MM Science Journal | www.mmscience.eu

ISSN 1803-1269 (Print) | ISSN 1805-0476 (Online)

Special Issue | HSM 2023

17th International Conference on High Speed Machining

October 25-28, 2023, Nanjing, China

DOI: 10.17973/MMSJ.2023_11_2023119

HSM2023-00056

PROCESS CONTROL COMBINING MACHINE LEARNING AND FINGERPRINT APPROACHES

A. Garnier [1], C. Cecchinel[1], X. Beudaert[2*]

[1] DataThings S.A., Luxembourg, Luxembourg

[2] IDEKO, Dynamics & Control Department, Elgoibar, Basque Country, Spain

*Corresponding author; e-mail: xbeudaert@ideko.es

Abstract

Manufacturing operations in large machine tools often requires several hours per part. Ensuring output quality is vital to avoid time and financial losses. While quality assurance was always problematic and costly, the recent advent of Industry 4.0 brought a new perspective to the problem as cutting machines are now fully digitized. This paper proposes a process control framework that combines a fingerprint approach that detects deviations with respect to the validated process and a Long Short-Term Memory (LSTM) algorithm that predicts the upcoming signals. This paper demonstrates how combining these two methodologies surpasses the performance of previous purely learning-based algorithms.

Keywords:

Process control, Machine learning, Monitoring, Predictive maintenance

1 INTRODUCTION

Process control plays a critical role in achieving autonomous machining. With modern advancements in Industry 4.0 and machine learning, the potential for transformation in the manufacturing industry is immense. However, despite the conceptual framework of big data applications in machine tools being well established [Gao 2020], the practical implementation in real production environments remains largely untapped.

The cutting tool condition constitutes a primary factor causing disruptions in the machining processes. Hence, the study of tool wear and tool breakage has been a substantial focus of previous research [Li 2022]. [Zhang 2023] analysed the tool wear and tool breakage based on physical models.

Recently, a lot of attention has been dedicated to the use of artificial intelligence for the detection of tool breakage [Xiao 2022]. [Li 2019] proposed a tool wear monitoring and prediction under varying cutting conditions using a meta-learning approach. [Wang 2019] trained a neural network to evaluate the tool wear based on the spindle power. However, these studies are often conducted in controlled environments and do not fully address the challenges faced in real-world manufacturing settings.

Only few research publications are using real production data to build a system able to detect tool breakage issues. [Zhang 2020] trained a Gradient Boosting Decision Tree to predict tool failure based on production data. In real-world manufacturing processes, datasets are typically unbalanced, with most samples corresponding to error-free processes. This imbalance necessitates the development of specific methodologies for automatic process control [Li 2022].

In addition to tool breakage, various other factors can also lead to disruptions in machining processes. These factors can include variations in material properties, machine settings or environmental conditions. Machining processes are often characterized by repetition and continuity. In such scenarios, each instance of the process should ideally exhibit a consistent behaviour similar to its predecessors. In [MacGregor 1995], continuous chemical processes are monitored through the implementation of multivariate statistical process control (SPC) techniques. This enables real-time online monitoring of intricate processes, ensuring timely detection of out-of-control situations and maintaining optimal production quality while the only requirement for applying this method is a large database on past operations. More recently, in [Tangjitsitcharoen 2013], the utilization of in-process monitoring and statistical process control in the CNC turning process for surface roughness evaluation is explored. The authors emphasize the monitoring of cutting force and the application of a surface roughness prediction model to assess product quality. However, it is noteworthy that their approach assumes an existing model and places greater emphasis on product quality, potentially overlooking the process aspect.

Existing work has considered process fingerprint approaches and machine learning techniques in isolation. This paper aims to bridge this gap by presenting a framework applied to machining operations that merges machine learning and fingerprint approaches. The objective is to construct a system capable of automatically stopping the machining process in case of anomalous behaviour, thereby minimizing the need for constant human monitoring. This paper demonstrates how combining these two methodologies surpasses the performance of previous learning-based algorithms.

2 FRAMEWORK COMBINING PROCESS FINGERPRINT & MACHINE LEARNING

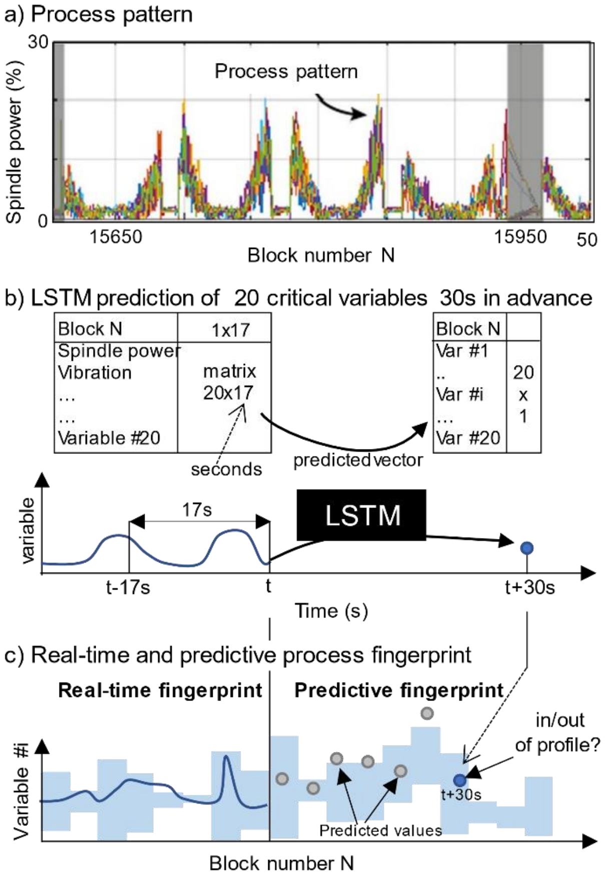

Fig. 1: Process control framework combining process fingerprint and machine learning methodologies.

