Using Time Series Forecasting Techniques to Predict Future Tool Wear Events

Rebecca Nixon (*Author*)
Computing, Engineering, and the Built Environment
University of Ulster
Belfast, Northern Ireland, UK.
nixon-r8@ulster.ac.uk

Jospeh Rafferty
School of Computing and Mathematics
University of Ulster
Belfast, Northern Ireland, UK.
j.rafferty@ulster.ac.uk

***Abstract*— This research explores a means of predicting future tool wear behaviour using low cost, open-source technologies, and real production data with an industry-adoptive focus. ARIMA regression and LSTM, BiLSTM and GRU neural network models were applied to the data and the forecasting ability compared with the GRU model offering the most favorable performance. In addition, a means of potentially offering real time, on-line indications of tool wear types currently acting on the tool is described. From the success of this preliminary work, the next phase will implement a system of part traceability and microscopy work to quantify the wear.**

Keywords—tool; wear; life; forecast; prediction; torque; manufacturing; ARIMA; LSTM; BiLSTM; GRU

I. Introduction

In the manufacturing industry surface finish, part quality, and maximizing machine uptime to improve productivity are priorities for machining managers. Premature tool changes have an inevitable negative impact on machine uptime but worn cutting tools, or more critically tool breakages, have a much more detrimental effect on the productivity losses. Equally worn tools reduce surface desirable finish and quality of a workpiece, by affecting the precision of the tool and increasing internal machine stress. Controlling the downtime of the machine through planned tool changes, irrespective of the premature nature, is a more welcome trade-off than the unknown of unplanned stops due to tool and/or machine damage. Moreover, the production of scrap material as a direct result of worn tools is similarly undesirable.

There has been much work in the area of monitoring tool wear and breakage. In particular, the extensive review by Li, Liu, Yue, Liang, and Wang is an excellent resource [1]. In many computer numerical control (CNC) machines, there is no means of directly monitoring tool wear. There is a metric for operators to assess the activity of the tool in time, correlating to wear, called Tool Life Remaining, which is the approximated tool life left in minutes for that tool. This metric is fixed at the outset and as such only considers time as a parameter, not the work that the tool has done. This means that if the tool encounters impurities in the raw material and works harder to machine the surface, this metric does not consider that circumstance as affecting the tool life when inevitably it does. Likewise, if the tool works less than expected this metric will lead to premature removal of the tool costing, both in terms of machine downtime and in monetary terms for regrind or replacement tool.

Due to the lack of unequivocal tool wear monitoring, researchers have devised many ways to indirectly monitor it. When a tool is active, the spindle that houses the tool emits noise, the changes of which researchers have correlated with tool wear [2]–[4]. Similarly physical responses such as spindle vibration, load experienced and energy consumed have frequently been used as a means of detecting changes in tool health because of wear [5]–[7].

These researchers have provided useful insights to the behaviour of CNC machines; however, these are off-line experiments requiring additional sensors, equipment and dedicated G-code programs suited to their respective experimental design. There have been some researchers who have attempted on-line methods of data extraction and analysis but again the experimental procedure is not practical for industrial adoption. Kious, Ouahabi, Boudraa, Serra and Cheknan [8] acknowledged the need for indirect methods to measure parameters during the machining process but they still used external equipment and machined a simple, non-production part. Wang, Hong, and Lin [9] achieved a more industry-relevant method of tool wear monitoring by using a communication protocol present on the CNC controller to monitor machine parameters; however, they did not use this technique to monitor real production data. Instead, understandably, they used an experimentally relevant G-code program specific to their work.

It is acknowledged that the reasons for prescriptive, dedicated G-code machine programs is to increase scientific rigor and generate robust results and conclusions. Equally, this method reduces risk to production and offers control to the researcher as well as offering insights that may not have been achievable otherwise. Since tools are prematurely changed and production is constantly being optimized to reduce the risk associated with tool wear and breakage, there is an inevitable lack of useful data available to researchers, creating an ongoing and persistent challenge for industrial researchers.

This research explores a means of predicting future tool wear behaviour that would impact wear rate and, therefore so to, the life of the tool. The methodology consists of a low cost, open-source, and non-intrusive method of monitoring machine production processes. The analysis utilizes machine learning algorithms to attempt to predict and forecast future machine parameters linked to tool wear of a Gundrill. The eventual goal is to accurately forecast this tool behaviour on-line, in real time, forecasting tool wear events which can be avoided through pre-emptive remedial action.

II. Background

This section offers a brief overview of tool wear and the relevant technologies used in this project.

A. Tool Wear

Tool wear is directly related to the cutting conditions of the machining process such as feed rate, cutting speed, spindle speed, depth of cut, temperature, material type etc. hence the difficulty in extracting the true state of tool wear unless examined under a microscope. There is a general equation used to describe the relationship between cutting conditions and tool life known as the Taylor equation, provided in (1). Fig 1. provides microscope images of some forms of tool wear that can occur on a gundrill whilst Fig 2 presents a schematic providing an overview of how tool wear may occur and the associated consequences.

