Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Evaluating 1D CNN-Based Sensor Informatics for Run-Level Surface Roughness Prediction from Dynamometer Signals in Ti-6Al-4V Turning under Varying Lubrication Conditions

Lidiya Priyadarsini Korsapati

lidiyapriyadarsinik@gmail.com

Jadavpur University Faculty of Engineering and Technology https://orcid.org/0000-0002-7143-6755

Abhishek Mandal

Jadavpur University Faculty of Engineering and Technology

Rajat Subhra Sen

Jadavpur University Faculty of Engineering and Technology

Research Article

Keywords: Ti-6Al-4V turning, Machining data analytics, 1D temporal CNN, Surface roughness prediction, Dynamometer signals

Posted Date: November 18th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11599759/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Evaluating 1D CNN-Based Sensor Informatics for Run-Level Surface Roughness Prediction from Dynamometer Signals in Ti-6Al-4V Turning under Varying Lubrication Conditions

Mrs. Lidiya Priyadarsini K (Corresponding Author)
Research Scholar, Department of Mechanical Engineering, Jadavpur University, Kolkata 700032 WB, India
lidiyapriyadarsinik@gmail.com
https://orcid.org/0000-0002-7143-6755

Dr Abhishek Mandal
Assistant Professor, Department of Mechanical Engineering, Jadavpur University, Kolkata 700032 WB, India
amandal.mech@jadavpuruniversity.in

Dr Rajat Subhra Sen
Professor, Department of Mechanical Engineering, Jadavpur University, Kolkata 700032 WB, India
rajat_sen64@yahoo.com

Evaluating 1D CNN-Based Sensor Informatics for Run-Level Surface Roughness Prediction from Dynamometer Signals in Ti-6Al-4V Turning under Varying Lubrication Conditions

Abstract

Surface roughness (Ra) is an important quality indicator for Ti-6Al-4V components in aerospace and biomedical applications. Although stylus profilometry provides reference measurements of the finished surface, it does not provide continuous feedback during cutting. Dynamometer signals capture the mechanical response at the tool-workpiece interface, but the predictive value of their learned temporal representations remains insufficiently investigated. The present study evaluates run-level Ra prediction across 54 Ti-6Al-4V turning runs under dry, minimum-quantity lubrication (MQL) and nano-MQL (NMQL) conditions. Four neural configurations, namely machining parameters alone, parameters with dynamic root-mean-square (RMS) features, parameters with a one-dimensional convolutional neural network (1D CNN), and parameters with both RMS and CNN features, are compared with parameter-only tree models using run-disjoint evaluation. Across five repeated five-fold partitions, the hybrid model achieves mean MAE and RMSE values of 0.292 and 0.388 μm, respectively, without improving pooled RMSE over parameter-only gradient boosting or the RMS-based neural comparator. Its strongest temporal-branch benefit occurs in the nine-run MQL subset, with mean $R^2 = 0.829$ and RMSE of 0.178 μm, representing RMSE reductions of 44.6% and 26.1% relative to gradient boosting and parameters plus RMS. Gradient boosting performs best under dry conditions, whereas parameters plus RMS performs best under NMQL. These results support the feasibility of run-level prediction while showing that learned temporal features do not provide a uniform advantage. By integrating machining parameters with learned representations of densely sampled dynamometer signals, the framework provides a methodological foundation for sensor informatics and future machining big-data analytics that can be extended to other machining processes.

Keywords: Ti-6Al-4V turning; Machining data analytics; 1D temporal CNN; Surface roughness prediction; Dynamometer signals

1. Introduction

Surface roughness (Ra) is a primary quality criterion for turned Ti-6Al-4V components, because it governs fatigue initiation, sealing and wear in aerospace and biomedical parts [1, 2]. The alloy's low thermal conductivity, chemical reactivity and low elastic modulus make Ra difficult to control [1, 2]. Feed sets the kinematic surface profile and is usually the dominant factor [3, 4]; cutting speed and depth of cut act through temperature and force, and flank wear can outweigh the prescribed parameters once it develops [5]. Lubrication also alters the surface: minimum quantity lubrication (MQL), nanofluid MQL (NMQL) and hybrid cryogenic strategies each change Ra relative to dry cutting [6, 7].

Early Ra models were response-surface regressions on speed, feed and depth of cut [3, 4]. Neural networks fitted Ra more closely than multiple regression [8, 9], although their accuracy depends on network configuration [10] and has mostly been demonstrated under a single coolant condition [11]. Tree ensembles are now a strong reference: random forests were competitive with neural networks and support vector regression [12] and were the most accurate learners on small Ti-6Al-4V datasets [5, 13].

Any benefit attributed to process signals should therefore be measured against tuned tree models, not only against a parameter-only neural network.

Prescribed parameters describe what the machine was instructed to do, whereas process signals record what occurred at the cut, including tool wear, chatter and lubrication effectiveness [14]. Dynamometer forces have been used as additional inputs for Ra prediction in turning [15], with low-frequency radial and tangential components carrying most of the surface-finish information [16]. Vibration is more common because accelerometers are cheaper and less intrusive [17-19], while acoustic emission, spindle current and controller signals appear mainly in grinding and milling [20-22]. Most studies summarise signals as time-domain statistics, spectral descriptors or decomposition features [16, 18, 23, 24], often followed by feature selection [20, 24], and compact feature sets combined with machining parameters have performed well [25, 26]. Deep models instead learn the representation [27, 28]: 1D CNNs have predicted Ra from vibration and sound [21, 29, 30] or classified roughness from raw machine signals [31], and other studies convert signals into time-frequency or Gramian images for 2D CNNs [22, 32] or analyse photographs of the finished surface [33].