In manufacturing, real production datasets are often imbalanced, with an overwhelming majority of samples representing error-free processes. This characteristic becomes even more pronounced when certain tools and processes tend to induce tool breakage, while others function flawlessly. When it comes to large machine tools machining high-value workpieces, it is quite typical for an operator to be dedicated to continuously monitoring the process. This constant vigilance is vital to prevent potential damage to the part in case of tool failure. Ideally, the machine should be able to automatically stop the production in case process deviation and human intervention should only be necessary to solve the problem and restart the production process. This would significantly reduce human monitoring efforts and associated costs.

As shown in Fig. 1a, we establish a process fingerprint based on the dataset representing error-free production, aiming to identify process deviations in real-time. Indeed, it is crucial to implement a monitoring solution even in machining operations that have historically been problem-free. This provides a reactive safety net in the event of unexpected issues. Moreover, the ability to predict future monitored values provided by the machine learning approach allows a proactive anticipation of production failures. Hence, our proposed process control framework merges both reactive and proactive approaches using fingerprint and learning methodologies (Fig. 1b&c).

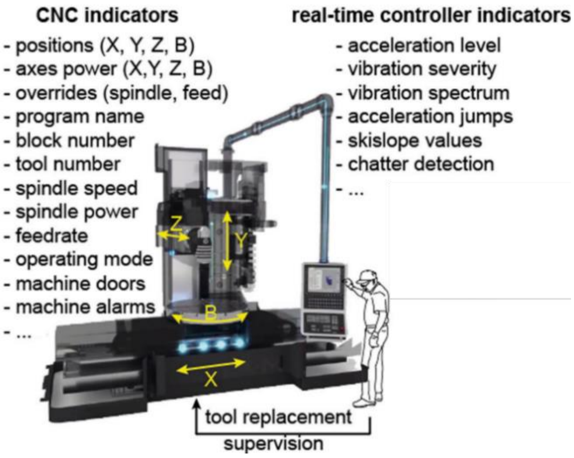

Fig. 2: Most relevant monitored data.

We employ a process fingerprinting approach on error-free production data to swiftly detect and respond to any deviations. This approach involves applying the fingerprinting technique to all monitored variables and generating a statistical profile for each CNC block, comprising minimum, maximum, average, and standard deviation values. From there, a threshold can be defined so that, during subsequent productions, the machine is automatically stopped as a preventive measure whenever a variable deviates more than the chosen threshold from the error-free productions considered as the norm.

Then, on the same dataset, enhanced with machine stop labels, we utilize a machine learning technique called Long Short-Term Memory (LSTM) [Hochreiter 1997]. During execution, the process control system applies real-time values to the LSTM model to predict the near future. These predicted values are then compared to the statistical fingerprint profile, and whether the predicted values are in-profile or out-of-profile a risk score is computed accordingly to determine the likelihood of a machine stoppage.

3 INDUSTRIAL USE-CASE DESCRIPTION

This section details the implementation of the monitoring and control platform on a large milling machine running long and repetitive programs.

3.1 Data acquisition

The digitization of the machine tool industry presents new opportunities for intelligent manufacturing and more efficient production. Efficient monitoring requires the combination of both high-frequency and low-frequency data, covering both slowly changing variables (tool parameters, spindle speed, program line number) and high dynamic variables (acceleration, spindle power). The access to the machine state allows for the tracking of key production variables. Both PLC and CNC variables are measured to capture the machine state, although acquisition frequency is limited to 1Hz to prevent data exchange overload with the CNC. Fig.2 presents the most relevant data that are monitored in this milling machine.

The vibration level is monitored through two accelerometers located close to the cutting point with a 4kHz sampling frequency. To reduce the cost and burden of transferring all high-frequency data to the cloud platform, local processing of the acceleration signals is performed to extract meaningful information. The vibration frequency spectrum is obtained through the Fast Fourier Transform, with only the ten highest peaks transferred to the cloud platform, along with the computed vibration severity for different frequency ranges. The lower frequency range corresponds to the vibration frequencies of the complete machine structure (from 20Hz to 80Hz) and it is monitored with special attention because it is the frequency range in which strong chatter vibrations causing tool breakages can appear. The second vibration severity range covers the complete spectrum from 20 to 1000Hz. Hence, the vibration level can be well reflected with a limited amount of data thus increasing the ratio between data and information.