$$C = v_c T_n \tag{1}$$

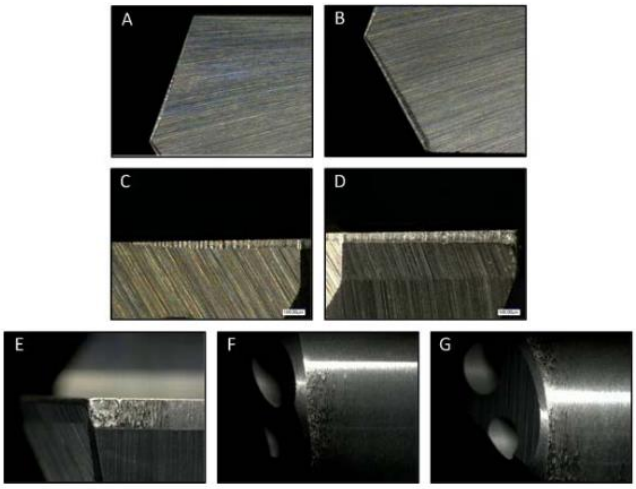

Fig. 1. Stable tool wear on gun drill tip, (a) Crater wear on inner cutting edge (b) Crater wear on outer cutting edge (c) Flank wear on inner cutting edge (d) Flank wear on outer cutting edge (e) Side margin wear (f) Lower bearing pad wear (g) Upper bearing pad wear (see online version for colours) [10].

B. Time Series Forecasting

Forecasting is a challenging facet of time series analysis. The complexity varies depending on the type of data and the context from which the data stems [11]. Compared to standard data analysis using machine learning, with time series, data cannot be randomly split into test and training as the sequence is important to the training. Likewise, the learning is regression-based but each data point can carry weight, lending meaning to the entire dataset such as seasonality. In this project Arima and three flavours of Recurrent Neural Networks were used: LSTM, BiLSTM and GRU.

1) Arima: Auto-Regressive Integrated Moving Average (ARIMA) is a traditional, linear regression-based time series data analysis technique, which performs well for short-term forecasts but performance deteriorates for long-term predictions [11]. It is for this reason, the three neural netowrks were implemented for comparison.

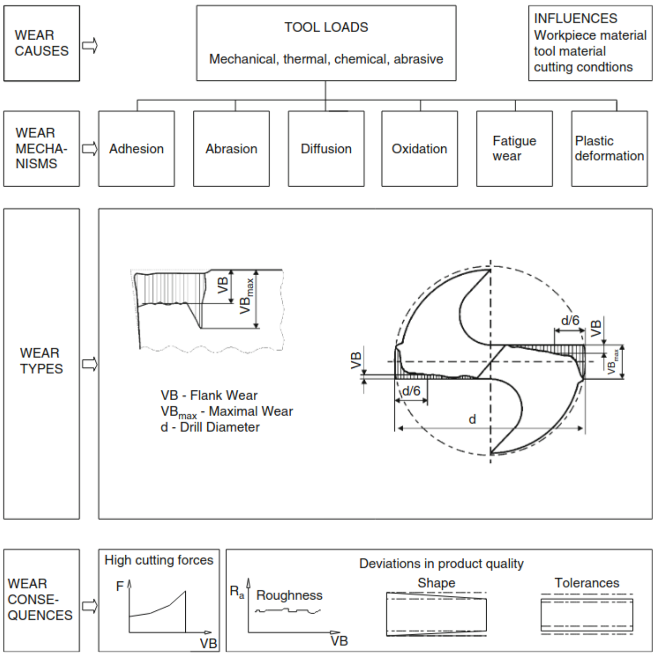

Fig. 2. The causes, mechanisms, types and mechanisms of tool wear [12]

2) LSTM: Long Short Term Memory (LSTM) is a neural network based on the Recurrent Neural Network (RNN) model. A downside to RNNs is that with each epoch (learning cycle) the weight of the network is updated with partial derivaties of the error function, which, if small can limit the training or halt it altogehter. Subsequently, with each iteration, the gradient of the model tends to decrease to zero. With LSTM this issue with the vanishing gradient problem is corrected. Each LSTM layer has a ‘forget’, ‘input’ and ‘output’ gate which augments the data according to perceived relevancy, discarding information deemed useless and guaranteeing constant error flow [13]. The complexity and learning capability of the model can be increased by, among others, increasing the number of hidden layers and/or neurons. The LSTM structure is presented in Fig 3.

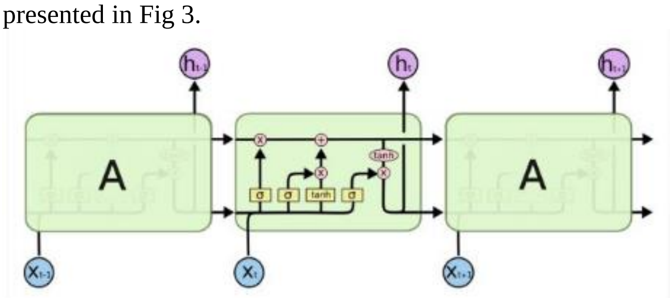

Fig. 3. LSTM Structure [13].

3) BiLSTM: Bidirectional LSTM (BiLSTM) is a regular LSTM model but instead of data flowing through the model in one direction, it flows back through the model again i.e left to right then right to left. This means that the network structure and learning is doubled since each network layer has the capability of accessing both future and past information. Having the capability of utlilising the data twice offers the potential benefit of improving predictive quality [14].

4) GRU: The Gated Recurrent Unit (GRU) model is a version of LSTM whereby the gating unit of the LSTM is merged i.e. the cell state and the hidden layer state are combined, as is depicted in Fig 4. Doing this increases model speed and reduces the processing required by reducing the number of parameters that need to be learned during model training [15].