The comparative evidence is nevertheless uneven. An explicit parameters-only versus parameters-plus-signal comparison appears in [34], whereas most studies compare sensors or architectures [20, 21, 24, 29], and some report classification accuracy [24, 31] that cannot be compared with RMSE or MAE for continuous Ra. Attributing a benefit to learned temporal features requires an input ablation (parameters only, plus a simple signal summary, plus learned features, and plus both) on identical runs and splits. Evaluation design matters equally: windowing turns a few physical runs into many observations, and if windows from one run fall on both sides of a split, test scores reflect represented runs rather than unseen ones [35, 36]. Lubrication regime is also handled inconsistently: some models are built for a single condition [11, 34], others include regime as an input [13], and most report only pooled accuracy, so how a CNN's benefit varies between dry, MQL and NMQL cutting [6, 7] remains insufficiently characterised.

This study therefore evaluates, for dynamometer-instrumented Ti-6Al-4V turning under dry, MQL and NMQL conditions, whether a 1D CNN force branch improves run-level Ra prediction over parameter-only and parameter-plus-RMS models and over tuned tree ensembles, using run-disjoint evaluation, repeated partitions and regime-specific errors.

2. Materials and methods

Run-level Ra in Ti-6Al-4V turning was predicted from known machining parameters and dynamometer signals. Neural input-ablation models were compared with parameter-only tree models under run-disjoint cross-validation as per the elaborate workflow diagram given in Fig. 1.

2.1. Experimental setup and dataset

Cylindrical Ti-6Al-4V specimens were turned on an American Pacemaker lathe using PVD-coated tungsten carbide inserts with a 0.8 mm nose radius (Table 1, Fig. 2). Cutting forces were measured with a Kistler dynamometer and 5167A charge amplifier (Section 2.2), and roughness with a MarSurf M 310 stylus instrument. For MQL, a castor-oil emulsion (water : castor oil = 90 : 10) was atomised at an air pressure of 6 bar through a 2 mm nozzle with a 60° spray cone angle. For NMQL, Graphene nanoparticles, purity > 99%, thickness 5-10 nm, bulk density 0.166 g cm^{-3} were dispersed in the same emulsion. Machining parameter levels are listed in Table 2.