The CNC also provides information on the current part program, program line, feed rate, and spindle speed. The actions of the operator are also relevant for the monitoring platform. In this application, the manufacturing process should be fully automated. Hence, the unplanned stops are automatically marked in the historical training dataset based on a sequence of action that the operator realizes when a production failure appears (feed override reduction, opening of the door, switch from automatic to manual mode…). This automated marking strategy enables retrospective labelling of the historical dataset.

3.2 Machining operations

The machine tool runs 2 shifts daily with 4 fixed machining programs and 10 tools to complete each workpiece. Each workpiece takes more than 40 hours to machine. The program and cutting conditions are frozen; hence, variation arises primarily from the raw material properties and tool wear. The rotary axis B is constantly changing its orientation during the machining operations, so the CNC blocks are very short. To minimize the risk of scrapping a valuable part, a machine operator continuously monitors the machining process. Hence, in this repetitive manufacturing context, it would be beneficial to have an automated system capable of performing the task of process surveillance.

4 PREDICTIVE PROCESS CONTROL

Successfully preventing faulty behaviours in a 40 hours industrial process requires to understand better the way monitored data relate to said faults. Indeed, while it is impractical to define upper and lower values for each variable over the full length of the process, the superposition of 12 executions of a same program showed very similar patterns for fault-less executions (See Fig. 1a). These variables move around a lot, but they do so consistently from execution to execution, such that for a same block number, variables do not deviate that much overall.

From this observation, we devise a fingerprinting profile of processes, that relies on the Gaussian distribution of variables, not over a whole execution, but for each block number over a span of fault-less executions. It is expected that using these profiles to identify when variables strongly deviate from the norm should help to prevent faults in processes. Profiling, as well as every step described in this paper, is agnostically applied to the hundreds of numerical variables provided by the machine, unless stated otherwise.

Program blocks represent indivisible operations executed on the machine, encompassing essential actions like tool movement, spindle speed adjustment, and material cutting. Depending on the complexity of the operation, such operations can span over a large interval of durations. With a collection rate of 1Hz, it is not uncommon for some block numbers that last for a few seconds to span over a few successive collected data points, and for some others that last less that one second to be completely missed out in the collected data. While fortunately, block numbers are incrementally numbered such that missing ones can be interpolated, such an interpolation inevitably leads to a loss in precision. The data collection rate of 1Hz also means that block number duration can only be estimated down to the second.

4.1 Fingerprint of production process

A common practice in profiling is to define an acceptance range based on the statistics obtained from the learnt data [Dare 2006]. Depending on the desired sensitivity, a threshold of acceptability can be defined from values a number (*n*) times the standard deviation (*std*) above and below the average value (*avg*): [*avg*−*n*×*std*;*avg*+*n*×*std*]. Assuming a normal distribution of the values, 2 standard deviations from the average enclose 95% of the data.

Using statistical profiles, the expected range of values for each block and each variable is known. By taking all these profiles, a representation of the expected values of the process, which constitutes the fingerprint, can be obtained. This fingerprint describing an optimal operation of the process (without stop) can be compared to new data to check if the latter also reflects an optimal operation of the process.

4.2 Predicting profiles with a neural network

Long short-term memory (LSTM) [Hochreiter 1997] is a supervised machine learning model designed to deal with data that are sequential in nature, like video or, as for our use case, time series. In fact, it relies on the definition of a data sequence of fixed length, like a succession of pictures for a video, or the recent history up to now for each time series of variables collected by production sensors. Being implemented as a layer of all prominent neural network frameworks, it allows to be trained toward a large panel of tasks. Among them we will focus here on regression, which is the one implemented in our contribution. Regression consists of refining a mathematical polynomial approximation to match sequence of observations with other numerical outputs. We use it to predict future values that variables will take in the coming seconds, based on their immediate history.

The way LSTM (Long Short-Term Memory) works relies heavily on the length of the data sequences it learns from. This length, essentially a window into the past, should encompass patterns in production that could suggest a fault is about to happen. Choosing this length needs a lot of thought, as it depends heavily on the specific machine and process.

In addition to this LSTM parameter, there is another parameter that relates to our objective to predict unplanned stops to come, which is the time span separating the recent history from the point in time we want to predict for. While this is not a parameter of LSTM *per se*, regarding our approach it is a parameter of equal importance to the size of the input sequences. The fine-tuning of both these parameters is detailed in Section 5.4.

Two major issues are faced when trying to predict unplanned production stops frow raw physical measurements. First, such unplanned stops are not so frequent and very punctual events, *i.e.*, once a process stops, most physical measurements also stop to be of interest since the machine is not working anymore, until the process takes place again. Second, the complexity of such long and diverse milling processes is reflected in the large number of features that store physical measurements in the dataset. Thus, it turns out hard to identify behaviours, tendencies in these measurements that accompany and could define an unplanned stop. From there, rather than a peculiar event that appears suddenly in the machine's behaviour and instantaneously comes with an unplanned stop, the assumption is that the machine's behaviour has gone steadily worse over the time until it ended up with an unplanned stop. Such an assumption leads to a shift in how to predict productions stops from trying to predict very occasional and punctual events. Instead, the proposed approach consists of identifying, among the physical measurements at disposal, the ones that show peculiar behaviours or tendencies when getting closer to unplanned productions stops. Once done, the matter becomes to predict the values that these measurements will take based on their recent history. At last, if a set of successive predictions consistently exhibit out-of-profile values, in such a proportion known to accompany unplanned production stops in historical data, then it gives a certain confidence that such a stop is likely to happen.

