Learning machining stability diagrams from data using neural networks

Namras Amakkattil Shanavas, Mohit Law*, Manjesh K. Singh

Indian Institute of Technology Kanpur, Kanpur, India

Presented in International Conference on Precision, Micro, Meso and Nano Engineering (COPEN - 12: 2022)
December 8th - 10th, 2022 IIT Kanpur, India

KEYWORDS

Machining,
Stability,
Chatter,
Machine Learning,
Neural Network,
Hyperparameters.

ABSTRACT

Machining instabilities are detrimental. model predicted stability charts help identify cutting parameters for stability. Since models disregard speed-varying cutting force characteristics and dynamics, charts fail to guide stable cutting in industrial praxis. This study shows how supervised neural networks can learn stability charts from data. The learning capacity of this machine learning model depends on the size of the training dataset, its train-test split, the learning rate, the activation function, the number of hidden layers, and the number of neurons in each layer. This is the first study to examine how hyperparameters influence learning machining stability diagrams. Learnings from a linear stability dataset are transferrable to nonlinear datasets, demonstrating the prediction model is physics-agnostic. Predictions accuracies of up to 97.2% were obtained. Since the data used to train the model includes all the vagaries and uncertainties of the cutting process, the results can inform self-optimizing and autonomous machining systems.

1. Introduction

Machines vibrate under the action of cutting forces. Sometimes these vibrations grow to result in chatter instabilities that can damage the part, the tool, and elements of the machine. Instabilities must hence be avoided. Analytical model-predicted stability charts can guide selection of cutting parameters to ensure stable processes (Altintas et al., 2020). Inputs to these models are the measured dynamics of the machine, the cutting force coefficients, and cutting strategies. Though these model-predicted stability charts have proved useful, since models seldom account for how coefficients change with speed and/or for how dynamics change with speed and position of the tool in the machine's work volume, and since models make several linearizing assumptions, charts often fail to guide stable cutting in industrial praxis (Munoa et al., 2016). Since instabilities are detrimental, this paper will demonstrate learning the stability chart from experimental data using supervised machine learning (ML) models. Data that the model will use to learn stability will be cutting conditions at which instabilities are detected during machining. Since monitoring the machine's states to detect chatter is possible with modern transducers and data acquisition systems, and since measured data includes in it all the vagaries and uncertainties associated with the cutting process, the learnt stability is expected to be more accurate, and hence more useful in guiding selection of stable cutting processes.

Use of ML in the context of machine tools is not new. Research has successfully demonstrated the utility of learning from data to monitor the condition of tools and machines (Möhring et al., 2020; Aggogeri et al., 2021), to identify trends in process forces (Vaishnav et al., 2020), to monitor and compensate thermal errors (Reddy et al., 2020; Liu & Du, 2021), to identify dynamics (Postel et al., 2020b; Liu & Altintas, 2021), to detect chatter (Tarng & Chen, 1994; Shi & Cao et al., 2020; Yesilli et al., 2020; Shi, Cao, Zhang & Chen, 2020; Kvinevskiy et al., 2020; Rahimi et al., 2021; Cornelius et al., 2021; Unver & Sener, 2021; Wang et al., 2021), and in the context of the topic of interest of this paper, i.e., to learn the stability chart (Friedrich et al., 2017; Friedrich et al., 2018; Saadallah et al., 2018; Denkana et al., 2020; Postel et al., 2020a, 2020b; Cherukuri et al., 2019; Karandikar et al., 2020; Chen et al., 2021; Bergmann & Reimer, 2021).

Prior research on learning stability using ML models has discussed the use of artificial neural

*Corresponding author E-mail: mlaw@iitk.ac.in

https://doi.org/10.58368/MTT.22.2.2023.29-41

networks (ANN) (Friedrich et al., 2017; Friedrich et al., 2018; Saadallah et al., 2018; Denkana et al., 2020; Postel et al., 2020a, 2020b; Cherukuri et al., 2019), support vector machines (SVM) (Friedrich et al., 2017; Friedrich et al., 2018; Denkana et al., 2020), k-nearest neighborhood (kNN) (Denkana et al., 2020), and Bayesian methods (Karandikar et al., 2020; Chen et al., 2021). Of these, some ML models can work with less data and hence can be trained fast, whereas others are data hungry, and hence are slower to train. Since most ML models become better at prediction with more and better-quality data, even if speed is traded for accuracy, since chatter has destructive potential, models that are accurate should be preferred even if they are slower. Since ANN models predict with good accuracy even though they generally require larger amounts of data, they have been preferred to learn stability (Friedrich et al., 2017; Friedrich et al., 2018; Saadallah et al., 2018; Denkana et al., 2020; Postel et al., 2020a, 2020b; Cherukuri et al., 2019), to detect chatter (Kvinevskiy et al., 2020; Rahimi et al., 2021), to monitor process forces (Vaishnav et al., 2020), and to monitor tool condition (Aggogeri et al., 2021). Seeing that most prior research on learning machining stability and using ML in the context of machine tools has preferred to use ANN models, this research too focuses its attention on ANN.

In an ANN, there are inputs, which in this case are combinations of cutting conditions which result in chatter and not, and there are outputs, which in this case is a prediction about which of the inputs result in chatter or not. In that sense, it becomes a classification problem. Between the input and output layers, there are hidden layers. Each of those has neurons in them. Neurons in one hidden layer are connected to the ones in the next layer and the information flows forward through neurons in layers. Whether or not a particular neuron should be activated or not is decided by calculating a weighted sum and further adding bias with it. In neural networks, these weights and biases are updated by back propagation in an iterative manner. The rate at which weight's update is called the learning rate. There can be many hidden layers, with many neurons in each. There can also be different ways to activate those neurons. There can also be different learning rates and different number of iterations. Some prior work (Cherukuri et al., 2019) has limitedly addressed how some of these hyperparameters influence prediction accuracy in learning a stability diagram. However, there is no report that systematically characterizes how all hyperparameters influence prediction. Furthermore, there is also report on how the size of input data, its train-test split, and distribution of different data classes influence prediction. It is the aim of this paper to present such systematic analysis with ANN to discuss its suitability for learning machining stability. Since the use of ANN and ML is growing, we believe our systematic analysis can further contribute to understanding the limitations and potential of ANN for learning machining stability.

Since the experimental pathway to gather large amounts of data that is needed to properly train an ANN is costly due to the destructive potential of chatter, most prior research on the use of ANN to learn stability has preferred to learn from data generated from analytical model-based simulations (Friedrich et al., 2017; Friedrich et al., 2018; Saadallah et al., 2018; Cherukuri et al., 2019). Though some research used experimental data to train the ANN (Denkana et al., 2020; Postel et al., 2020a, 2020b) those studies did not report on the influence of hyperparameters and/or about the size, train-test split, or about the influence of data classification. To make learning as realistic as possible, this paper trains and tests the ANN using data obtained from emulations on an in-house developed hardware-in-the-loop (HiL) simulator that was built to study chatter (Sahu et al., 2020; Sahu & Law, 2022). Since the HiL simulator emulates cutting, data gathered from experiments on it can be used to train/test the ML model without having to resort to destructive chatter experiments The HiL simulator has a hardware layer approximating the machine and a software layer in which the cutting process is simulated. The closed-loop interaction between these layers emulates cutting processes, even those with nonlinearities in them. As such, we will also train the ANN to learn stability of processes with nonlinear force characteristics that result in regions of conditional stabilities (Sahu et al., 2021a) and for processes exhibiting the interesting process damping phenomena that results in an increase in stability in the low-speed region (Sahu et al., 2021b).

