	●
10000	50	Down	●	●
10000	60	Up	●	●
10000	60	Down	●	●
12000	40	Up	●	●
12000	40	Down	●	●
12000	50	Up	●	●
12000	50	Down	●	●
12000	60	Up	●	●
12000	60	Down	●	●

Table A. 8 OOD 1 (Steel 4140, Tool A:Long-Worn condition) cutting parameters; DOC=1 in., WOC=0.05 in.

Spindle speed (RPM)	Feed rate (IPM)	Cutting direction	Label	
			Expert 1	Expert 2
4500	50	Up	○	○
4500	50	Down	○	○

Table A. 9 OOD 2 (Steel 4140, Tool A:Short-New condition) cutting parameters; DOC=1 in., WOC=0.025 in.

Spindle speed (RPM)	Feed rate (IPM)	Cutting direction	Label	
			Expert 1	Expert 2
3000	60	Up	○	○
3000	60	Down	○	○
3000	90	Up	○	○
3000	90	Down	○	○
3000	120	Up	○	○
3000	120	Down	○	○
3000	150	Up	○	○
3000	150	Down	○	○
5000	60	Up	○	○
5000	60	Down	○	○
5000	90	Up	○	○
5000	90	Down	○	○
5000	120	Up	○	○
5000	120	Down	○	○
5000	150	Up	○	○
5000	150	Down	○	○
7000	60	Up	◐	◐
7000	60	Down	○	○
7000	90	Up	◐	◐
7000	90	Down	○	○
7000	120	Up	◐	●
7000	120	Down	○	○
7000	150	Up	◐	●
7000	150	Down	○	○
9000	60	Up	●	●
9000	60	Down	○	○
9000	90	Up	●	●
9000	90	Down	○	○
9000	120	Up	●	●
9000	120	Down	○	○
9000	150	Up	●	●
9000	150	Down	○	○

Table A. 10 OOD 3 (Steel 4140, Tool B:Short-New condition) cutting parameters; DOC=0.5 in., WOC=0.01 in.

Spindle speed (RPM)	Feed rate (IPM)	Cutting direction	Label	
			Expert 1	Expert 2
4000	10	Up	●	●
4000	10	Down	●	●
4000	15	Up	●	●
4000	15	Down	○	◐
4000	20	Up	●	●
4000	20	Down	○	◐
4000	25	Up	●	●
4000	25	Down	○	◐
5000	10	Up	●	●

5000	10	Down	●	◐
5000	15	Up	●	●
5000	15	Down	◐	◐
5000	20	Up	●	●
5000	20	Down	○	◐
5000	25	Up	●	●
5000	25	Down	○	○
6000	10	Up	●	●
6000	10	Down	●	●
6000	15	Up	●	●
6000	15	Down	◐	◐
6000	20	Up	●	●
6000	20	Down	○	○
6000	25	Up	●	●
6000	25	Down	○	◐
7000	10	Up	●	●
7000	10	Down	●	●
7000	15	Up	●	●
7000	15	Down	◐	○
7000	20	Up	●	●
7000	20	Down	◐	◐
7000	25	Up	●	●
7000	25	Down	○	◐

Table A. 11 OOD 4 (Aluminum 6061, Tool B:Short-New condition) cutting parameters; DOC=0.5 in., WOC=0.02 in.

Spindle speed (RPM)	Feed rate (IPM)	Cutting direction	Label	
			Expert 1	Expert 2
8000	40	Up	●	●
8000	40	Down	●	◐
8000	50	Up	●	●
8000	50	Down	●	◐
8000	60	Up	●	●
8000	60	Down	○	○
10000	40	Up	●	●
10000	40	Down	●	●
10000	50	Up	●	●
10000	50	Down	●	●
10000	60	Up	●	●
10000	60	Down	○	○
12000	40	Up	●	●
12000	40	Down	◐	●
12000	50	Up	●	●
12000	50	Down	◐	●
12000	60	Up	●	●
12000	60	Down	○	○

Table A. 12 OOD 4 (Aluminum 6061, Tool B:Short-New condition) cutting parameters; DOC=0.5 in., WOC=0.025 in.

Spindle speed (RPM)	Feed rate (IPM)	Cutting direction	Label	
			Expert 1	Expert 2
8000	40	Up	●	●
8000	40	Down	●	●
8000	50	Up	●	●
8000	50	Down	●	●
8000	60	Up	●	●
8000	60	Down	○	○
10000	40	Up	●	●
10000	40	Down	●	●
10000	50	Up	●	●
10000	50	Down	◐	◐
10000	60	Up	●	●
10000	60	Down	●	●
12000	40	Up	●	●
12000	40	Down	◐	●
12000	50	Up	●	●
12000	50	Down	◐	◐
12000	60	Up	●	●
12000	60	Down	●	●

Table A. 13 OOD 4 (Aluminum 6061, Tool B:Short-New condition) cutting parameters; DOC=0.5 in., WOC=0.03 in.