In practice, the prediction model used toward this makes use of machine learning techniques. Two LSTM-based sequential neural networks (NN) are trained on a recent history of 17 seconds over a specific selection of features to try and predict the values of these same features 30 seconds ahead:

- The first NN trains to predict the values themselves within 30 seconds. Results have shown that if this NN grasps well the 'tendency' of the values to come (if they will be below or above the expectancy for their block number in the program execution), it does not work well at predicting the amplitude from this expectancy.
- Hence, the second NN was introduced to train to predict specifically this amplitude, disregarding if it will be below or above the expectancy.
- Finally, the compound of both NN makes use of the second NN predictions to rescale the first NN predictions, such that the model can grasp both tendency and amplitude in the evolution of features, which better help to recognize potential patterns of concern in the prediction.

4.3 Risk score

The model described in Section 4.2 allows to predict features to come from their recent history. Hence it does not allow to predict stops as is. To this end, a risk score is introduced that relies on the behaviour that the model predicts, so that a suspicious behaviour raises the risk of a stop happening.

By forecasting the behaviour of features of concern in the cutting process lifecycle, the idea is thus to predict the behaviours that may lead to a stop, rather than the stop itself. However, the ultimate question remains as binary, whether there will be an unplanned production stop happening soon or not. Toward this end, an out-of-profile binary label is introduced that allows the computation of several scores to evaluate the quality of the predictive model from the quality of its predictions. Concretely, this out-of-profile (OOP) label turns true when any of the selected features deviate from the mean more than n times its standard deviation and stays false otherwise. This indeed reflects the acceptance range discussed in Section 4.1. As for the risk of stops happening, the idea is to make use of these scores to determine the accuracy of a prediction being true, no matter if positive or negative. Toward this, positive and negative predictive values (*PPV* and *NPV*) can be computed:

$$PPV = \frac{\#true\ positives}{\#predicted\ positives} \tag{1}$$

$$NPV = \frac{\#true\ negatives}{\#predicted\ negatives} \tag{2}$$

PPV is the same score as the precision, and *NPV* its negative counterpart. From there the *risk* score of a prediction *p* is defined as:

$$risk(p) = \begin{cases} PPV\ if\ p \\ 1 - NPV\ if\ \bar{p} \end{cases} \tag{3}$$

4.4 Automatic stop of the machine

The fingerprint is used with both live and predicted data to create a decision process that can automatically stop the machine and avoid any production issues. On one hand, there are undoubtedly out-of-profile recent behaviour that can be detected by comparing live data with expected values. On the other hand, the use of the risk score aims at estimating a probability that metrics will be out-of-profile in 30 seconds. Both cases have in common that they can be reduced, for each datapoint, to a single probability, which for live data can only be either 0% or 100%. From there, a unified decision process should be defined. The questions that such an approach must answer are the following:

- *(Q1)* How long a succession of out-of-profile data points constitute a behaviour degraded enough for the machine to be stopped?
- *(Q2)* Even when leading to a stop, out-of-profile data points are mixed within in-profile data; what percentage of the data the out-of-profile part must represent to be of any concern?
- *(Q3)* In the mathematical sense, a data point is considered abnormal when its distance to the mean goes beyond the standard deviation; however, following this strict rule shows a lot of slightly out-of-profile data points that are of no concern since they do not lead to an unplanned stop; how far from the mean a data point must be to be considered out-of-profile?

Tackling these questions requires the careful choice of associated parameters, such as:

- A standard deviation coefficient working as a threshold above which data are considered out-of-profile.
- The time span in which a succession of out-of-profile data are worrying enough to justify the pre-emptive stop of the machine.
- A density threshold to check over said time span for out-of-profile data to be considered present enough to be of concern.

Such parameters must be considered independently both for live and predicted data. Balancing both decision algorithms to make the ultimate choice of stopping the machine is another issue that must be addressed. Finally, if the time span over the predicted values goes over the prediction lapse, the possibility arises to compare older predicted out-of-profile data with live, potentially out-of-profile, data to weigh in the decision to stop the machine.

These hard questions are not addressed in this paper and should be the object of future works.

5 INDUSTRIAL IMPLEMENTATION

The performance of the predictive framework is tested and demonstrated on a dataset collected by the experimental