Since measured/emulated data makes no assumption about the underlying physical mechanism responsible for chatter, the ML models to be developed herein will be shown to be agnostic to the process or to any potential nonlinearities in it. The aim is to fit a model to data to show that stability can be learnt for linear and nonlinear cutting processes, and that the ML model architecture for both cases can be the same. We believe that this paper will be the first to demonstrate the use of ANN to learn stability of machining processes with potential nonlinearities. This is also our modest contribution to the literature. The aim of this work is to also present solutions that will help the community move closer towards self-optimizing and autonomous machining systems in which cutting parameter selection can be adapted autonomously and in real-time based on predictions from a ML model that trains itself on data that is gathered from continuously monitoring the process.

The remainder of the paper is organized as follows. At first, in the Section titled 'Gathering data for the ML model', we briefly outline the data gathering phase of the study by describing emulations on the HiL simulator and discussing the resulting stability behavior for linear and nonlinear processes. This data is fed to an ANN, whose architecture is briefly overviewed in the Section titled 'Overview of ANN'. For the case of linear stability, in the Section titled 'Influence of hyperparameters on learning linear stability' we present systematic analysis to characterize the influence of different hyperparameters and data on the prediction accuracy. Knowledge from that analysis is then deployed to learn stability for processes with nonlinear characteristics and for those with process damping. Those cases are discussed in the Section titled 'Linear stability with nonlinear characteristics'. The main 'Conclusions and outlook' follow.

2. Gathering Data for the ML Model

Experimental data necessary to train/test the ML model for it to learn the machining stability diagram were obtained from emulations on a hardware-in-the-loop (HiL) simulator. The hardware had a flexure that approximated a flexible workpiece, a shaker with its power amplifier that supplied a calculated cutting force, a force sensor that measured the force, an accelerometer that measured the response of the flexure, a Compact RIO controller with analog input and output modules, and a computer that connected the hardware and software layer via a LAN connection. The software layer ran on LabVIEW. It included data acquisition, filtering, regenerative cutting force calculation, real-time plotting of time-series and frequency data, data logging for further post-processing, and a graphical user interface to perform controlled experiments. Details about the HiL simulator are available in (Sahu et al., 2020; Sahu & Law, 2022).

Experiments on the HiL simulator to classify which of the combinations of depths of cuts and spindle speeds that resulted in chatter and not were carried out as follows. The flexure was perturbed by applying a static force on it. The depth of cut (b) was then increased in steps at a specified spindle speed (N) and the response was monitored to detect if the system was stable or not. This is much like the process of identifying the stability boundary during real cutting experiments. To distinguish between which combinations of depths of cuts and speeds result in chatter, force and response were monitored. When stable, the flexure was seen to respond to perturbation, with its transients slowly dying out. However, for the case of an unstable depth of cut, due to regenerative effects after the initial perturbation, the response x(t) was observed to grow with time and saturate at finite amplitudes of displacements and forces. The critical depth of cut for that case was recorded. These stability conditions correspond to the global unstable limits for the case of cutting with a process that has linear characteristics. For emulated cutting of steel with assumed linear force characteristics, the resulting data corresponding to the border of stability is shown in Fig. 1(a). Data shown in Fig. 1(a) was obtained as detailed in (Sahu et al., 2020).

When the process has nonlinear force characteristics, there exist regions of conditional stabilities that are are characterized by the process being stable for small perturbations and unstable for larger ones for cutting at parameters within the conditionally stable regions. The procedure to find the global unstable limits in this case is the same as for the case of cutting with linear force characteristics. And, to find the global stable limits, i.e., to find the lower limits of the bistable regions, for every speed of interest, the depths of cuts were decreased in the same step size as they were increased. And the last but one depth of cut at which the finite amplitude instabilities disappear, was recorded as the lower limit of the bistable region. For emulated cutting of steel with assumed exponential force characteristics, the resulting data corresponding to the borders of the globally stable and unstable limits is shown in Fig. 1(b). Data shown in Fig. 1(b) was obtained as detailed in (Sahu et al., 2021a).

When the cutting process exhibits process damping due to interference of a worn tool's flank face with a previously cut surface, the critical chatter-free depths of cuts at low-speeds are observed to be higher than those at highspeeds. This gain in stability was also successfully emulated on the HiL simulator (Sahu et al., 2021b). Experimental data characterizing the stability boundaries for cutting with a process having linear force characteristics and exhibiting process damping is shown in Fig. 1(c). Data shown in Fig. 1(c) was obtained as detailed in (Sahu et al., 2021b).

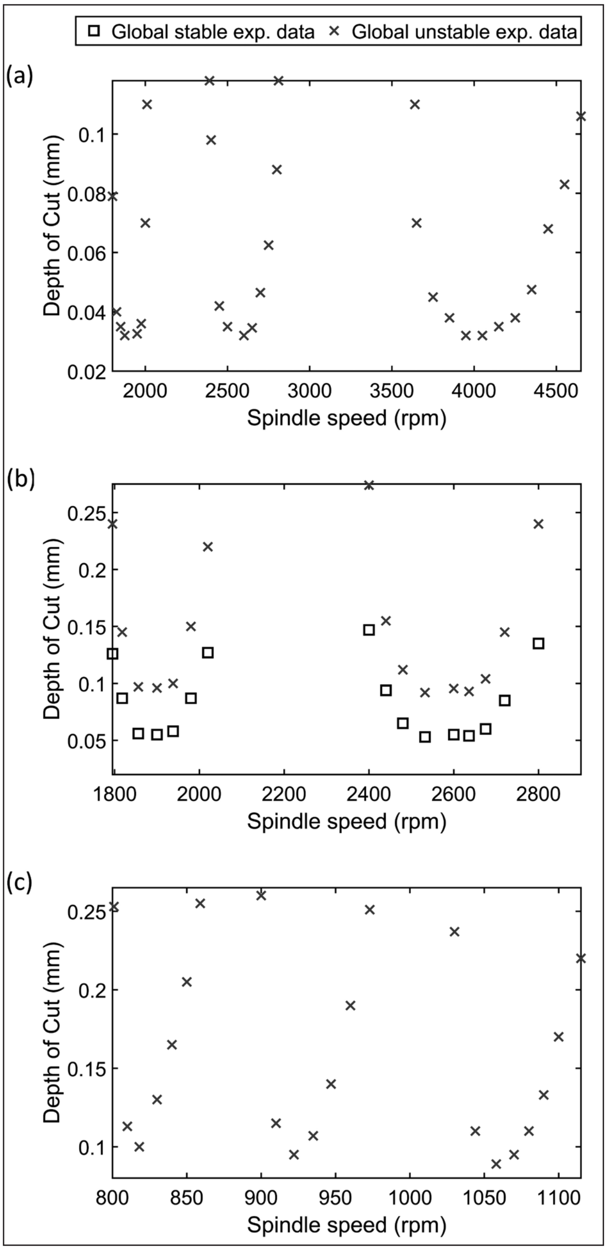


Fig. 1. Experimental data corresponding to unstable cutting conditions obtained from emulations on the HiL simulator. (a) data for the case of a process with linear force characteristics, (b) data for the case of a process with nonlinear characteristics, (c) data for the case of a process with linear characteristics exhibiting process damping.