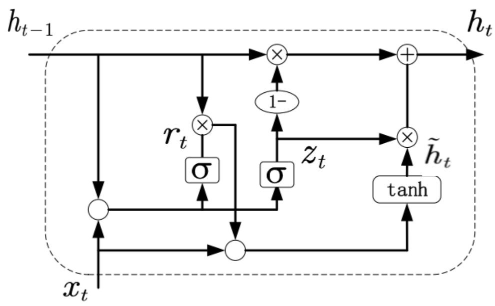

Fig. 4. GRU Structure [15]

III. Method

This section outlines the equipment used in this project and how the project was executed.

A. Equipment

The CNC machine in question is a 5-axis Index G220 Mill Turn Machine with a Siemens 840D sl Controller and S7300 PLC. The tool being investigated is a 11.8mm TiAln coated Carbide Flat Bottom Drill. Details of the part being manufactured cannot be disclosed due to confidentiality, but the material is Stainless Steel. Each production run produces 74 parts, and 4 shifts were analysed in this project with one being selected for focus. A Raspberry Pi 4 was used to run the software to communicate and extract the data from the machine.

B. Connectivity and Data Extraction

There are several communication protocols which have been reported in the literature, including MQTT, CoAP, MTConnect, as well as the Open Platform Communications Architecture (OPCUA), which was what was selected in this project [16]. OPCUA is an open-source communication protocol developed by the OPC foundation as a means of standardizing the many ways that manufacturing machines communicate [17]. An OPCUA server was available on the Siemens controller and provided access to the desired machine parameters.

The machine data was extracted using a Python script spinning up an OPCUA client and Mosquitto MQTT broker on a Raspberry Pi 4. The data was then published to the MQTT broker at a rate of 1Hz. Telegraf was used as a plugin for Influx DB to subscribe to the data and then store in the database. Data was then visualized through Grafana, which gave the operator and workshop floor management instant access to production information.

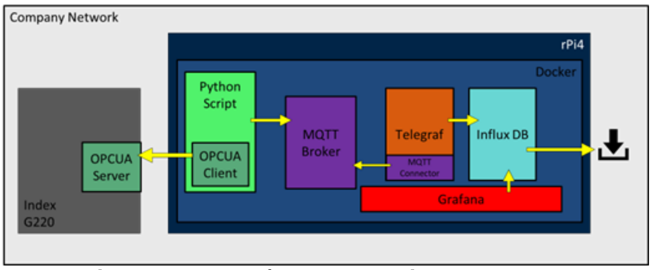

Fig. 5. Schematic overview of connectivity architecture

All these services were all hosted in a docker container, running on the Pi4. A schematic of the architecture is presented in Fig 5 and an example of the dashboard is presented in Fig 6.

Regarding security, TLS certification was used by means of OpenSSH and user authentication was implemented for all services. Furthermore, the entire system was run within a highly secure, firewalled company network with full knowledge from the Network Services and IT departments.

Fig. 6. Example dashboard presenting live production information using the Grafana service.

C. Data Analysis and Modelling

The data was extracted from the time series database Influx to RStudio where the desirable parameters for in-depth analysis were identified and some initial machine learning implemented. The data was then exported from R to Jupyter Notebook where Python was used to implement the time series analysis using ARIMA and the LSTM models using the Tensorflow and Keras packages.

The initial action after loading the data was cleaning and formatting into a readable format before then plotting various metrics in different forms. This plotting and wrangling were both for the raw data which consisted of approximately 60 data points per part, per tool, resulting in approximately 83,000 rows of data, and the summarized data for selected tools of interest. The summarization process condensed the data to one dataset per statistical metric, per tool and containing one value of this statistical metric per part. For instance, one shift, containing 83,000 rows of data was filtered for a range of tools, including the Gundrill, with a data frame created for each. The tool-specific data was then further reduced by taking the mean, median, standard deviation, median absolute error, etc. for each part and a separate data frame created for these statistical measures. Once the parameter of interest was identified and plotted to observe any evident patterns, a clustering algorithm was applied to determine if additional information could be obtained. The data was then exported to Jupyter Notebook to commence the time series forecasting.

Using Jupyter Notebook and Python with the relevant associated libraries, time series analysis was conducted on both the raw and summarized data with the goal of being able to forecast future values which may reveal insights regarding the future life of the tool. To assess forecasting performance the data for both sets, was trimmed so there was a dataset for training and testing and then a forecast for values equal to the length of the trimmed data. Performance of these models were made against each other and the baseline metrics of Naïve forecasting, Linear Regression and a simple RNN. The performance metric used was Root Mean Squared Error (RMSE). The repeatability of the models was tested by repeating the model fitting 30 times and the RMSE captured for comparison

IV. Results

A. Parameter Analysis

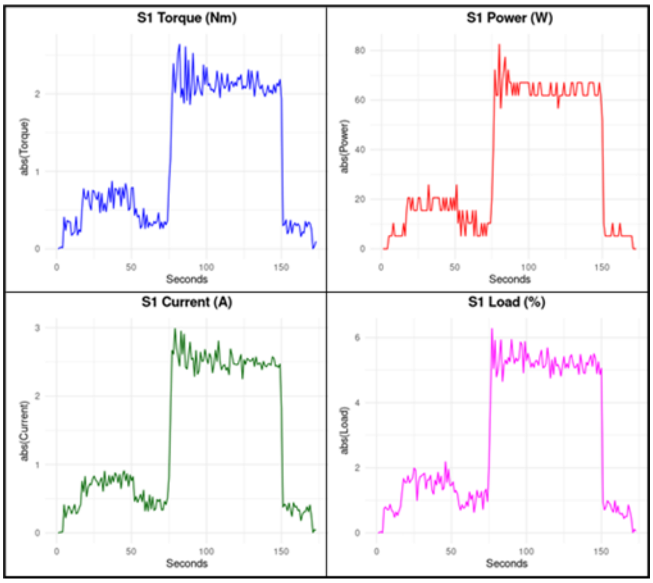

Fig. 7. Torque, Power, Current and Load for main milling spindle for one part over one shift.