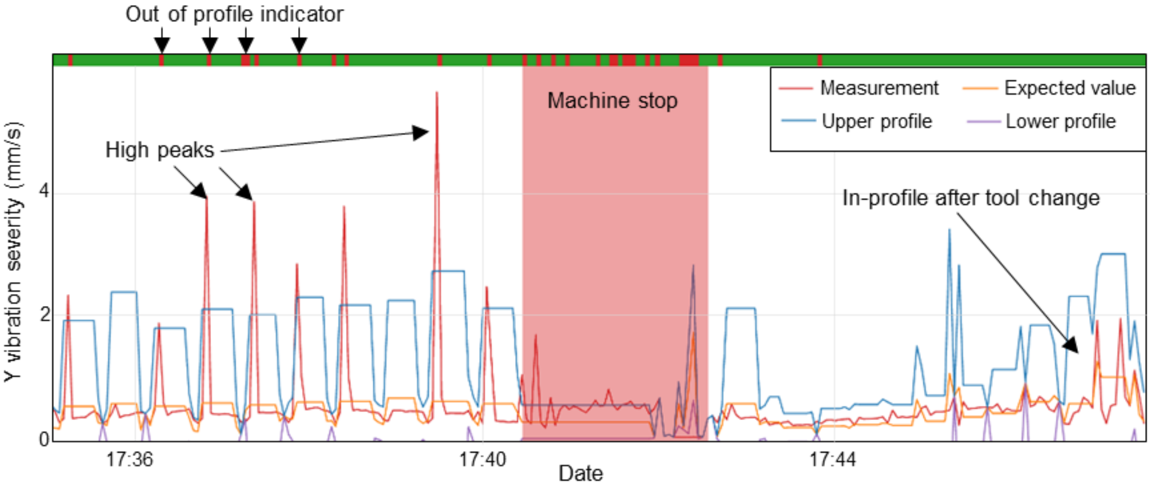

Fig. 3: Live values for a given variable going out of the profile before production stop, staying in the profile after the tool change.

setup described in section 3. To compare it with previous works [Zhang 2020] and provide a better insight into the efficiency of the framework, we focus on the same periods of time in the dataset, both for training the model and measuring statistical scores to compare predictions with actual data.

5.1 Dataset

The dataset comprises CSV files that encompass program executions, variable measurements, cutting tool names, and various contextual information. The data collection period spans approximately four years, from March, 2017 to May, 2021. The measurement files contain approximately 38 million entries, totalling a data weight of 150 gigabytes (GB). All these CSV files contain a shared timestamp and/or a unique key that facilitates data synchronization across the multiple files. Consequently, we can harmonize and integrate the data into the model, described hereafter, by either joining based on the timestamp or using the unique identifier.

5.2 Modelling the cutting process

To ensure efficient storage and browsing of the dataset, we constructed a database that incorporates both a graph structure, which models relationships between concepts, and a temporal component for storing large-value time series data. The model comprises a machine to which multiple programs are attached. Each program consists of a set of instructions known as blocks, which represent specific actions to be executed within the process. These actions can include moving the tool to coordinates *{x, y, z}*, starting the spindle, adjusting the feed rate... Every program maintains a time series of runs, enabling the storage of multiple executions of the same program.

Furthermore, each block is associated with variables that capture physical values measured during the process using various sensors (such as vibration sensors, and power sensors) as well as contextual events (such as tool number). Each variable possesses its own time series for storing the values collected throughout the execution of the block.

In addition to updating the timeseries for each block, we also update a Gaussian profile to provide statistical information, including metrics such as minimum, maximum, average, and standard deviation of the values per block.

To instantiate the graph and time series-based model, we utilized GreyCat[1] and developed a specialized importer able of parsing CSV files line by line. This importer efficiently feeds the model with the corresponding values. By leveraging its functionality, the importer automatically identifies the start and end points of each program run, based on changes in the program identifier numbers. Consequently, it dynamically constructs block collections, incorporating the associated variables, and efficiently populates the variable time series with the corresponding values. This automated process ensures accurate and seamless integration of the dataset into the model. This process took 5 hours on a 4-core, 32GB ram virtual machine executed on a server equipped with an AMD EPYC 7662 64-Core processor. Following the completion of the import process, the GreyCat database achieved a weight of 27 gigabytes (GB) due to internal efficient compression techniques and the elimination of consecutive repetitions of identical data.

5.3 Visualizing the program fingerprint

To provide real-time visibility into incoming production data and its alignment with the fingerprint, we developed a web-based dashboard. This dashboard offers a user-friendly interface that enables to monitor and visualize the production data as it is generated, facilitating immediate insights into its correlation with the established fingerprint.

In Fig. 3, we can observe a variable alongside its respective fingerprints. The fingerprint provides information for each block, including the expected value (calculated as the average of all values observed for that specific block), the actual measurement, and the lower and upper fences (obtained by subtracting and adding the standard deviation from the average, respectively).

The variable deviates significantly from its expected value multiple times before an unexpected machine stop, indicated by the red area spanning over the curves. More importantly we see that the variable shows high peaks that significantly deviate from the expected value. Conversely, after the stop the variable consistently remains close to its expected value, indicating that the program is operating similarly to what has been observed thus far. With only two out-of-profile measurements that barely go outside the

[1] https://greycat.io/

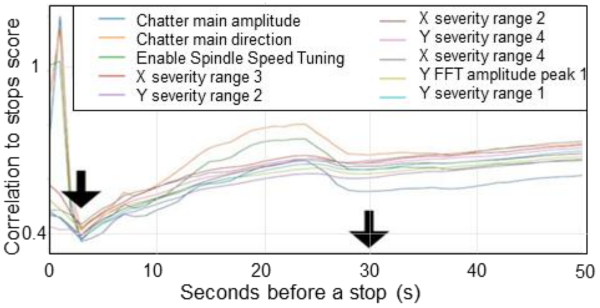

Fig. 4: Top 10 feature correlations before a marked stop.

fingerprint, this stable behaviour indicates a return to a reliable and consistent execution of the program.