As is evident from Fig. 1, stability behavior for all three cases is different. Differences arise due to the underlying cutting process mechanics and dynamics being different. The data size is also not the same for all three cases, with the case the linear force characteristics (Fig. 1(a)) having a total of 30 data points which is more than the case of the nonlinear force characteristics that has a total of 32 (16 each for global stable and global unstable case) data points (Fig. 1(b)) and/or for the case of the process exhibiting process damping that has a total of 22 data points (Fig. 1(c)). Since the main idea of fitting a model to data is for the model to be agnostic to the data type, a comparison between the cases is not intended and/or recommended. What is however clear is that for the case with the nonlinear force characteristics there exists a clear bistable region with there being a globally stable boundary as well as a globally unstable boundary. This case will evidently need a ternary type of classification of the data with some data being stable, some being conditionally stable, and some being unstable. The other two cases, i.e., the case of the linear force characteristics and the case exhibiting process damping, the classification will be binary with some data being stable and some being unstable.

Since training an ANN model may need more data than we have obtained from emulations on the HiL simulator, we synthesize the emulated data with more data. Since the region below the boundary is stable and above unstable, we add data points at depth of cut intervals of choice to pad the emulated data. In this manner, even though our emulated data has data points ranging from 9 to 42, we can generate data ranging from 60 - 6000 to train/test our ANN. This allows us to systematically investigate the influence of data size on the prediction accuracy. More on how the synthesized data looks along with its train-test split is discussed in the Section titled ‘Influence of hyperparameters on learning linear stability’.

3. Overview of ANN

ANNs are learning algorithms inspired by the human nervous system (Gurney, 1997). It comprises of nodes, or neurons, constructed in layers and interconnected to replicate the human brain's learning capacity - see Fig. 2. These nodes perform simple mathematical calculations to imitate neuron activity - Fig. 3.

This research employs this learning algorithm to predict chatter occurrence for a given pair of

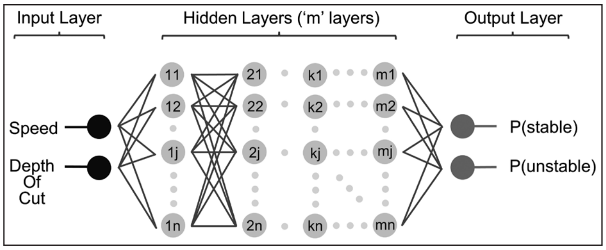


Fig. 2. Basic structure of ANN.

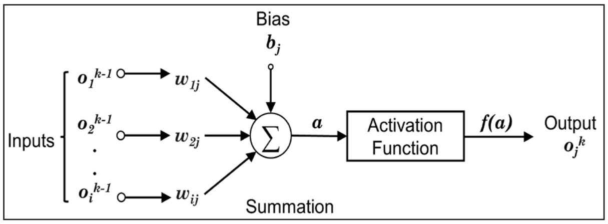


Fig. 3. Mathematical operations within a neuron 'j' in layer 'k'.

spindle speed and depth of cut denoted as $\vec{x}_i$. We follow a supervised learning model, i.e., the model leverages known input-output pairs for training. Training uses a two-stage iterative technique with forward and backward propagation to update parameters. During forward propagation, each neuron in each layer performs a simple linear mathematical operation on the inputs $\vec{O}_i^{k-1}$ to get α, which is a function of weighted inputs and biases. This in turn is operated upon by an activation function to result in an output, O_j^k, which is the input for succeeding layers. The activation function activates neurons based on its relevance to model predictions. The neurons interact, and the output layer predicts the probability of a stable data point or not at the end of forward-propagation.

Since the learning is supervised, we compare the known result with the predicted results and compute the error using a loss function. We use categorical cross entropy function that is amongst the commonly used loss functions for multi class classification problems:

$$E(X,\theta) = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)] \quad ....(1)$$

wherein X= $\{(\vec{x}_1, y_1), ..., (\vec{x}_n, y_n)\}$ denotes training data with y_i being the known class for an input vector $\vec{x}_i$, θ denotes the parameters of the neural network, $\hat{y}_i$ is the computed output of the neural network, and N represents the training batch size. Since the weights and biases influence the prediction made by the network, the error is also a function of weights and biases.

To minimize the error, we use the gradient descent method and update parameters in a backward propagation sense:

$$\theta^{t+1} = \theta^t - \alpha\frac{\partial E(X,\theta^t)}{\partial\theta} \quad ....(2)$$

wherein θ^t represents the parameters of the neural network at t^{th} iteration (epoch), and α denotes the user-defined learning rate.

There are seven hyperparameters that affect the ANN's learning capacity. These include the number of layers, the number of neurons in each layer, the activation function, the learning rate, the data size, the test-train split, and the number of epochs. Weights and biases are estimated during training and are internal model parameters. These do not hence count as hyperparameters that are external to the model. The number of epochs are decided upon in a closed-loop feedback manner by comparing the model's performance improvement in every epoch to a pre-defined tolerance. Since the number of epochs is not fixed a priori, but the tolerance level is, epochs are counted as a hyperparameter. Systematic parametric analysis discussing the influence of each of these parameters on the learning capacity is discussed next.

4. Influence of Hyperparameters on Learning Linear Stability

This section discusses the influence of hyperparameters on learning linear stability. For the data size, we test with data sets of three orders: a set that has 60 data points, a second set that has 600 data points, and a third set that has 6000 data points. For each data set, we have two train-test splits in the ratio of 75-25, and 60-40, respectively. These data sets with their respective splits are shown in Fig. 4. Though the experimentally emulated stability data was used to synthesize data for learning purposes, the experimental data is not used to train and/or test, and only the synthesized data is. The synthesized data was generated by adding data at uniform intervals of speeds and depths of cut, with the intervals being different for different sized data sets. Even though data was uniformly distributed, the distribution of data within a split is taken randomly to avoid clustering. The different classes of data, i.e., the boundary between stable and unstable data points that is to be learnt becomes starker with larger data.