The initial exploration of the data was to investigate what parameters could be used to infer information on the tool activity and therefore, life and wear behaviour. From the literature and domain knowledge it was deemed the most relevant features to explore were Torque, Load, Power and Current with the chosen tool being the Gundrill. Fig. 7. presents a plot of these features for one part in one shift for the main milling spindle. Due to the highly correlated plots in Fig. 7, a decision was made to focus on Torque moving forwards.

Torque is a parameter which is intuitively related to the work exerted on and by the substrate and therefore related to the wear experieced by the tool. The parameter 'load' refers to the percentage of total load that is exerted on the spindle [18] but with non-revealing units, it is difficult to ascertain whether this parameter is comparable to other machines.

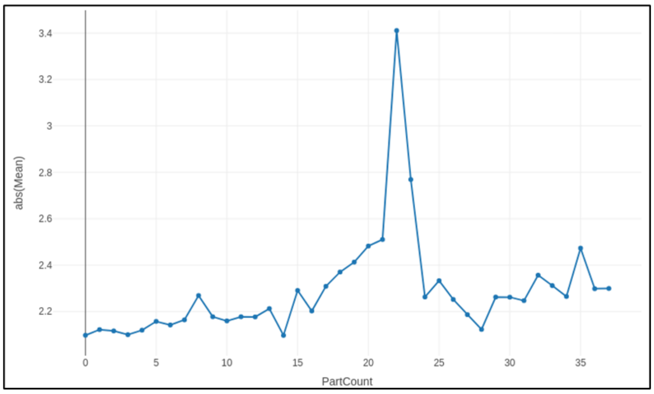

Fig. 8. Summarised mean values of Torque per part, experienced by the Gundrill over its life.

Likewise with energy consumption, these parameters will offer insights into machine performance but can be affected by external factors and/or clever power management systems which may be implemented by third parties.

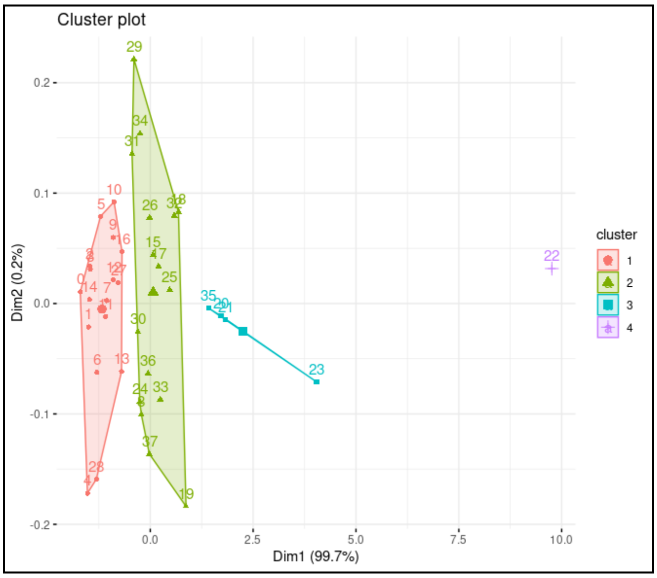

Fig. 9. Result of a k-means cluster algorithm applied to the summarised mean values of Torque. (No. of clusters = 4)

Despite knowing that the machine in question is a constant power machine, it may not be the case for others. Torque, on the other hand, is a reliably constant paramter in that it is the relationship between the force exerteed on/by the tool and the dimensions of the tool. Fig 8 presents the mean of Torque, per part, across the life of a single Gundrill in one shift. The number of parts available per Gundrill tool is 37 because the tool expires during the shift and a 'sister tool' is then loaded to continue with the production of 74 parts.

With the parameter and tool chosen and the data plotted, visually there was evdence that perhaps types of tool wear such as chipping was evident, denoted by sharp peaks in the data. A k-means clustering algorithm was applied to the data to determine if there were such events occuring. The results of the Scree Plot suggested 4 clusters were optimal; the resulting k-means cluster plot is presented in Fig 9.

B. *Time Series Forecasting*

The time series forecasting results are split into their relevant models and results for both the raw (if available) and the summarized will be displayed.

1) ARIMA: A comparison between the predicted torque values and the actual for the raw, are provided in Fig. 10 alongside the forecast outcome. SARIMAX was the model identified as the most optimal by the Auto Arima optimiser function from the pmdarima library. Fig. 11 presents the Predicted vs Actual plots for the summarised data using SARIMAX including the forecast comparison with the test data. The figure also presents the forecast predicted by the model.