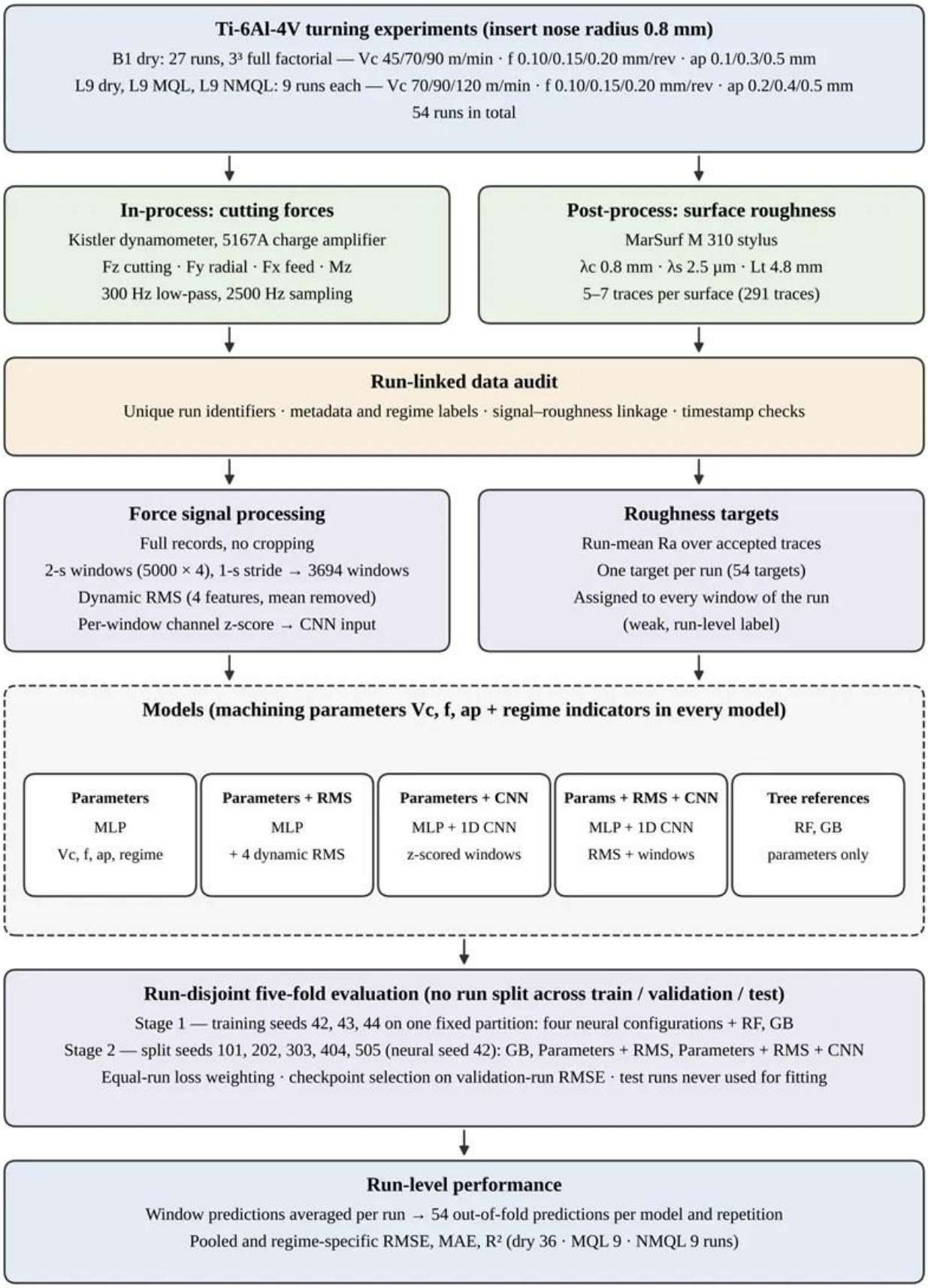


Fig. 1 Overview of the experimental, data-processing, modelling and evaluation workflow.

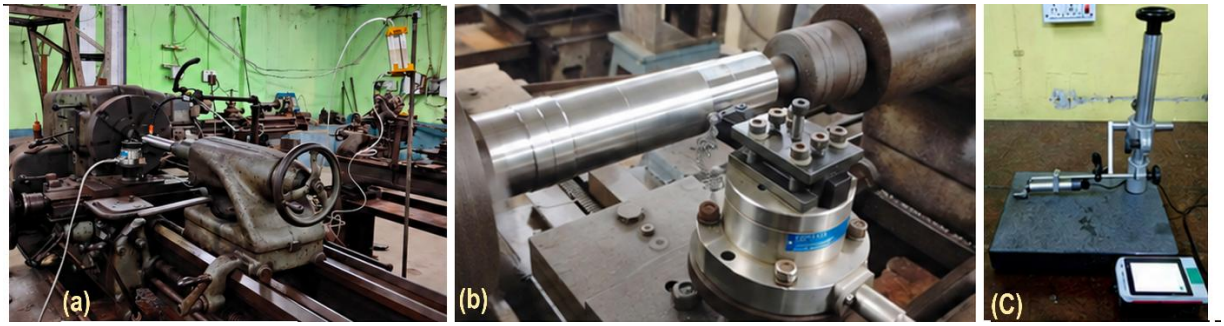


Fig. 2 Experimental setup: (a) lathe with the dynamometer and MQL/NMQL delivery system; (b) tool holder mounted on the dynamometer; (c) stylus profilometer used for roughness measurement

Table 1. Workpiece, tooling, instrumentation and lubrication conditions.

Item	Specification
Machine tool	American Pacemaker lathe
Workpiece	Ti-6Al-4V cylindrical specimens
Cutting insert	PVD-coated tungsten carbide; nose radius 0.8 mm
Dynamometer and amplifier	Kistler dynamometer; Kistler 5167A charge amplifier (time constant Long, not reset before each cut)
Force channels	F_z cutting, F_y radial, F_x feed, M_z
Data acquisition	300 Hz low-pass filter; 2500 Hz sampling
MQL system	Castor-oil emulsion (water : castor oil 90 : 10); air pressure 6 bar; nozzle diameter 2 mm; spray cone angle 60°, flowrate 25ml/hr
Nanofluid (NMQL)	Purity > 99%; thickness 5-10 nm; bulk density 0.166 g cm^{-3}; dispersed in the MQL emulsion
Roughness tester	MarSurf M 310; traversing length 4.8 mm; λ_c 0.8 mm (ISO 16610-21); λ_s 2.5 µm; 5-7 traces per surface

Table 2. Machining parameter levels in the B1 and L9 experimental blocks.

Parameter	Unit	B1 dry block (27 runs)	L9 blocks (dry, MQL, NMQL)
Cutting speed, V_c (nominal)	m/min	45, 70, 90	70, 90, 120
Feed, f	mm/rev	0.10, 0.15, 0.20	0.10, 0.15, 0.20
Depth of cut, a_p	mm	0.1, 0.3, 0.5	0.2, 0.4, 0.5
Design	-	3^3 full factorial	Taguchi L9 (same array for each regime)

The dataset contained 54 physical machining runs, as in Table 3, under dry, MQL and NMQL conditions. The dry subset comprised a 27-run B1 block and a nine-run L9 dry block, and the MQL and NMQL subsets each comprised nine L9 runs.

Table 3. Experimental blocks and recorded roughness traces.

Experimental block	Runs	Recorded Ra traces
B1 dry	27	142
L9 dry	9	51
L9 MQL	9	46
L9 NMQL	9	52
Total	54	291

The parameter vector comprised nominal cutting speed (V_c), feed (f), depth of cut (a_p) and lubrication regime. Regime was included in every parameter-based configuration and encoded using three binary indicators ordered as dry, MQL and NMQL, so the parameter-only input contained three continuous variables and three regime indicators. Parameters, signal files and roughness measurements were linked by run identifier. All 54 run labels were verified against acquisition timestamps, handwritten notes and design sheets, including the nano-MQL block labels, the trace direction of the nano-MQL specimens and the reassignment of batch-1 runs, and all 54 runs were retained.