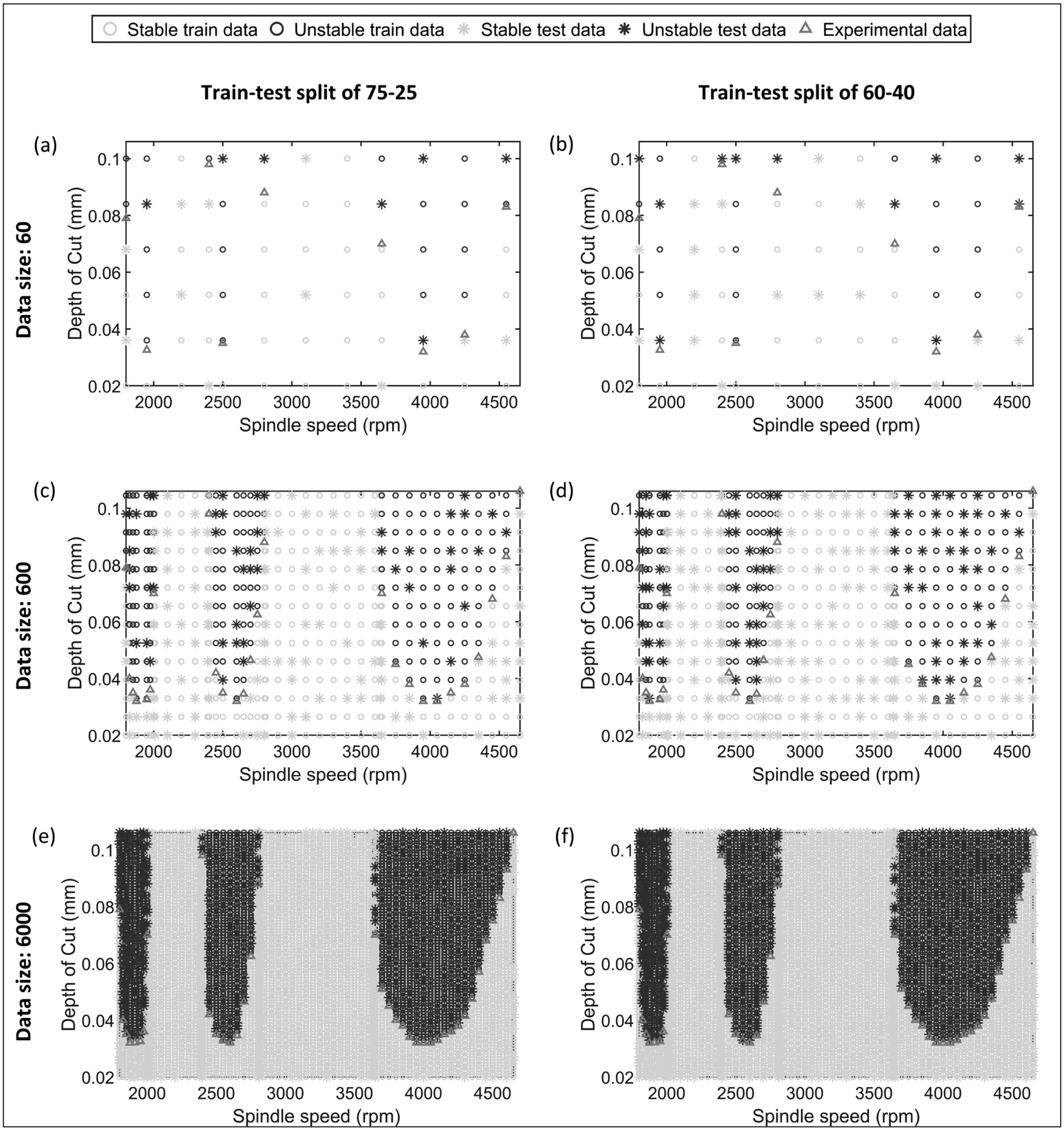


Fig. 4. Different data sets with different train-test splits. (a) 60 data points with a 75-25 split, (b) 60 data points with a 60-40 split, (c) 600 data points with a 75-25 split, (d) 600 data points with a 60-40 split, (e) 6000 data points with a 75-25 split, (f) 6000 data points with a 60-40 split.

For each data set and split, we study the influence of three different activation functions. These include the often-used ReLU, sigmoid, and tanH functions. And, for each data set, its split, and activation function, we also investigate the influence of four different orders of learning rates, namely, 0.01, 0.001, 0.0001, and 0.00001. Furthermore, for every combination of data set and its split, and activation function and learning rate we investigate the influence of number of hidden layers and neurons in every layer. We vary the number of layers from 2 to 20 in steps of 1, and number of neurons in every layer from 32 to 512 in steps of 64. Our choice of different levels/types of hyperparameters is informed by established practices in the use of ANN.

Though we have conducted a detailed parametric and sensitivity analysis to characterize the influence of each hyperparameter, the analysis presented in the subsequent subsections discusses the hyperparameters in order of their influence on the learning capacity. As such, we first present results for how different activation functions influence learning, followed by the role that the learning rate plays. That analysis is followed by the role of the split in the data. For each case we treat the number of layers and number of neurons in each layer as dynamic parameters, i.e., we always show results with these parameters varying within their range of study. All analysis presented herein was conducted in TensorFlow (Abadi et al., 2015). All data used herein is posted on the open science framework (Shanavas, 2022) along with all source codes.

4.1. Accuracy changing with epochs

Prior to investigating the role of different hyperparameters influencing the learning capacity, we discuss herein a representative case for how the loss and accuracy change with epochs (iterations). To optimize the loss function given in Eq. (1), we use the stochastic gradient descent method. Weights and biases are updated in each epoch using a pre-defined tolerance level of a 0.1% improvement as a stopping criterion over the previous 20 epochs. For a representative case of using the ReLU activation function for a dataset size of 600, with a train-test split of 75-25, and with a learning rate of 0.001, loss and accuracy changing with epochs are shown in Fig. 5. As is evident, there is an inverse relationship between the loss function and the accuracy. As the loss reduces over epochs – see Fig. 5(a), the accuracy improves – see Fig. 5(b). Fig. 5 shows results for the training and the testing data, and as is evident, results are consistent. Such analysis forms the basis of parametric analysis of other hyperparameters.

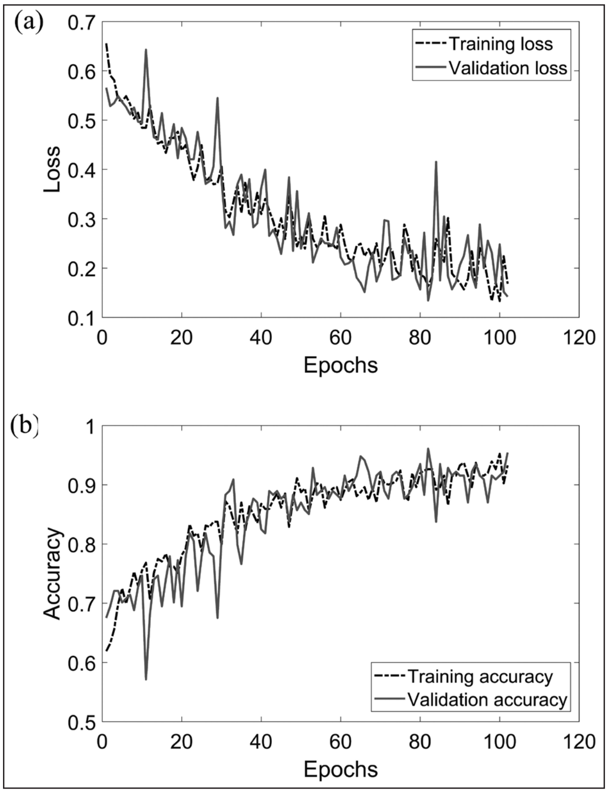


Fig. 5. Variation of (a) Loss and (b) Accuracy with epochs.