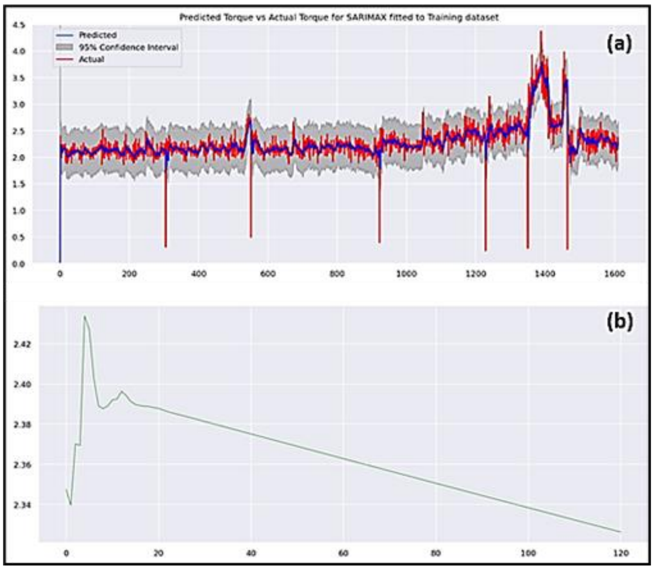

Fig. 10. **(a)** Predicted Torque vs Actual Torque values using SARIMAX (8,0,0)(0,0,0). **(b)** Forecast prediction for the test dataset of 120 values.

2) Neural Networks: A comparison of the summarised Predicted Torque values and the Actual values for each of the models is presented in Fig 12. In addition there is a plot of the predicted values and the forecast of the Torque values using SARIMAX but without the actual, for comparison with the other models.

3) Evaluation of Performance: Table 1 presents the Root Mean Square Error values for the different models and datasets, i.e. raw or summarised. For clarity, these RMSE values are the result of comparing the Train, Valid and Forecast values predicted by the model, to the actual values obtained. Included in this table are the RMSE results for the baseline metrics to compare how the models selected fair against more vanilla versions. Fig. 13 presents boxplots of the reliability of the models after being repeated 30 times.

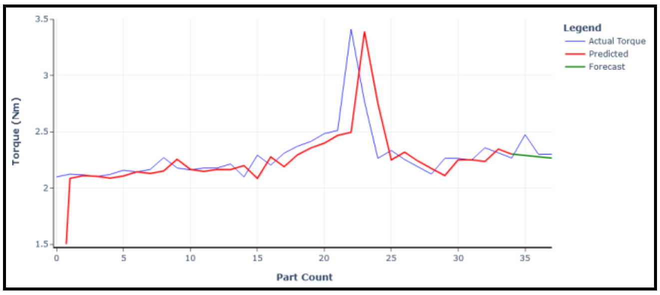

Fig. 11. Comparison of the Predicted and Actual summarised Torque values for all data with parts 35-37 forecasted using SARIMAX(1,0,0)(0,0,0).

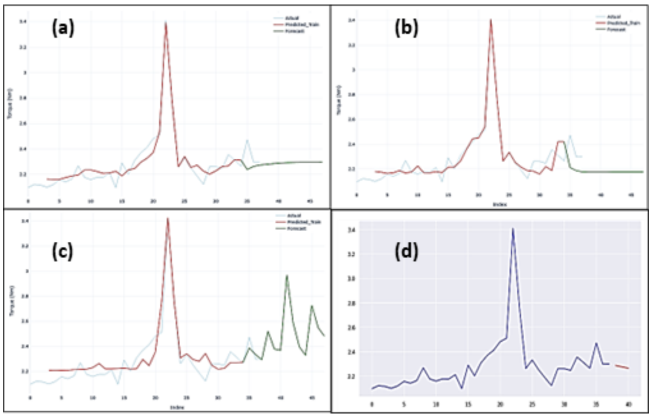

Fig. 12. Comparison of the Predicted and Actual summarised Torque values as well as the forecast for the data + 10 more values using **(a)** LSTM; **(b)** BiLSTM; **(c)** GRU; **(d)** the predicted values for entire summarised dataset including the forecast value for the data modelled using SARIMAX(1,0,0)(0,0,0).

TABLE I. RMSE VALUES FOR ALL MODELS FOR BOTH RAW AND SUMMARISED DATA

Model	Raw			Summarised		
	Train	Valid	Forecast	Train	Valid	Forecast
Naïve	-	0.070	-	-	0.110	-
Linear Regression	-	0.060	-	-	0.060	-
Simple RNN	-	0.060	-	-	1.690	-
ARIMA	0.249	0.248	0.248	0.460	0.080	0.110
LSTM	0.212	0.217	0.330	0.060	0.061	0.137
BiLSTM	-	-	-	0.064	0.111	0.179
GRU	0.167	0.222	0.310	0.090	0.093	0.054

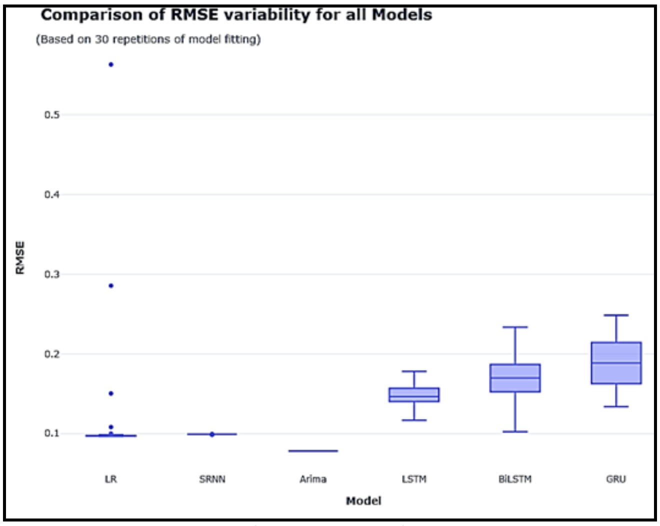

Fig. 13. **(a)** Comparison of the variability of the selected models (based on 30 repeats of model fit).