The fingerprint approach proves to be highly valuable in providing real-time insights into the ongoing operations of a machine. While it can be constructed using historical datasets, its versatility allows for the incorporation of live data as well. This is achieved through the ability to update the fingerprint profile for each new record received. Consequently, this approach is particularly advantageous for manufacturers seeking to rapidly establish a monitoring process without the prerequisite of extensive historical data. By leveraging the fingerprint approach, manufacturers can quickly bootstrap their monitoring capabilities and gain immediate visibility into the performance of their machines, enhancing operational efficiency and enabling timely intervention when deviations occur.

However, the presence of a substantial historical dataset, including annotated machine stops, opens possibilities for alternative approaches aimed at predicting these marked stops. While the fingerprint approach primarily focuses on the current state of the machine, the integration of LSTM (Long Short-Term Memory) models can provide the ability to forecast the near future. By leveraging LSTM models, it becomes feasible to identify patterns and detect emerging conditions that may lead to a stop before it occurs. This proactive prediction empowers operators with advanced warnings, allowing them to take pre-emptive action if the conditions conducive to a stop start to converge. By combining the real-time insights from the fingerprint approach with predictive capabilities enabled by LSTM models, manufacturers can optimize their operational efficiency and minimize disruptions by addressing potential stoppage factors before they manifest.

5.4 Training the LSTM models

The data acquisition presented in section 3 revolves around a plethora of collected metrics that pertain to the monitored machining processes. A challenge toward training a model to predict future machining behavior from the recent one is identifying which metrics participate in the evolution of this behavior. Aiming at anticipating unplanned production stops, we can focus, in historical data, on metrics that temporally correlate to identified past stops.

More precisely, our objective is to define a selection of metrics as a set of training features for our machine learning model. For the training to have any chance of being efficient, we operate a temporal correlation analysis between pre-identified past stops in stored data on one hand, and all metrics up to one minute prior to said stops on the other hand. From there we identify a selection of the 20 features that temporally correlate the most to incoming stops, as illustrated in Fig. 4.

The figure plots a correlation score for the top 10 selected features depending on the time before a stop. The lower the score is, the most correlated the feature is to the incoming stop. In addition to allowing us to identify which metrics correlate the most to stops and make them our set of training features, this also illustrates that all these features consistently show their highest correlation two seconds prior to unplanned stops. While such a short time span is hardly practical in predicting and preventing stops in production, we could identify another correlation, lesser but as consistent among all selected features, 30 seconds prior to unplanned stops. This consistency is what led us to define the prediction lapse of 30 seconds to train our machine learning models.

To assess the performance of the trained model, a binary label is defined for each prediction. This label incorporates a coefficient, denoted as *c*, multiplied by the standard deviation *σ*. If any predicted feature exhibits a distance from its mean greater than *c* times its standard deviation, the prediction is labelled as positive. Conversely, if all predicted features remain sufficiently close to their average values, the prediction is labelled as negative. This labelling approach, known as out-of-profile (OOP), facilitates the comparison between predictions and the ground truth on test sets. It enables the classification of predictions as true or false positives, as well as true or false negatives. Using the OOP label, quantitative scores can be computed over the predictions on the testing set by comparing them with the ground truth. Examples of such scores include sensitivity and specificity, which provide insights into the accuracy and reliability of the predictions.

$$Sensitivity = \frac{\#\, true\ positives}{\#\, true\ positives + \#\, false\ negatives} \tag{4}$$

$$Specificity = \frac{\#\, true\ negatives}{\#\, true\ negatives + \#\, false\ positives} \tag{5}$$

Both scores are used to choose, among all possible values of *c*, the one that maximizes *Sensitivity* × *Specificity*. Once *c* is set, precision and F1 score (harmonic mean of *Sensitivity* and *Precision*) can be computed to get an insight as how well the model predicts behaviours of concern for the selected features:

$$Precision = \frac{\#\, true\ positives}{\#\, true\ positives + \#\, false\ positives} \tag{6}$$

$$F1\ score = \left(\frac{Sensitivity^{-1} + Precision^{-1}}{2}\right)^{-1} \tag{7}$$

As introduced in 4.2, the predicting model is implemented as the combination of two LSTM-based neural networks. Both networks operate on the same time window over the same features as inputs. However, both are trained to output different data:

1. The first network is trained to predict 30 seconds in the future the same features it is trained on. More technically, it is expected, once trained, to output a 20-dimensional prediction when fed a 20-dimensional sequence of inputs.
2. Over the same data, the second network is trained to predict the Euclidean distance to the mean, in the 20-dimensional feature space, of the datapoint to come 30 seconds in the future. Put more simply, it is expected to predict as a single figure how much the 20 features will be deviating from the normal behaviour overall.