4.2. Influence of the activation function

To characterize the influence of the three different activation functions on the learning capacity, we show representative results in Fig. 6 for the data size being 600 and for the 75-25 split, and with the learning rate being fixed at 0.001. We vary the number of hidden layers and the number of neurons in them and report on the accuracy of the predictions. We fit a surface to the results and that fitted surface is shown in Fig. 6 for all three activation functions. As is amply evident, the ReLU activation function is more accurate than the sigmoid and/or the tanH function, reaching a peak accuracy of 95.45% as opposed to peak accuracies of 70.13% and 72.73% for the sigmoid and the tanH functions, respectively. We also see that the number of layers and the neurons per layer both influence the accuracy. Similar observations were made for the different learning rates and with the data size and splits also being different, and since results were consistently better for the ReLU function than with the sigmoid and/or tanH functions, those results are not shown herein. For all further analysis on the influence of other hyperparameters, the activation function was taken to be ReLU.

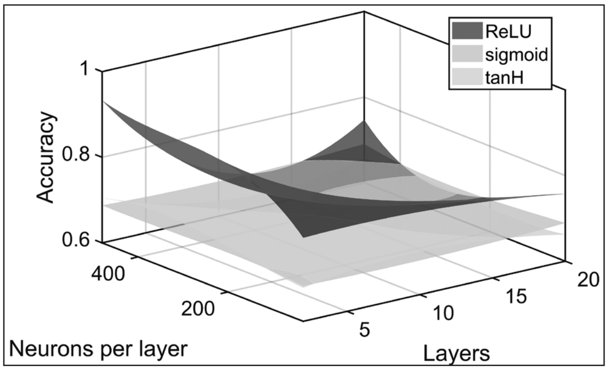


Fig. 6. Fitted accuracy surface plot for different activation functions. Dataset size: 600, Learning rate: 0.001, and train-test split: 75-25.

4.3. Influence of the learning rates and data size

To characterize the influence of learning rates (LR) and data size on the learning capacity for varying levels of layers and number of neurons per layer, we show representative fitted surface results in Fig. 7 for a fixed activation function of the ReLU type and for the train-test split to be 75-25. Similar observations for were made of the 60-40 split, and hence those results are not shown herein. Results suggest that better accuracy is achieved with larger learning rates for when the data size is larger, and that for smaller sized data, a smaller learning rate results in better accuracy. We also observe that as the learning rates increase, the fitted surface's convex nature slowly becomes concave, and that the accuracy improves for when the number of layers is small, with the neurons per layer having a negligible effect. We also observe that larger data in general results in higher accuracies. Peak accuracy for the case of the large data set with a learning rate of 0.01 for the case of 2 layers and 160 neurons in each was observed to be 97.20%. For the case of the medium sized data with a learning rate of 0.001 and with 5 layers and 224 neurons in each, the peak accuracy was 95.45%. And for the small sized data with a learning rate of 0.0001 and for the case of 2 layers and 96 neurons in each, the peak accuracy was observed to be 77.78%.

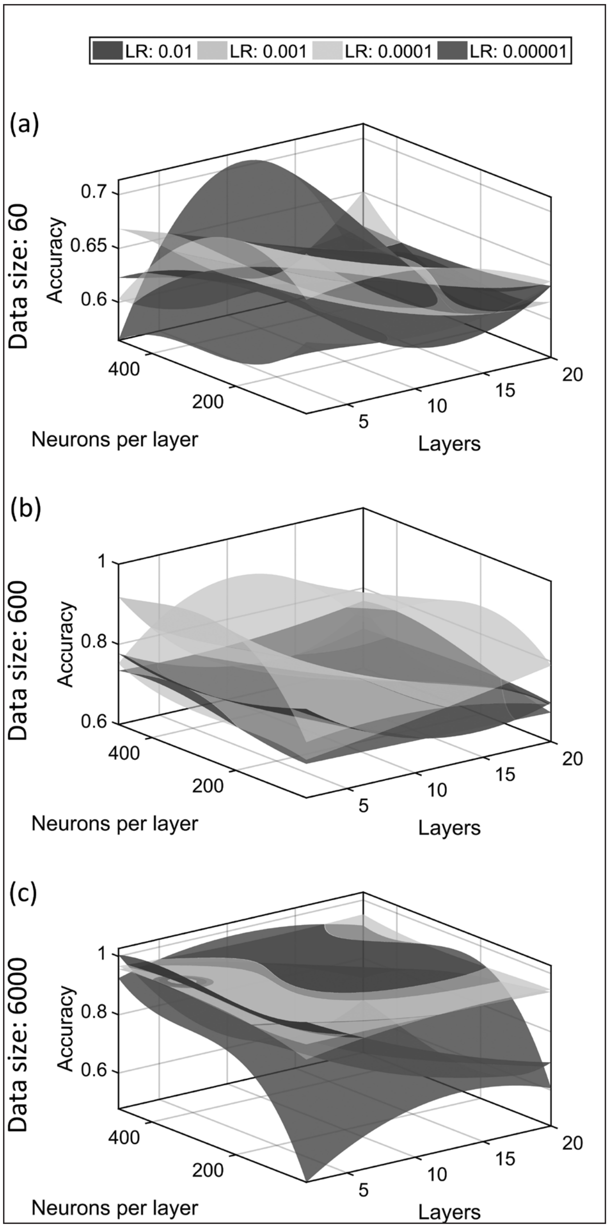


Fig. 7. Fitted accuracy plots for different learning rates (a) dataset sized 60, (b) dataset sized 600, and (c) dataset sized 6000. All results are for the 75-25 train-test split.

Our results suggest that analysis as is presented herein is necessary to make prescriptions about what combination of learning rate, data size, layers, and number of neurons in each will result in the best accuracy. Such systematic analysis for learning machining stability was missing from the literature.