Surface roughness was measured with the MarSurf M 310 stylus instrument (traversing length 4.8 mm, λ_c 0.8 mm per ISO 16610-21, λ_s 2.5 µm) using five to seven successive traces along the machined cylindrical surface. For run r, the reference target was the arithmetic mean of its accepted trace-level Ra measurements:

$$\mathrm{Ra}_r = \frac{1}{m_r}\sum_{j=1}^{m_r} \mathrm{Ra}_{r,j} \tag{1}$$

where m_r is the number of accepted roughness traces for run r and $\mathrm{Ra}_{r,j}$ is its jth trace measurement. Each run contributed one unweighted target, irrespective of its trace count. Trace-to-trace repeatability was assessed with a one-way random-effects intraclass correlation over the 52 runs with at least two traces (289 traces): ICC(1,1) = 0.87 for single traces and ICC(1,k) = 0.97 for the run-mean target.

Every signal window from a run was assigned that run's mean Ra. This is a weak label: it characterises the run-level surface rather than the local surface under each 2-s window, and windows were not aligned with trace positions.

2.2. Dynamometer signal processing and features

Dynamometer records were sampled at 2500 Hz through the Kistler 5167A charge amplifier with a 300 Hz low-pass filter on four channels: F_z (cutting force), F_y (radial force), F_x (feed force) and M_z. Each run was checked for a unique identifier, complete metadata, a valid regime label, finite values, a matching signal file and strictly increasing timestamps at the nominal 1/2500 s interval; files shorter than one window were rejected.

Full recordings were used without time-based cropping or duration normalisation, because machining began under automatic feed and initial face-side contact and penetration form part of the recorded process. The charge amplifier was operated with a long time constant and was not reset before each cut, so the quasi-static force level drifts slowly within a record in Fig. 3. Absolute force levels were therefore not used as features; per-window mean removal (Eqs. 2-4) suppresses this drift.

Each four-channel recording was segmented into overlapping windows of 5000 samples with a stride of 2500 samples, corresponding at 2500 Hz to a 2-s window, a 1-s step and 50% overlap. Only complete windows were retained; trailing incomplete portions were not padded. The 54 recordings produced 3694 windows, each of dimension 5000 × 4.

For channel c in a window of N samples $x_{i,c}$, the window mean and the dynamic RMS were computed from the unstandardised samples; the dynamic RMS is therefore the population standard deviation of the window:

$$\mu_c = \frac{1}{N}\sum_{i=1}^{N} x_{i,c} \tag{2}$$

$$\mathrm{RMS}_c = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(x_{i,c} - \mu_c\right)^2} \tag{3}$$

This produced four amplitude features per window. Separately, the temporal input was standardised channel-wise within each window:

$$z_{i,c} = \frac{x_{i,c} - \mu_c}{s_c} \tag{4}$$

where $s_c = RMS_c$, or $s_c = 1$ when $RMS_c < 10^{-8}$. Amplitude information therefore reaches only the numerical branch, while the temporal branch receives standardised waveform structure.