Spindle speed (RPM)	Feed rate (IPM)	Cutting direction	Label	
			Expert 1	Expert 2
8000	40	Up	●	●
8000	40	Down	●	●
8000	50	Up	●	●
8000	50	Down	●	●
8000	60	Up	●	●
8000	60	Down	◐	●
10000	40	Up	●	●
10000	40	Down	●	●
10000	50	Up	●	●
10000	50	Down	●	●
10000	60	Up	●	●
10000	60	Down	○	◐
12000	40	Up	●	●
12000	40	Down	◐	◐
12000	50	Up	●	●
12000	50	Down	◐	●
12000	60	Up	●	●
12000	60	Down	●	●

Appendix B. More details about the attention mechanism

The attention mechanism encompasses a series of mathematical operations aimed at computing attention scores and generating a context vector. In essence, it revolves around an input sequence and a query vector that encapsulates the specific information of interest to the model. The attention mechanism operates through two fundamental steps:

1. Score Calculation:
   - Each element within the input sequence is associated a key vector $\boldsymbol{K}$. Attention scores are then computed by evaluating the similarity between the query vector ($\boldsymbol{Q}$) and each key vector. Typically, this similarity is gauged using a scoring function such as the dot product.
   - The attention scores are obtained by applying the softmax function to the computed similarities. This ensures normalization, with the sum of scores equating to 1. Consequently, each element in the sequence is assigned a weight that reflects its relevance to the query. Notably, these weights are adaptable parameters, enabling the model to refine its attention mechanism during training.
   - Mathematically, the attention score can be expressed as

$$\text{attention score} = \text{softmax}(\boldsymbol{W}_k \boldsymbol{K}^T \boldsymbol{W}_q \boldsymbol{Q})$$

   where $\boldsymbol{W}_k$ and $\boldsymbol{W}_q$ denote the weights of dense layers applied to the key and query vectors, respectively.

2. Weighted Sum:
   - The obtained attention scores serve as weights, determining the significance of each element in the input sequence ($\boldsymbol{V}$). Subsequently, the weighted sum of the input sequence is computed, with each element multiplied by its corresponding attention score. Its formula is $\boldsymbol{W}_v \boldsymbol{V}$ attention score with $\boldsymbol{W}_v$ denote the weights of dense layers applied to $\boldsymbol{V}$. This process yields a context vector that accentuates elements in the input sequence based on their relevance to the provided query. Consequently, the context vector encapsulates the most vital information pertinent in the task at hand.

The attention mechanism, a cornerstone of machine learning, facilitates dynamic focus on different segments of the input sequence in response to specific queries. This capability significantly enhances model performance across various tasks by enabling targeted information processing.

Appendix C. Validation of averaging approach for expert opinions

Considering the *i*-th sample in *n* samples, we have the predicted output x_i and $\bar{y}_i = {}^{(y_{i_1}+y_{i_2})}\!/_{2}$, where $y_{i_1}, y_{i_2} \in [0,1]$ and y_{i_1}, y_{i_2} represent the opinions from two experts. The proof remains applicable for cases involving more than two experts.

Binary Cross Entropy Loss is computed with the given x_i and $\bar{y}_i$, where $X_i \sim P(x \mid \bar{y}_i)$ and $i = \{1, 2, ..., n\}$, where n represents the number of samples.

$$
\begin{aligned}
L &= \frac{1}{n}\sum_{i=1}^{n} L_i \\
&= \frac{1}{n}\sum_{i=1}^{n}\left[\bar{y}_i \log x_i + (1-\bar{y}_i)\log(1-x_i)\right] \\
&= \frac{1}{n}\sum_{i=1}^{n}\left[\left(\frac{y_{i_1}+y_{i_2}}{2}\right)\log x_i + \left(1-\frac{y_{i_1}+y_{i_2}}{2}\right)\log(1-x_i)\right] \\
&= \frac{1}{n}\sum_{i=1}^{n}\left[\frac{y_{i_1}}{2}\log x_i + \frac{y_{i_2}}{2}\log x_i + \left(1-\frac{y_{i_1}}{2}-\frac{y_{i_2}}{2}\right)\log(1-x_i)\right] \\
&= \frac{1}{n}\sum_{i=1}^{n}\left[\frac{y_{i_1}}{2}\log x_i + \frac{y_{i_2}}{2}\log x_i + \log(1-x_i) - \frac{y_{i_1}}{2}\log(1-x_i) - \frac{y_{i_2}}{2}\log(1-x_i)\right] \\
&= \underbrace{\frac{1}{2n}\sum_{i=1}^{n}\left[y_{i_1}\log x_i + (1-y_{i_1})\log(1-x_i)\right]}_{\frac{1}{2}\text{Binary Cross Entropy}\left(x_i, y_{i_1}\right)} + \underbrace{\frac{1}{2n}\sum_{i=1}^{n}\left[y_{i_2}\log x_i + (1-y_{i_2})\log(1-x_i)\right]}_{\frac{1}{2}\text{Binary Cross Entropy}\left(x_i, y_{i_2}\right)}
\end{aligned}
$$

$\Rightarrow$ The loss of average approach $\frac{1}{2}\text{Binary Cross Entropy}\left(x_i, \bar{y}_i\right)$

The loss of average approach is the loss of treat y_{i_1}, y_{i_2} separately. Thus, the approach is valid.