4.4. Influence of the train-test split within data

To characterize the influence of the train-test split within data on the learning capacity for varying levels of layers and number of neurons per layer, we show fitted surface results in Fig. 8 for a fixed activation function of the ReLU type and for the learning rates that resulted in the best accuracies for different data sets. As is evident from Fig. 8, for the case of the data size being 60, it is difficult to infer which of the splits results in better accuracies since there is no clear trend of the accuracies for changing number of layers and number of neuros in each. This is likely due there being too little data to train the model properly. For the data size of 60, the peak accuracy for the 60-40 split is 79.31%, and for the 75-25 split, it is 77.78%. Both these are low and not acceptable for a data learning model that seeks to predict machining stability behavior. The accuracy is higher for the larger data of 600 and 6000, becoming 95.45% for the data set of 600 for the split being 75-25, and 97.20% for the data set of 6000 for the split being 75-25. For these larger data sets there also appears to be a consistent trend which suggests that prediction accuracy degrades with an increase in the number of hidden layers. For the data set of 6000, it is

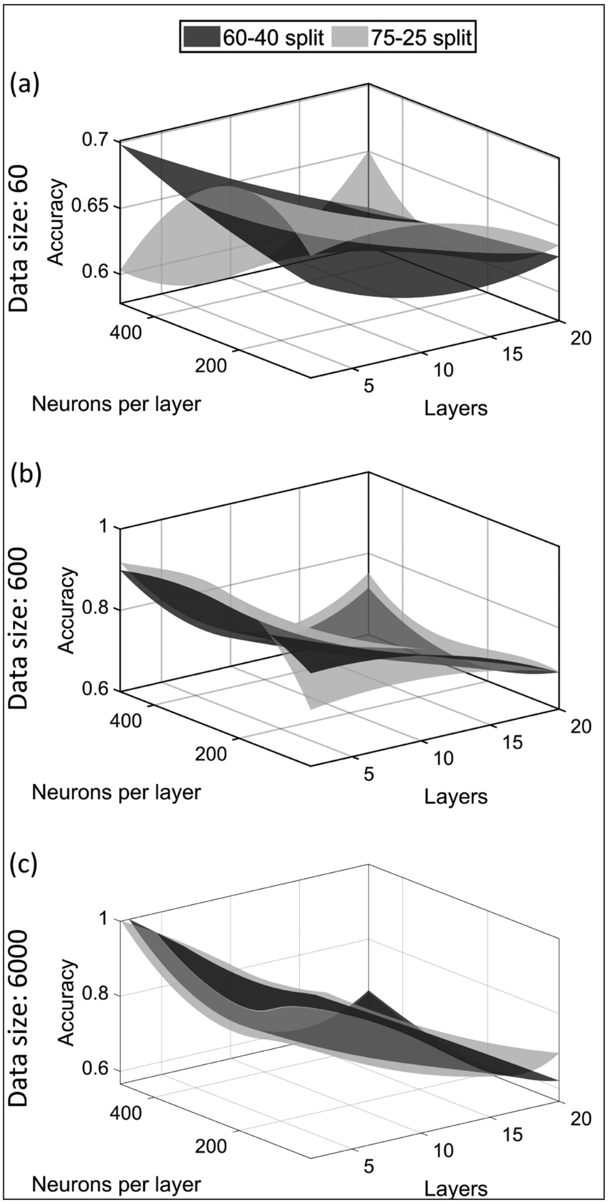


Fig. 8. Fitted accuracy plots for different train-test splits (a) dataset sized 60, (b) dataset sized 600, and (c) dataset sized 6000.

furthermore evident that since there is enough data to train the model with both levels of splits.

Having systematically characterized the influence of different hyperparameters on the learning capacity for linear machining stability, we can draw the following inferences:

- The ReLU activation function results in better prediction accuracy than the sigmoid and/or the tanH functions.
- The choice of the learning rate depends on the data set under consideration, with larger data sets faring better with larger learning rates.
- No significant improvement in prediction accuracy was observed for higher number of hidden layers. However, the number of neurons in the layers were sometimes observed to play a role, see Fig. 8 for example.
- The train-test split of the data was found to be less important for when large data sets were used for training.
- Prediction accuracy improves with increasing size of data, improving from a max accuracy of 95.45% for the case of 600 data points to 97.20% accuracy for 6000 data points. This improvement may seem incremental, but small changes are meaningful if mistakes are costly, since getting the stability diagram wrong may result in incorrect selection of cutting parameters that may result in instabilities that can damage parts of the machine tool system.

To check the generalizability of these inferences, we extend learnings from here to learn stability with nonlinear characteristics.

5. Learning Stability With Nonlinear Characteristics

This section first discusses learning stability for a machining process exhibiting process damping followed by a discussion on learning stability for a process exhibiting bistable behavior. For analysis herein the data sets obtained from experiments on the HiL simulator is synthesized to be of the order of hundreds of data points, with the data size being 530 for the case of process damping and 434 for the process with bistabilities. We use a learning rate of 0.001 for both cases. The activation function is taken to be of the ReLU type. The train-test split is taken to be 75-25. The choice of these hyperparameters is informed by our linear stability analysis for a data set that had a similar order (600 data points). For both cases we investigate the influence of changing number of hidden layers and number of neurons in each.

5.1. Learning stability with process damping

The train-test split for this case is shown in Fig. 9(a) and the fitted surface of the prediction accuracy is shown in Fig. 9(b). And, as is evident, the peak prediction accuracy is observed to be 96.29% for the number of layers being 2 with 160 neurons in each. We also observe the prediction accuracy to degrade with increasing number of layers.

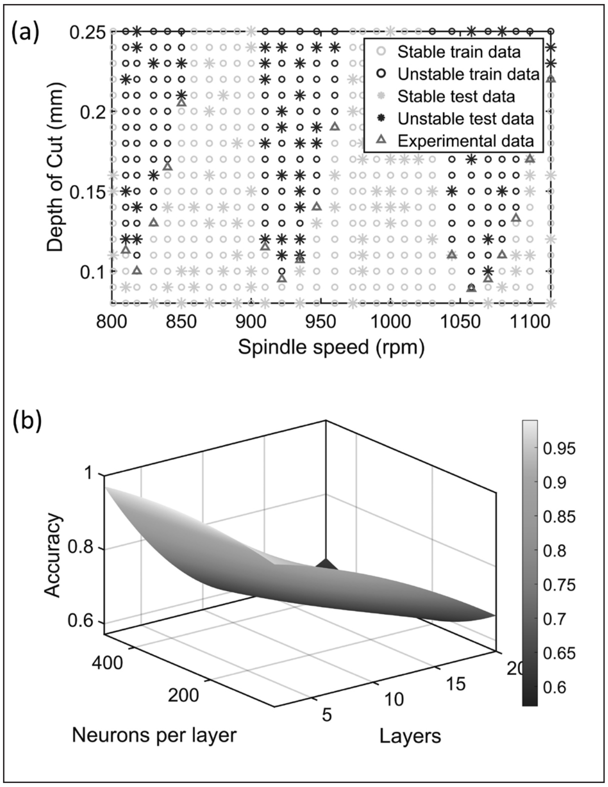


Fig. 9. (a) Data used to learn stability with process damping, (b) Fitted surface showing prediction accuracy of the ANN model.