V. Discussion

The data presented in this paper covers three sections: (a) parameter analysis and identification; (b) time series analysis; (c) performance evaluation. These three sections will be discussed separately, in order. Table 2 lists the hyperparameters chosen for each of the models. For the neural network models, these were chosen through exhaustive trial and error. There is scope to continue the trial and error to improve the performance further.

A. Parameter Analysis and Identification

The goal of this initial part of the data analysis was to assess what parameters could be used to infer information on the tool activity and therefore, life and wear behaviour. After consulting the literature and subsequently narrowing down the parameters of interest and tool of choice, plots of these parameters were created for the selected tool for one part to explore behaviour. As evident in Fig.7, the selected parameters were equally similar in profile meaning that the choice of parameter to progress with was at the discretion of the researcher. Torque was selected as the parameter to focus on.

The focus was further refined to the life of one tool, which in real terms was half a shift with part count = 37. The resulting plots in Fig. 8 visually indicate activity or 'Torque Events'. Since this is data extracted during real production, the process could not be stopped and tool examined after each spike to assess what form of wear was occurring, if any. Furthermore, with a current lack of traceability of the part it was impossible to identify the parts associated with the Torque Events and examine the relevant parts for indications of wear. As a result, it was decided to find a way of quantitatively and automatically identifying these Torque Events and their associated parts for future reference when traceability has been implemented. Two additional benefits to this are that, if successful, it could potentially be a means of conducting on-line monitoring of live production data where classifications of tool wear can be instantly assigned, and parts flagged for part quality inspection. This would lead to being able to historically review the frequency of certain classifications during one production run or many. In turn, this could expose useful information for supply chain managers if a batch of material or tools are differing in behaviour. Similarly, production managers would benefit from this for means of process optimization due to a quicker and simpler method of attributing process parameters to degree or type of tool wear, which ultimately could lead to an improvement in part quality, reduction in downtime, scrap material and other costs associated with tool wear.

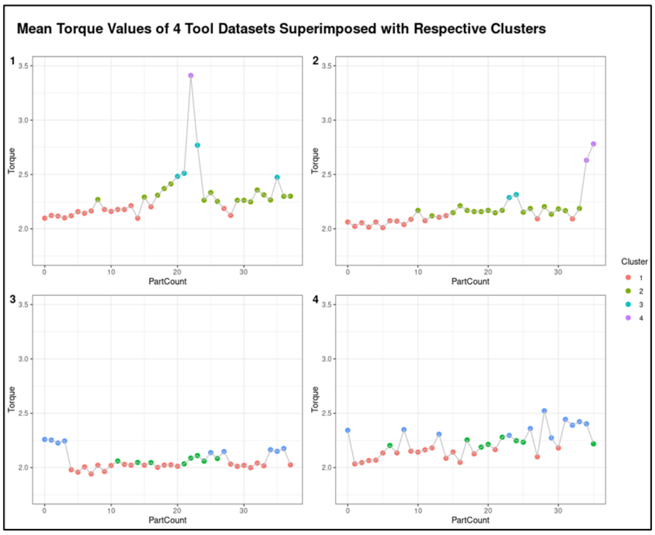

Fig. 14. Results of the k-means cluster analysis applied to the mean Torque values from four datasets of four different tools over two shifts.

A k-means clustering algorithm was chosen as a means of attempting to identify the observable Torque Events and developing a classification structure for these events. A scree plot was generated, Fig. 14, and it was determined that the optimal number of clusters was 4. The results of the k-means cluster analysis using a cluster number of 4, annotated with the respective part count, is presented in Fig. 9, and correlates with the Torque plot in Fig. 8. With a perfect identification between the clusters and the observable Torque Events, the clustering algorithm was then applied to the four other datasets. The results of this analysis were then superimposed onto the respective Torque plots to confirm perfect matching; these plots are presented in Fig. 14 and reveals success in identifying the observable peaks, which could be attributed to wear events. This is an extremely useful outcome for the next phase of work.

B. Time Series Analysis

For the time series analysis, it was decided to continue with the univariate format that had been completed in the parameter identification section. Future work will extend this to investigate a multivariate dataset incorporating other production parameters for analysis.

TABLE II. LIST OF HYPERPARAMETERS CHOSEN FOR THE MACHINE LEARNING ALGORITHMS

		Raw				Summarised			
		SARIMAX	LSTM	BiLSTM	GRU	SARIMAX	LSTM	BiLSTM	GRU
Hyperparameters	p	8	-	-	-	1	-	-	-
	d	0	-	-	-	0	-	-	-
	q	0	-	-	-	0	-	-	-
	Scaler	Standard	MinMax [0,1]	MinMax [0,1]	MinMax [0,1]	MinMax [-1,1]	MinMax [-1,1]	MinMax [-1,1]	MinMax [-1,1]
	Layers	-	2	2	2	-	2	2	2
	Neurons		64		512	-	256	256	512
		-	32		512	-	256	256	512
	Window	-	10	-	10	-	3	3	3
	Epochs	-	100		1000	-	1000	1000	1000
	Batch Size	-	100		100	-	3	3	3
	Activation	-	ReLu	-	ReLu	-	tanh	tanh	tanh
	Optimiser		Default Adam	Default Adam	Default Adam	-	Default Adam	Default Adam	Default Adam
	Dropout		0.2	0.2	0.2	-	0.5	0.5	0.8
			0.2	0.2	0.2	-	0.5	0.5	0.8
	Shuffle	-	FALSE	FALSE	FALSE	-	FALSE	FALSE	FALSE
	Stateful	-	FALSE	FALSE	FALSE	-	FALSE	FALSE	FALSE

1) ARIMA : ARIMA is a descriptive acronym for Autoregression (AR) Integrated (I) Moving Average (MA). These three sections of the name have integer values attributed to them, denoted by letters, p,d,q respectively, to indicate the parameters of the model being run. Typically the model will be referred to with the letters succeding the name; for insance in this work, for the raw values SARIMAX(8,0,0) was used and for the summarised data SARIMAX(1,0,0).