References

[1] B. Chen, J. Wan, L. Shu, P. Li, M. Mukherjee, B. Yin, Smart Factory of Industry 4.0: Key Technologies, Application Case, and Challenges, IEEE Access 2018;6;6505-6519.

[2] W. Chen, Intelligent manufacturing production line data monitoring system for industrial internet of things, Computer Communications 2020;151;31-41.

[3] G. Quintana, J. Ciurana, Chatter in machining processes: A review, International Journal of Machine Tools and Manufacture 2011;51;363-376.

[4] E. Graham, M. Mehrpouya, S.S. Park, Robust prediction of chatter stability in milling based on the analytical chatter stability, Journal of Manufacturing Processes 2013;15;508-517.

[5] E. Budak, L.T. Tunç, S. Alan, H.N. Özgüven, Prediction of workpiece dynamics and its effects on chatter stability in milling, CIRP Annals 2012;61;339-342.

[6] Z. Wang, Y. Yang, Y. Liu, K. Liu, Y. Wu, Prediction of time-varying chatter stability: effect of tool wear, The International Journal of Advanced Manufacturing Technology 2018;99;2705-2716.

[7] K. Weinert, T. Surmann, D. Enk, O. Webber, The effect of runout on the milling tool vibration and surface quality, Production Engineering 2007;1;265-270.

[8] P.V. Bayly, B.P. Mann, T.L. Schmitz, D.A. Peters, G. Stepan, T. Insperger, Effects of radial immersion and cutting direction on chatter instability in end-milling, in: ASME International Mechanical Engineering Congress and Exposition; 2002, p. 351-363.

[9] E. Budak, A. Tekeli, Maximizing Chatter Free Material Removal Rate in Milling through Optimal Selection of Axial and Radial Depth of Cut Pairs, CIRP Annals 2005;54;353-356.

[10] T. Insperger, B.P. Mann, G. Stépán, P.V. Bayly, Stability of up-milling and down-milling, part 1: alternative analytical methods, International journal of Machine tools and manufacture 2003;43;25-34.

[11] R.G. Landers, A.G. Ulsoy, Nonlinear feed effect in machining chatter analysis, 2008.

[12] Y. Roh, G. Heo, S.E. Whang, A Survey on Data Collection for Machine Learning: A Big Data - AI Integration Perspective, IEEE Transactions on Knowledge and Data Engineering 2021;33;1328-1347.

[13] I. Kvinevskiy, S. Bedi, S. Mann, Detecting machine chatter using audio data and machine learning, The International Journal of Advanced Manufacturing Technology 2020;108;3707-3716.

[14] M.Q. Tran, M.K. Liu, M. Elsisi, Effective multi-sensor data fusion for chatter detection in milling process, ISA Trans 2022;125;514-527.

[15] B. Sener, M.U. Gudelek, A.M. Ozbayoglu, H.O. Unver, A novel chatter detection method for milling using deep convolution neural networks, Measurement 2021;182.

[16] M.H. Rahimi, H.N. Huynh, Y. Altintas, On-line chatter detection in milling with hybrid machine learning and physics-based model, CIRP Journal of Manufacturing Science and Technology 2021;35;25-40.

[17] X. Yan, S. Melkote, A.K. Mishra, S. Rajagopalan, A digital apprentice for chatter detection in machining via human–machine interaction, Journal of Intelligent Manufacturing 2022;34;3039-3052.

[18] M. Standard, ANSI/MTC1. 4-2018, M. Institute, 2018, in.

[19] E. Kim, H. Yun, O.C. Araujo, M.B.G. Jun, Sound Recognition based on Convolutional Neural Network for Real-Time Cutting State Monitoring of Tube Cutting Machine, International Journal of Precision Engineering and Manufacturing-Smart Technology 2023;1;1-18.

[20] J. Li, W. Dai, F. Metze, S. Qu, S. Das, A comparison of deep learning methods for environmental sound detection, in: 2017 IEEE International conference on acoustics, speech and signal processing (ICASSP), IEEE; 2017, p. 126-130.

[21] J. Shen, R. Pang, R.J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, Natural tts synthesis by conditioning wavenet on mel spectrogram predictions, in: 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP), IEEE; 2018, p. 4779-4783.

[22] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Advances in neural information processing systems 2012;25.

[23] H.O. Unver, B. Sener, A novel transfer learning framework for chatter detection using convolutional neural networks, Journal of Intelligent Manufacturing 2021;34;1105-1124.

[24] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE conference on computer vision and pattern recognition, Ieee; 2009, p. 248-255.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 2017;30.

[26] Z. Niu, G. Zhong, H. Yu, A review on the attention mechanism of deep learning, Neurocomputing 2021;452;48-62

.