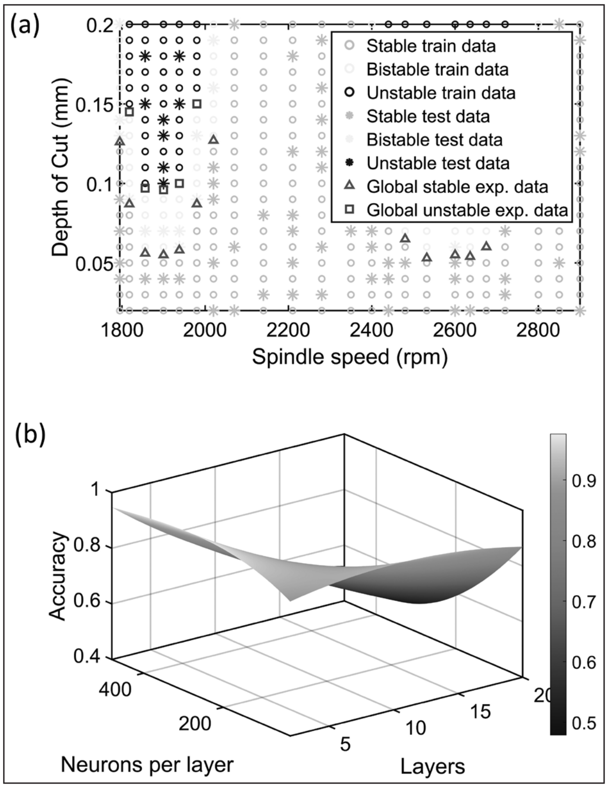


Fig. 10. (a) Data used to learn bistable behavior, (b) Fitted surface showing prediction accuracy of the ANN model.

Since process damping is an interesting phenomenon in which the absolute minimum stability limit improves for lower speeds while remaining unchanged for higher speeds – as is evident from Fig. 9(a), and since improvements are usually observed as the tool wear progresses, and since modelling the tool wear's influence on stability is non-trivial, learning the stability diagram from data as has been demonstrated herein is useful to guide selection of cutting parameters to stabilize a process and to improve the productivity potential.

5.2. Learning stability for a process exhibiting bistable behavior

Processes exhibiting bistable behaviour are characterized by data points that are stable, conditionally stable, and unstable. The ANN model is accordingly updated to have ternary type of classifications. The train-test split for this case for all classes of data is shown in Fig. 10(a) and the fitted surface of the prediction accuracy is shown in Fig. 10(b). And, as is evident, the peak prediction accuracy is observed to be 94.50% for the number of layers being 2 with 160 neurons in each. Interestingly, in this case do not see as strong a trend of the accuracy degrading with increasing number of hidden layers as we did for the case of learning process damping, especially so when the number of neurons in one layer are low. However, in this case too we can conclude that there is no real benefit in increasing the number of hidden layers and neurons in those layers.

Since bistabilities are characterized by the process being stable for small perturbations and unstable for larger ones, learning this behaviour from data can guide selection of cutting parameters to lie outside these zones of conditional instabilities. Moreover, since bistabilities occur due to nonlinearities in cutting force characteristics, which can be difficult to identify and/or model, and since the data learning model used herein is shown to be agnostic to the underlying causes of the observed bistable behavior, and since it is still able to learn that bistable behavior, these results are also useful.

Results discussed in this section show that data learning models can learn stability for cutting processes exhibiting process damping and can also learn bistable behaviour for cutting processes with nonlinear force characteristics. These reports are our modest contribution to the state-of-the-art on using data learning models to learn machining stability behavior.

6. Conclusions and Outlook

Since machining instabilities should be avoided, and since analytical model-predicted stability diagrams often fail to guide stable cutting parameter selection in praxis due to the assumptions the models make and due to the vagaries and uncertainties in the inputs to the model, this paper demonstrated successfully that the stability diagram can instead by learnt from experimental data using a supervised neural network. Systematic analysis was conducted to test for the learning capacity of the model is influenced by its hyperparameters. Though our investigations suggest that hyperparameters must be separately tuned for each data set type, we observed in general that the ReLU activation function results in better prediction accuracy than the sigmoid and/or the tanH functions that for all data set types. We also found that the choice of the learning rate depends on the data set under consideration, with larger data sets faring better with higher learning rates. We also observed that there was no significant improvement in prediction accuracy for higher number of hidden layers. However, the number of neurons in the layers were sometimes observed to play a role. We also found that for larger data sets the train-test split did not play much of a role. And, though we found the prediction accuracy to improve marginally with increasing the size of data, we argue that small changes are meaningful since mistakenly guiding wrong cutting parameter selection can be costly due to the destructive nature of chatter.

Our observations were found to be consistent across three different types of datasets gathered from emulated experiments, suggesting that the learning model is agnostic to the underlying process physics. This is the first such report in the literature of a machine learning model being blind to potential nonlinearities in the cutting process. This is also the strength of the learning model. Such analysis can inform future research to help the community move closer towards self-optimizing and autonomous machining systems in which cutting parameter selection can be adapted autonomously and in real-time based on predictions from a ML model that trains itself on data that is gathered from continuously monitoring the process.

7. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jozefowicz, R., Jia, Y., Kaiser, L., Kudlur, M., ... Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Aggogeri, F., Pellegrini, N., & Tagliani, F. L. (2021). Recent Advances on Machine Learning Applications in Machining Processes. *Applied Sciences, 11*(18), 8764. https://doi.org/10.3390/app11188764

Altintas, Y., Stepan, G., Budak, E., Schmitz, T., & Kilic, Z. M. (2020). Chatter stability of machining operations. *ASME. Journal of Manufacturing Science and Engineering, 142*(11), 110801. https://doi.org/10.1115/1.4047391

Bergman, B., & Reimer, S. (2021). Online adaption of milling parameters for a stable and productive process. *CIRP Annals – Manufacturing Technology, 70*, 341-344.

Chen, G., Li, Y., Liu, X., & Yang, B. (2021). Physics-informed Bayesian inference for milling stability analysis. *International Journal of Machine Tools and Manufacture, 167*, 103767.

Cherukuri, H., Bernabeu, E. P., Selles, M., & Schmitz, T. (2019). Machining chatter prediction using a data learning model. *Journal of Manufacturing and Materials Processing, 3*(2), 45.

Cornelius, A., Karandikar, J., Gomez, M., & Schmitz, T. (2021). A Bayesian framework for milling stability prediction and reverse parameter identification. *Procedia Manufacturing, 53,* 760-772. https://doiorg/10.1016/j.promfg.2021.06.073

Denkena, B., Bergmann, B., & Reimer, S. (2020). Analysis of different machine learning algorithms to learn stability lobe diagram. *Procedia CIRP, 88,* 282-287.

Friedrich, J., Hinze, C., Renner, A., & Verl, A. W. (2017). Estimation of stability lobe diagrams in milling with continuous learning algorithms. *Robotics and Computer-Integrated Manufacturing, 43,* 124-134.