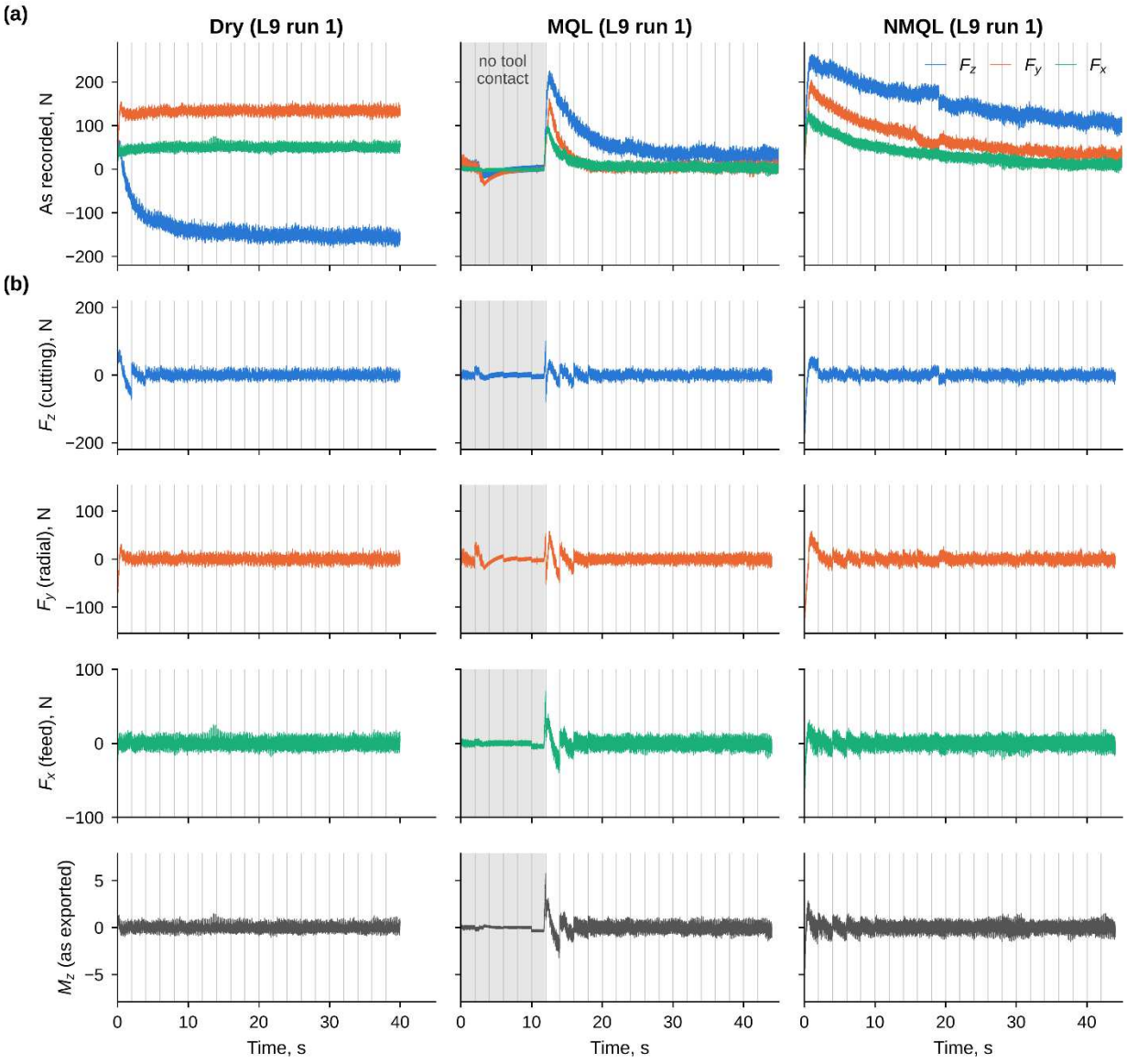


Fig. 3 Dynamometer signals for L9 run 1 under dry, MQL and NMQL conditions (Vc = 70 m/min, f = 0.20 mm/rev, ap = 0.5 mm). (a) Forces as recorded; the slow change in level is charge-amplifier drift. (b) Fz, Fy, Fx and Mz after per-window mean removal (Eqs. 2–4), as supplied to the models. Vertical lines mark 2-s windows; the 50%-overlapping windows are omitted for clarity. Shading marks the interval before tool–workpiece contact

2.3. Numerical preprocessing and equal-run weighting

Continuous numerical inputs were standardised using a scaler fitted exclusively on the training subset of each outer fold. For the RMS configurations, the continuous variables were V_c, f, a_p and the four

dynamic RMS features; the three regime indicators were appended without scaling, giving ten numerical input features.

To prevent longer recordings from dominating the training loss, each window of training run r was weighted by $1/n_r$, where n_r is its number of windows, and the weights were normalised to a mean of one. Each training run therefore carried equal total weight, and the scaler was fitted with the same weights.

2.4. Model configurations

Four neural configurations given in Table 4 were evaluated to distinguish the effects of numerical amplitude features and standardised temporal inputs.

Table 4. Neural input-ablation configurations.

Configuration	Numerical branch	Temporal branch
Parameters	V_c, f, a_p; three regime indicators	None
Parameters + RMS	Parameters; four dynamic RMS features	None
Parameters + CNN	Parameters	Standardised 5000 × 4 window
Parameters + RMS + CNN	Parameters; four dynamic RMS features	Standardised 5000 × 4 window

The numerical branch consisted of fully connected layers with 32 and 16 units and rectified linear unit (ReLU) activations. When present, the temporal branch comprised three one-dimensional convolutional layers. The first used 16 filters, a kernel width of 15 and a stride of two, followed by max pooling with a pool size of four; the second used 32 filters and a kernel width of nine, followed by max pooling of size four; and the third used 32 filters and a kernel width of five. All convolutional layers used same padding and ReLU activation, and global average pooling and dropout at a rate of 0.25 completed the temporal branch as in Fig. 4.

For temporal configurations, the numerical and temporal representations were concatenated. The fused representation passed through a 32-unit ReLU layer and dropout at a rate of 0.20, followed by a single linear output for Ra. Configurations without a temporal branch passed the numerical representation directly to the same prediction head. Hidden dense and convolutional layers used L2 regularisation with a coefficient of 10^{-4}. Adding the temporal branch also changes model capacity; the input ablation was therefore not a parameter-count-matched experiment.

Parameter-only Random Forest (RF) and gradient boosting (GB) regressors were evaluated using the same run partitions as the neural models. Each physical run supplied one training observation containing nominal cutting speed, feed, depth of cut and three lubrication-regime indicators; no numerical scaling was applied to the tree models. RF used 300 trees, with maximum depth selected from {2, 4, unrestricted} and minimum samples per leaf from {1, 2, 4}; bootstrap sampling was enabled, and all input features were considered at each split. GB used {100, 300} boosting stages, learning rates of {0.03, 0.10}, maximum tree depths of {1, 2} and minimum samples per leaf of {2, 4}, with squared-error loss and a subsampling fraction of 1.0. In Stage 1, the resulting nine RF and sixteen GB candidates were evaluated separately for each fold and seed (42, 43 and 44), and the candidate with the lowest validation-run RMSE was retained without refitting on the combined training and validation sets. In Stage 2, GB was selected in the same way with seed 42, and RF was not included. The training-run mean Ra served as a naïve baseline.