The decision to train two neural networks instead of one stemmed from the rather poor efficiency of the first to predict, on historical data, actual out-of-profile behaviours, *i.e.,* behaviours that deviate significantly from the expected one. To tackle this, we devised an alternative approach that

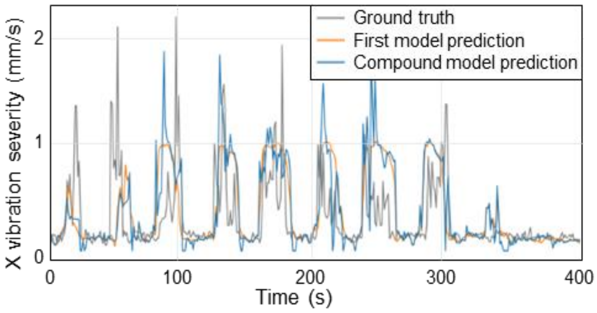

Fig. 5: Predictions for vibration severity in X direction.

focuses specifically on predicting this distance to the expected behaviour, *i.e.,* to the norm.

As for the combination of both models, the distance d_p to the mean predicted by the second network is used to rescale the features predicted by the first by adjusting their distance to the mean to match d_p. For the remaining of the document, we refer to the overall model as the compound model. Table 1 details quantitative scores computed for each neural network as well as for the compound model.

Fig. 5 gives a more visual insight at the quality of the prediction for the X vibration severity, by comparing ground truth (in grey), the 1st NN prediction (in orange) and the compound model prediction (in blue). The period selected here in historical data shows a succession of peaks in actual data, that consistently correspond to out-of-profile periods. As we can see here, the first model prediction, while efficient at predicting the overall shape of the curve, completely misses these peaks, hence cannot trigger the prediction of out-of-profile behaviours. On the other hand, the compound model is better at predicting most of the peaks. Combining such out-of-profile predictions with observed out-of-profile live data gives a high confidence that the machine's behaviour is degraded, confidence that the first model only cannot strengthen in this example.

5.5 Comparison with a GBDT approach

To compare our approach with the GBDT-based one described in [Zhang 2020], we reused the same dataset spanning over three months. We followed a consistent split for both approaches, using the first two months for training and the third month for testing the predictions.

To begin, we need to outline the working principles and inherent differences between the two approaches. One of the primary challenges encountered when training models to predict stops lies in the infrequent occurrence of these unplanned events. In our approach, we addressed this issue by focusing on features that exhibited the most significant deviations from their expected values as a stop arise. Our model was trained to predict these deviating features, which provided insights about abnormal or normal behaviours based on the extent of their deviations. This methodology allowed for improved level balance during training, consequently yielding enhanced results. However, it also introduced the possibility of occasional false identifications of upcoming stops if abnormal behaviours occurred without resulting in an actual stop.

Regarding the prediction output, the GBDT approach provided a probability value ranging from 0 to 1, representing the likelihood of a stop occurring. Given the infrequency of stops, the resulting probabilities exhibited a bias toward lower values, favouring the notion of no stop occurring. To mitigate this bias, the proposed solution employed the 95th percentile of computed scores, which corresponded to the 95% to 5% ratio of non-faulty to faulty periods. Similarly, our approach involved calculating a threshold based on a multiple of the standard deviation to optimize the sensitivity × specificity trade-off.

Tab. 1: Trained NN prediction description.

	1st NN	2nd NN	Compound model
Sensitivity	58%	67%	79%
Specificity	85%	69%	81%
Precision	46%	50%	65%

Table 2 provides a comprehensive overview of the results obtained from both approaches. Notably, the impact of stop rarity is apparent in the GBDT predictions, with a significant number of true negatives overshadowing other categories. This dominance of true negatives is also reflected in the specificity metric, indicating that the GBDT model excels at accurately predicting the absence of stops but struggles with sensitivity, as stop predictions are incorrect more than 50% of the time.

Conversely, the LSTM approach, which focuses on predicting feature behaviour, exhibits a more balanced distribution between negative and positive predictions. While this may result in a higher occurrence of false positives and false negatives, the model does not prioritize specificity at the expense of sensitivity. Instead, both metrics converge, suggesting that the model has learned to effectively differentiate between the two behaviours. This improved discrimination capability is further reflected in a three-fold increase in precision compared to the GBDT approach.

As for the prediction threshold, its definition for both approaches differs slightly. Indeed, on one hand GBDT is trained to predict a probability of a stop happening. On the other hand, our LSTM compound model is expected to predict the datapoint for the 20 features we analyse. It is then the distance of that datapoint to the mean, related to the standard deviation, that we use to decide whether the machine behaviour is degraded enough to be considered as leading to a stop. Hence, for GBDT the threshold is the percentage above which predictions are considered true, while for LSTM the threshold is expressed as a coefficient of the standard deviation of the training dataset in the 20-dimensional feature space. While for GBDT, the threshold is computed as the percentile corresponding to the number of unplanned stops in the training data, for LSTM we devise it as the distance to the mean that ultimately maximizes the F1 score. If both cannot be mathematically compared as they do not follow the same construction, the values themselves can be discussed. In fact, a threshold as low as 1.58% is a strong hint that the GBDT model will struggle at

Tab. 2: GBDT v. LSTM

	GBDT	LSTM
Prediction threshold	1.58%	4.62 std
True positives	1,889	187,604
False positives	9,280	124,157
True negatives	1,260,446	298,513
False negative	2,198	39,988
Sensitivity	46%	76%
Specificity	99%	77%
Precision	18%	59%
F1 score	25%	66%

predicting any high probability of a stop happening, which reflects in the imbalance between sensitivity and specificity. Conversely, with a threshold of 4.62 the standard deviation, the LSTM compound model proves its ability at predicting datapoints that strongly deviate from the mean, hence leading to a clear distinction between in-profile and out-of-profile predictions.