Friedrich, J., Torzewski, J., & Verl, A. W. (2018). Online learning of stability lobe diagrams in milling. *Procedia CIRP, 67,* 278-283.

Gurney, K. (1997). An introduction to neural networks (1st ed.). CRC press.

Karandikar, J., Honeycutt, A., Schmitz, T., & Smith, S. (2020). Stability boundary and optimal operating parameter identification in milling using Bayesian learning. *Journal of Manufacturing Processes, 56,* 1252-1262.

Kvinevskiy, I., Bedi, S., & Mann, S. (2020). Detecting machine chatter using audio data and machine learning. *The International Journal of Advanced Manufacturing Technology, 108,* 3707-3716. https://doi.org/10.1007/s00170-020-05571-9

Liu, P. L., Du, Z. C., Li, H. M. et al. (2021). Thermal error modeling based on BiLSTM deep learning for CNC machine tool. *Advances in Manufacturing, 9,* 235-249. https://doi.org/10.1007/s40436-020-00342-x

Liu, Y., & Altintas, Y. (2021). Transmissibility Enhanced Inverse Chatter Stability Solution. *ASME Journal of Manufacturing Science and Engineering, 144*(1), 011002. https://doi.org/10.1115/1.4051286

Möhring, H. C., Wiederkehr, P., Erkorkmaz, K., & Kakinuma, Y. (2020). Self-optimizing machining systems. *CIRP Annals, 69*(2), 740-763. https://doi.org/10.1016/j.cirp.2020.05.007

Munoa, J., Beudaert, X., Dombovari, Z., Altintas, Y., Budak, E., Brecher, C., & Stepan, G. (2016). Chatter suppression techniques in metal cutting. *CIRP Annals, 65*(2), 785-808. https://doi.org/10.1016/j.cirp.2016.06.004

Postel, M., Bugdayci, B., & Wegener, K. (2020a). Ensemble transfer learning for refining stability predictions in milling using experimental stability states. *The International Journal of Advanced Manufacturing Technology, 107,* 4123-4139.

Postel, M., Bugdayci, B., Kuster, F., & Wegener, K. (2020b). Neural network supported inverse parameter identification for stability predictions in milling. *CIRP Journal of Manufacturing Science and Technology, 29(A),* 71-87. https://doi.org/10.1016/j.cirpj.2020.02.004

Rahimi, M. H., Huynh, H. N., & Altintas, Y. (2021). On-line chatter detection in milling with hybrid machine learning and physics-based model. *CIRP Journal of Manufacturing Science and Technology, 35,* 25-40. https://doi.org/10.1016/j.cirpj.2021.05.006

Reddy, T. N., Shanmugaraj, V., Vinod, P., & Krishna, S. G. (2020). Real-time thermal error compensation strategy for precision machine tools. *Materials Today: Proceedings, 22*(4), 2386-2396. https://doi.org/10.1016/j.matpr.2020.03.363

Saadallah, A. Finkeldey, F., Morik, K., & Wiederkehr, P. (2018). Stability prediction in milling processes using a simulation-based Machine Learning approach. *Procedia CIRP, 72,* 1493-1498.

Sahu, G. N., Jain, P., Wahi, P., & Law, M. (2021a). Emulating bistabilities in turning to devise gain tuning strategies to actively damp them using a hardware-in-the-loop simulator. *CIRP Journal of Manufacturing Science and Technology, 32,* 120-131.

Sahu, G. N., Jain, P., Law, M., & Wahi, P. (2021b). Emulating chatter with process damping in turning using a hardware-in-the-loop simulator. Proceedings of the 8th Int. and 29th National All India Manufacturing Technology, Design and Research Conference AIMTDR 2021.

Sahu, G. N., & Law, M. (2022). Hardware-in-the-loop simulator for emulation and active control of chatter. HardwareX, 11, e00273.

Sahu, G. N., Vashisht, S., Wahi, P., & Law, M. (2020). Validation of a hardware-in-the-loop simulator for investigating and actively damping regenerative chatter in orthogonal cutting. *CIRP Journal of Manufacturing Science and Technology, 29*(A), 115-129.

Shanavas, N. A. (2022). Learning machining stability diagrams from data using neural networks. Retrieved from osf.io/wds5g

Shi, F., Cao, H., Wang, Y., & Feng, B. (2020). Chatter detection in high-speed milling processes based on ON-LSTM and PBT. The International Journal of Advanced Manufacturing Technology, 111, 3361-3378. https://doi.org/10.1007/s00170-020-06292-9

Shi, F., Cao, H., Zhang, X., & Chen, X. (2020). A Reinforced k-Nearest Neighbors Method With Application to Chatter Identification in High-Speed Milling. IEEE Transactions on Industrial Electronics. PP. 1-1. 10.1109/TIE.2019.2962465.

Tarng, Y. S., & Chen, M.C. (1994). An intelligent sensor for detection of milling chatter. *Journal of Intelligent Manufacturing, 5,* 193-200. https://doi.org/10.1007/BF00123923

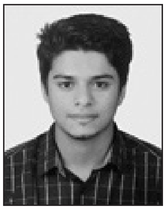
Unver, H. O., & Sener, B. (2021). A novel transfer learning framework for chatter detection using convolutional neural networks. *Journal of Intelligent Manufacturing.* https://doi.org/10.1007/s10845-021-01839-3

Vaishnav, S., Agarwal, A., & Desai, K. A. (2020). Machine learning-based instantaneous cutting force model for end milling operation. *Journal of Intelligent Manufacturing, 31,* 1353-1366.

https://doi.org/10.1007/s10845-019-01514-8

Wang, Y., Zhang, M., Tang, X., Peng, F., & Yan, R. (2021). A kMap optimized VMD-SVM model for milling chatter detection with an industrial robot. *Journal of Intelligent Manufacturing.* https://doi.org/10.1007/s10845-021-01736-9

Yesilli, M. C., Khasawneh, F. A., & Otto, A. (2020). On transfer learning for chatter detection in turning using wavelet packet transform and ensemble empirical mode decomposition. *CIRP Journal of Manufacturing Science and Technology, 28,* 118-135. https://doi.org/10.1016/j.cirpj.2019.11.003



Namras Amakkattil Shanavas is an Assistant Manager at the Analytics Department at EXL Service Holdings Inc. He received his Master's degree from the Indian Institute of Technology Kanpur, India. His research interests were focused on understanding and optimizing different machine learning algorithms especially centred around machine tools vibration.
(E-mail: namras0719@gmail.com)

Mohit Law is an Associate Professor at the Department of Mechanical Engineering at the Indian Institute of Technology Kanpur, India. He received his Ph.D from the University of British Columbia, Canada. His research interests centre on understanding how and why machine tools vibrate, measuring those vibrations, and on how best to mitigate them.

Manjesh K. Singh is an Assistant Professor at the Department of Mechanical Engineering, IIT Kanpur. He received his Ph.D from ETH Zurich, Switzerland in 2016. His research focuses broadly on soft matter and tribology involving both computer simulations and experiments.
(E-mail: manjesh@iitk.ac.in)