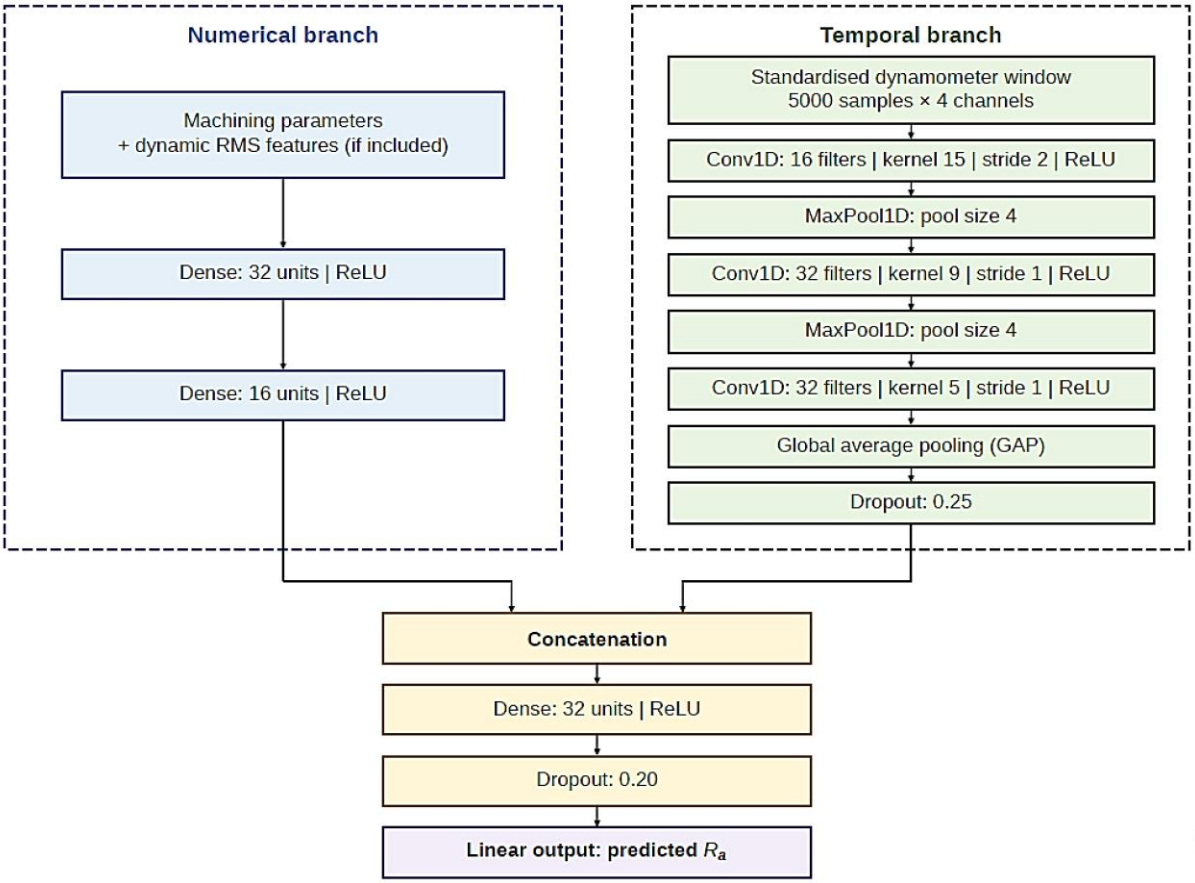


Fig. 4 Architecture of the two-branch network (Parameters + RMS + CNN configuration).

2.5. Evaluation protocol and training

The physical run was the unit of splitting and scoring: all windows from a run stayed in the same subset, so overlapping windows never crossed subset boundaries. Outer test runs were excluded from fitting, scaling, checkpoint selection, learning-rate adaptation and hyperparameter search. Run identifiers were permuted by a split seed into five outer test folds (four of 11 runs and one of 10), and 20% of each fold's development runs were held out for validation using a group-based split (seed incremented by fold number). Partitions were not stratified by regime or block, and split manifests were saved before fitting.

Neural models were trained with Adam (initial learning rate 3×10^{-4}, batch size 32, up to 150 epochs) on sample-weighted mean squared error plus the L2 penalties, with training windows shuffled. After each epoch, validation-window predictions were averaged per run, and the checkpoint with the lowest validation-run RMSE was retained for testing. The learning rate was halved after seven epochs without improvement (minimum 10^{-6}), and training stopped after 20. Neither neural nor tree models were refitted on training plus validation runs.

Two evaluation stages were performed. Stage 1 trained all four neural configurations with seeds 42, 43 and 44 on one fixed five-fold partition, alongside the tree models, to measure sensitivity to training randomness. Stage 2 measured sensitivity to run assignment using five new split seeds (101, 202, 303, 404 and 505), each defining a complete five-fold evaluation, with the neural seed fixed at 42. It compared parameter-only GB, Parameters + RMS and Parameters + RMS + CNN (75 outer evaluations; 810 run-level predictions). Model sets and seeds were fixed in advance, and all repetitions were retained. Both stages reuse the same 54 runs, so their standard deviations describe seed or partition

variability rather than uncertainty across independent datasets, and Stage 2 is an internal sensitivity check rather than external validation.