5.6 Threats to validity

Limited evaluation scope

The evaluation of our approach was performed on a single program and a single machine. While we believe that our approach is generic enough to apply to other use cases, specifically those involving repetitive time series data, further evaluation on a wider range of programs and machines is needed to validate its generalisability.

Data quality and availability

To train the LSTM models, we relied on a dataset containing four years of data with the unexpected stop already marked. However, it is uncommon to have access to such a long and flawless dataset. Before applying the approach presented in this paper for the LSTM training, a thorough data cleaning phase may be necessary to address any potential data quality issues or inconsistencies that could impact the learning phase.

Evolving State of the Art

At the time of our research, LSTM models were considered state-of-the-art for recurrent neural networks. However, transformer models [Vaswani 2017] have been introduced and outperform LSTM for predicting repetitive sequences. There is a possibility that using transformer models could yield even better results. Still, the results achieved with the LSTM approach were satisfactory, particularly in comparison to the GBDT approach.

Reproduction of the GBDT approach

To conduct a thorough comparison between our approach and the GBDT publication, we made a concerted effort to replicate the various data preparation steps described in the original paper. However, we encountered ambiguity regarding the specific methodology employed by the authors to rebalance the dataset, ensuring that unplanned stops constituted 5% of the data. In our experiments, we made the decision to retain the entire dataset. While this choice facilitated a more meaningful comparison, it is important to note that it may have an impact on the results obtained by the GBDT approach, as they might not accurately reflect the outcomes achieved with a meticulously cleaned dataset.

6 CONCLUSION

This article has presented a process control framework that combines both fingerprint and machine learning approaches to perform an automatic process surveillance. By uniting these two innovative techniques, the presented system enhances performance and offers improved predictive capabilities beyond what has been achieved by previous learning-based algorithms.

We experimentally show that this approach allows to avoid the risk of overfitting the absence of stops in predictions, a common pitfall when dealing with occurrences as rare as the unplanned stops observed in this use case. In fact, and for the same use case, our approach outperforms a previous one by a factor of more than 2.5 in terms of F1 score, which reflects how it performs significantly better at predicting actual stops while minimizing the raise of false negatives.

An aspect that cannot easily be tackled by LSTM neural networks, and recurrent neural networks in general, is the heterogeneity of behaviours that lead to unplanned stops, especially when these different behaviours differ in time span. Recent advances in the application of the transformer technology to not only natural language tasks, but also multivariate numerical time series such as the use case presented in this paper seem rather promises as an alternative approach to raise yet better results toward the prediction of unplanned production stops.

7 ACKNOWLEDGMENTS

This research was funded by the European Union’s Horizon 2020 Research and Innovation Programme under the project InterQ (grant agreement No. 958357).

8 REFERENCES

[Dare 2006] Dare, P. (2006). Linear and Nonlinear Models. Fixed effects, random effects, and mixed models. Geomatica, 60(4), 382-383.

[Gao 2020] Gao, R. X., Wang, L., Helu, M., & Teti, R. (2020). Big data analytics for smart factories of the future. CIRP annals, 69(2), 668-692.

[Hochreiter 1997] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

[Li 2019] Li, Y., Liu, C., Hua, J., Gao, J., & Maropoulos, P. (2019). A novel method for accurately monitoring and predicting tool wear under varying cutting conditions based on meta-learning. CIRP annals, 68(1), 487-490.

[Li 2022] Li, X., Liu, X., Yue, C., Liang, S. Y., & Wang, L. (2022). Systematic review on tool breakage monitoring techniques in machining operations. International Journal of Machine Tools and Manufacture, 103882.

[MacGregor 1995] MacGregor, J. F., & Kourti, T. (1995). Statistical process control of multivariate processes. Control engineering practice, 3(3), 403-414.

[Tangjitsitcharoen 2013] Tangjitsitcharoen, S., & Boranintr, V. (2013). Integration of in-process monitoring and statistical process control of surface roughness on CNC turning process. International Journal of Computer Integrated Manufacturing, 26(3), 227-236.

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

[Wang 2019] Wang, P., Liu, Z., Gao, R. X., & Guo, Y. (2019). Heterogeneous data-driven hybrid machine learning for tool condition prognosis. CIRP Annals, 68(1), 455-458.

[Xiao 2022] Xiao, W., Huang, J., Wang, B., & Ji, H. (2022). A systematic review of artificial intelligence in the detection of cutting tool breakage in machining operations. Measurement, 110748.

[Zhang 2020] Zhang, Y., Beudaert, X., Argandoña, J., Ratchev, S., & Munoa, J. (2020). A CPPS based on GBDT for predicting failure events in milling. The International Journal of Advanced Manufacturing Technology, 111, 341-357.

[Zhang 2023] Zhang, X., Gao, Y., Guo, Z., Zhang, W., Yin, J., & Zhao, W. (2023). Physical model-based tool wear and breakage monitoring in milling process. Mechanical Systems and Signal Processing, 184, 109641.