For ARIMA to work the data needs to be stationary, which means there is no seasonal trend or inherent pattern i.e., the mean, variance and auto correlation etc. are constant over time. There was confidence that this data was stationary and having plotted the relevant autocorrelation and differencing plots, alongside a Dickey-Fuller test, the results confirmed this to be the case. The results of the Dickey-Fuller test are presented in Fig. 15; note the p-value << 0.05 and the adf value = -5.7 which is less than -3.4 at 1% meaning we can reject the null hypothesis that the data is not stationary with a confidence level of less than 1% i.e., it is highly unlikely that this is a statistical fluke.

The Auto Arima optimiser function from the pmdarima library automatically discovers the optimal order for the ARIMA model [19]. For both the raw and summarized data it suggested Seasonal Auto-Regressive Integrated Moving Average with eXogenous factors (SARIMAX), which is an ARIMA for non-stationary data. However, the model was run with default check for seasonality and therefore returned a SARIMAX model but in addition, the (P,D,Q) factors to be used for non-stationary data were all zero, confirming the prior analysis that this data is in fact stationary.

```
In [50]: result=adfuller(df['Torque'])
         dict(zip(['adf', 'pvalue', 'usedlag', 'nobs', 'critical' 'values', 'icbest'],result))

Out[50]: {'adf': -5.660732463343433,
          'pvalue': 9.393170434591657e-07,
          'usedlag': 7,
          'nobs': 2295,
          'criticalvalues': {'1%': -3.433202561761398,
           '5%': -2.8628001971599955,
           '10%': -2.5674408601155303},
          'icbest': -357.28779713719086}
```

Fig. 15. Results of the Dickey-Fuller test

Visually the model fit well for the raw data, but Table 1 proves that it was the worst performer of all models. Conversely, it was the second-best performer for the summarized data. One noticeable aspect was the appearance of a lag = 1 in the predictions. This ‘delay’ is believed to be because of the MA component of the model whereby it is weighting, or dampening, the past Torque Events tending to the Mean. With the Autoregressive value, p, equal to 1 it is simply replicating the data. The forecast results, for parts 35, 36 and 37 together create a straight line with a downward trend and do not fit the peak of the actual data. It is believed that if the model were to forecast more values beyond 37, it would continue the straight line downwards. It is likely that more sample data would assist in model performance.

2) Neural Networks: These models differ from ARIMA and other linear regression-based models in that they do not attempt to predict the future based on the past behaviour of the same parameter. Instead they attempt to ‘learn’ the behaviour of the system and use that to predict. Since the data in this work was univariate, a target had to be generated somehow. This was done by converting the data into a supervised format by duplicating the column and then shifting the duplicated column by the window length +1. This shifted, duplicated column was now the target column.

LSTM was applied to the raw data first but as with ARIMA there are too many data points in one part count for the model to forecast the next part. It was decided to focus on modeling and forecasting with the summarised data instead.

From the RMSE scores in Table 1, none of the models perform as favourable as the baseline metric of Linear Regression at predicting with BiLSTM perfomring the worst comparatively. However, the plots demonstrate a favourable fit in pattern, which is adequate if searching for breaches in threshold or attributing classifications. Nonetheless, the goal was for forecasting capbility. None of the models seemed to perform as hoped but of all the models, the prefernce is for GRU. This preference is due to the other models seeming to struggle at forecasting values that are probable. It is expected that if the tool was not changed and worked for more parts, the Torque will trend upwards as more wear accumulates. Only GRU demonstrates this likelihood and is reflected in the forecast RMSE scores where it is more than 50% less than the rest.

One thing to note in this work is the variation of hyperparamters. The hyperparameters were chosen by trial and error including the activation change with a reduction on available data from the Rectified Linear Unit (ReLU) to the hyperbolic tangent (tanh) for the summarised data. With these changes, the range of the scaler function was amended accordingly.

C. Performance Evaluation: It was noticed that each time the code was run, the forecast values for the GRU model changed with more variation that for the other models. This is reflected in the boxplots in Fig. 13. Despite this variation, the forecasted values were still significantly more probable than those forecast by the other models. The train and validation RMSE scores were more comparable than with the other models, particularly ARIMA, indicating that the prediction accuracy of the model has the same spread of variance as the training data and that the model was a decent fit. The ARIMA model however, had a much higher RMSE for the training than the validation, suggesting it was undertrained.

VI. Conclusion

The goal of this research was to explore a means of predicting future tool wear behaviour that would impact wear rate and life of the tool using low cost, open-source technologies, and real production data with an industry-adoptive focus.

The tool chosen was a 11.8 mm Gundrill and Torque was used as the parameter for prediction. A k-means cluster analysis was performed on the data and the result was a means of potentially being able to offer real time, on-line indications of classes of tool wear occurring and therefore a measurable means of evaluating effects of attempts at remedial action.