2.6. Performance metrics and regime analysis

For neural models, the final prediction for held-out run r was the arithmetic mean of its n_r window predictions:

$$\widehat{\mathrm{Ra}}_r = \frac{1}{n_r}\sum_{i=1}^{n_r} \widehat{\mathrm{Ra}}_{r,i} \tag{5}$$

Tree models predicted each run directly. Out-of-fold predictions from the five test folds were pooled into 54 run-level predictions per model and repetition, and scores were calculated from these rather than by averaging fold scores or treating windows as independent observations:

$$\mathrm{RMSE} = \sqrt{\frac{1}{R}\sum_{r=1}^{R}\left(\mathrm{Ra}_r - \widehat{\mathrm{Ra}}_r\right)^2} \tag{6}$$

$$\mathrm{MAE} = \frac{1}{R}\sum_{r=1}^{R}\left|\mathrm{Ra}_r - \widehat{\mathrm{Ra}}_r\right| \tag{7}$$

$$R^2 = 1 - \frac{\sum_{r=1}^{R}\left(\mathrm{Ra}_r - \widehat{\mathrm{Ra}}_r\right)^2}{\sum_{r=1}^{R}\left(\mathrm{Ra}_r - \overline{\mathrm{Ra}}\right)^2} \tag{8}$$

where R is the number of evaluated runs and the overbar denotes the mean measured Ra of those runs. RMSE and MAE are reported in micrometres. Seed-level and repetition-level scores were summarised by their arithmetic mean and sample standard deviation.

For each partition repetition and model, the out-of-fold predictions were grouped into dry (36 runs), MQL (nine runs) and NMQL (nine runs) subsets; R^2, RMSE and MAE were calculated within each subset and then averaged across the five repetitions. Model differences were compared on the same held-out runs within a repetition, and the number of repetitions favouring each model was used to describe consistency. Relative RMSE reductions were calculated from the across-repetition mean RMSE values as

$$\Delta\mathrm{RMSE} = 100 \times \frac{\mathrm{RMSE}_{\mathrm{ref}} - \mathrm{RMSE}_{\mathrm{cmp}}}{\mathrm{RMSE}_{\mathrm{ref}}}\,\% \tag{9}$$

Regime analyses were treated as exploratory because the MQL and NMQL subsets each contained only nine runs, and no statistical significance was inferred from aggregate metric differences.

3. Results

Results are reported separately for the two evaluation stages described in Section 2.5: Stage 1 (three training seeds on a fixed partition) and Stage 2 (five run partitions with the neural seed fixed at 42).

3.1. Neural input ablation

Table 5 summarises the neural input-ablation experiment. Values are the mean and sample standard deviation of pooled out-of-fold metrics across three training seeds.

Table 5. Run-level prediction performance across three training seeds on fixed run partitions (Stage 1).

Input configuration	R^2	RMSE (µm)	MAE (µm)
Parameters	0.5349 ± 0.0436	0.4045 ± 0.0189	0.3326 ± 0.0188

Input configuration	R^2	RMSE (μm)	MAE (μm)
Parameters + RMS	0.5757 ± 0.0798	0.3855 ± 0.0357	0.2950 ± 0.0296
Parameters + CNN	0.5585 ± 0.0292	0.3942 ± 0.0130	0.3050 ± 0.0069
Parameters + RMS + CNN	0.5812 ± 0.0273	0.3839 ± 0.0125	0.2989 ± 0.0171

All three signal-informed configurations achieved lower mean RMSE and MAE than the parameter-only neural model. The combined RMS-CNN configuration achieved the highest mean R^2 and lowest mean RMSE, whereas Parameters + RMS achieved the lowest mean MAE. The difference between the combined model and Parameters + RMS was small: their mean RMSE values were 0.3839 and 0.3855 μm, respectively. The combined configuration improved both RMSE and MAE over the parameter-only neural model in all three training seeds, but its RMSE advantage over Parameters + RMS occurred in only one of the three seeds. Similarly, adding the CNN without RMS improved RMSE over the parameter-only neural model in two of three seeds. The ablation results therefore supported signal-informed prediction but did not establish a consistent incremental benefit from the temporal branch over RMS features.

For reference, the training-run-mean baseline achieved a pooled RMSE of 0.6090 μm, MAE of 0.5136 μm and $R^2 = -0.0529$ on these partitions; all evaluated neural configurations outperformed this baseline.

Fig. 5 compares the run-level predictions of the four configurations with measured Ra. In every panel the predictions are compressed towards the mean: the slope of predicted on measured Ra was 0.55-0.66, 76-88% of runs with Ra below 0.9 μm were over-predicted, and all four runs above 2.3 μm were under-predicted. The clearest effect of adding signal information was in the NMQL block, where block RMSE fell from 0.476 μm for the parameter-only MLP to 0.355-0.416 μm for the signal-informed configurations. In contrast, several L9 dry runs were strongly under-predicted by every configuration, and the L9 dry block had negative R^2 in all cases (−0.02 to −0.17), compared with 0.52-0.59 for the B1 block. The vertical bars show that the training seed alone changed individual run predictions by a median of 0.20-0.24 μm, a substantial fraction of the prediction error. The horizontal bars show that measured Ra also varied between traces: the median trace-to-trace standard deviation was 0.13 μm, about one third of the pooled RMSE, which indicates the measurement variability against which prediction errors should be judged.