ARIMA, LSTM, BiLSTM and GRU models were applied to the data to explore the possibility of accurately forecasting future Torque values. The future data was a trimmed portion of the original data against which the forecasts were compared. The GRU model was deemed the best performer despite having the larger variation in results after 30 repeats. It offered the lowest forecast RMSE and offered the more probable result scenario when used to forecast further beyond the available dataset.

VII. Future Work

- Implement a system of traceability so that parts identified by the clustering algorithm can be checked for indications of wear and associated types.
- Examine tools with a microscope to build an image database to correlate with the production run data and to take dimensional measurements.
- Repeat the research but using the contextual production data to create a multivariate dataset for the neural networks to utilise their unique ability to 'learn' and 'memorise' patterns.
- Attempt further optimization of the GRU model including transfer of model weights and vector outputs.

References

[1] X. Li, X. Liu, C. Yue, S. Y. Liang, and L. Wang, "Systematic review on tool breakage monitoring techniques in machining operations," *Int. J. Mach. Tools Manuf.*, vol. 176, p. 103882, 2022.

[2] N. Seemuang, T. McLeay, and T. Slatter, "Using spindle noise to monitor tool wear in a turning process," *Int. J. Adv. Manuf. Technol.*, vol. 86, no. 9–12, pp. 2781–2790, Oct. 2016.

[3] A. Duo, R. Basagoiti, P. J. Arrazola, J. Aperribay, and M. Cuesta, "The capacity of statistical features extracted from multiple signals to predict tool wear in the drilling process," *Int. J. Adv. Manuf. Technol.*, vol. 102, no. 5–8, pp. 2133–2146, Jun. 2019.

[4] S. Charoenprasit, N. Seemuang, and T. Slatter, "Monitoring tool wear in drilling process using spindle noise features," *Int. J. Mech. Eng. Robot. Res.*, vol. 7, no. 5, pp. 564–568, Sep. 2018.

[5] T. Xi, I. M. Benincá, S. Kehne, M. Fey, and C. Brecher, "Tool wear monitoring in roughing and finishing processes based on machine internal data," *Int. J. Adv. Manuf. Technol.*, vol. 113, no. 11–12, pp. 3543–3554, Apr. 2021.

[6] E. Ahearne, S. Baron, S. Keaveney, D. P. Dowling, and G. Byrne, "Tool Wear in Milling of Medical Grade Cobalt Chromium Alloy - Requirements for Advanced Process Monitoring and Data Analytics," 2016.

[7] C. Liu, X. Xu, Q. Peng, and Z. Zhou, "MTConnect-based Cyber-Physical Machine Tool: a case study," *Procedia CIRP*, vol. 72, pp. 492–497, Jan. 2018.

[8] M. Kious, A. Ouahabi, M. Boudraa, R. Serra, and A. Cheknane, "Detection process approach of tool wear in high speed milling," *Measurement*, vol. 43, no. 10, pp. 1439–1446, Dec. 2010.

[9] D. Wang, R. Hong, and X. Lin, "A method for predicting hobbing tool wear based on CNC real-time monitoring data and deep learning," *Precis. Eng.*, vol. 72, pp. 847–857, Nov. 2021.

[10] K. S. Woon, S. Kanno, and K. Liu, "The influence of drill geometric accuracy on steady tool wear development and catastrophic tool failure in high aspect ratio gundrilling of Inconel-718," *Int. J. Abras. Technol.*, vol. 6, no. 1, p. 1, 2013.

[11] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "A Comparative Analysis of Forecasting Financial Time Series Using ARIMA, LSTM, and BiLSTM." arXiv, 2019.

[12] J. Agapiou and D. A. Stephenson, "Cutting Tool Wear and Failure Mechanisms," in *Encyclopedia of Tribology*, Springer US, 2013, pp. 661–677.

[13] Q. Fang, Y. Zhong, C. Xie, H. Zhang, and S. Li, "Research on PCA-LSTM-based Short-term Load Forecasting Method," in *IOP Conference Series: Earth and Environmental Science*, 2020, vol. 495, no. 1, p. 012015.

[14] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "The Performance of LSTM and BiLSTM in Forecasting Time Series," in *Proceedings - 2019 IEEE International Conference on Big Data, Big Data 2019*, 2019, pp. 3285–3292.

[15] H. He, H. Wang, H. Ma, X. Liu, Y. Jia, and G. Gong, "Research on Short-term Power Load Forecasting Based on Bi-GRU," in *Journal of Physics: Conference Series*, 2020, vol. 1639, no. 1.

[16] E. P. Hinchy, N. P. O'Dowd, and C. T. McCarthy, "Using open-source microcontrollers to enable digital twin communication for smart manufacturing," *Procedia Manuf.*, vol. 38, pp. 1213–1219, Jan. 2019.

[17] "Unified Architecture - OPC Foundation." [Online]. Available: https://opcfoundation.org/about/opc-technologies/opc-ua/. [Accessed: 07-May-2022].

[18] M. Sadegh Amalnik, "Optimization of Spindle loading and Tool Wear for CNC Turning Machine by Using Intelligent System," vol. 6, no. 3, 2017.

[19] T. G. Smith, "pmdarima.arima.auto_arima," 2022. [Online]. Available: https://alkaline-ml.com/pmdarima/modules/generated/pmdarima.arima.auto_arima.html. [Accessed: 09-May-2022].