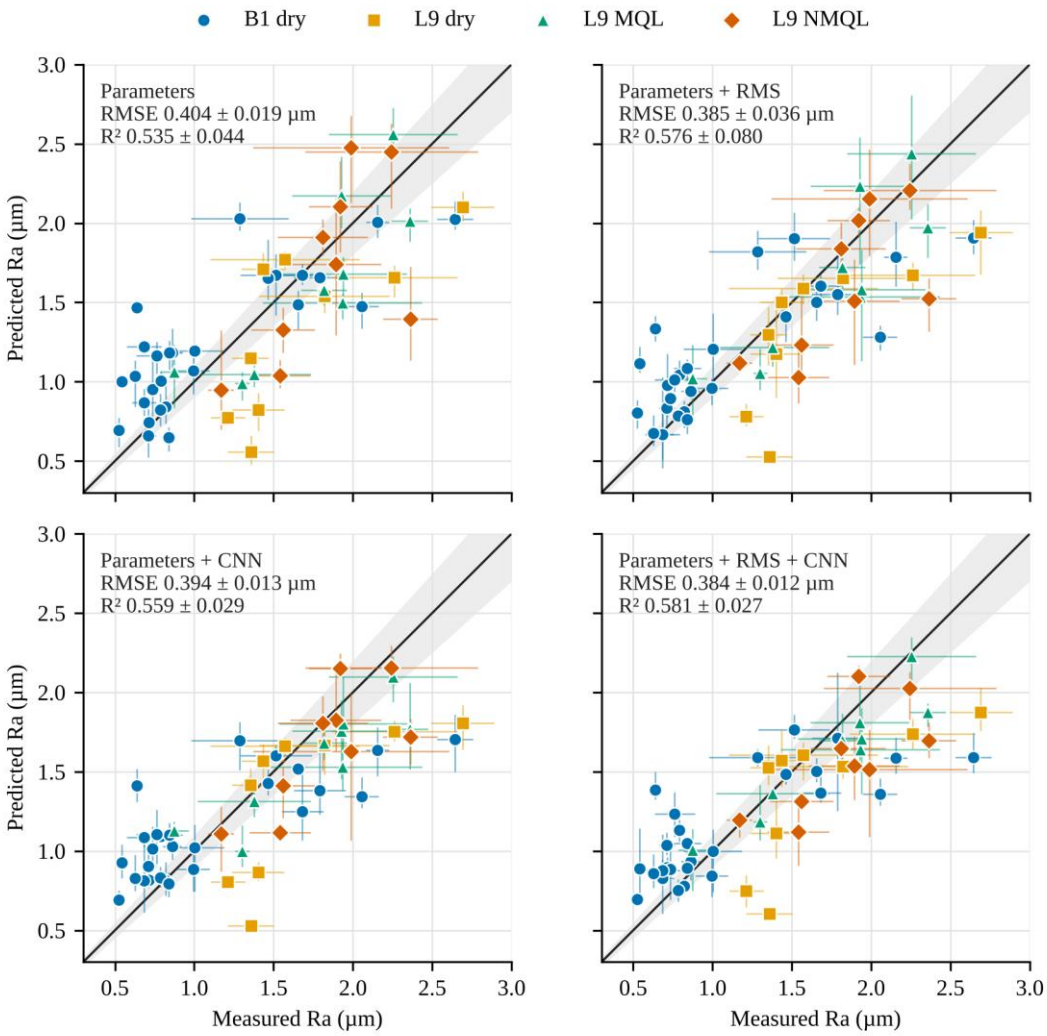


Fig. 5 Predicted versus measured run-level Ra for the four neural configurations (Stage 1). Points are the mean out-of-fold prediction over seeds 42-44; vertical bars span the three seeds and horizontal bars show the trace-to-trace standard deviation of measured Ra. The line is y = x and the shaded band ±10%. Panel scores are the mean ± SD of seed-level pooled metrics (Table 5).

3.2. Comparison with parameter-only tree models

Averaged across the three evaluated seeds, Random Forest achieved a pooled R^2 of 0.5589, RMSE of 0.3941 µm and MAE of 0.3075 µm, and gradient boosting achieved a pooled R^2 of 0.5804, RMSE of 0.3844 µm and MAE of 0.3022 µm. Gradient boosting therefore performed substantially better than the parameter-only neural model and closely matched the combined RMS-CNN model, which achieved mean RMSE and MAE values of 0.3839 and 0.2989 µm. These small aggregate differences did not establish a clear overall advantage for the signal-informed hybrid over a strong parameter-only comparator.

Fig. 6 places these averages in the context of individual seeds. The ranking of the neural configurations changed between seeds: Parameters + CNN gave the lowest RMSE for seed 42, whereas Parameters + RMS did so for seeds 43 and 44. Parameters + RMS also showed the widest seed-to-seed range (RMSE 0.358-0.426 µm) and the hybrid the narrowest (0.373-0.397 µm). The ranges of both configurations spanned the gradient-boosting reference, whereas the parameter-only MLP remained above it for every

seed (0.386-0.424 µm). The differences among the signal-informed configurations in Table 5 therefore lie within the variation caused by training randomness alone.

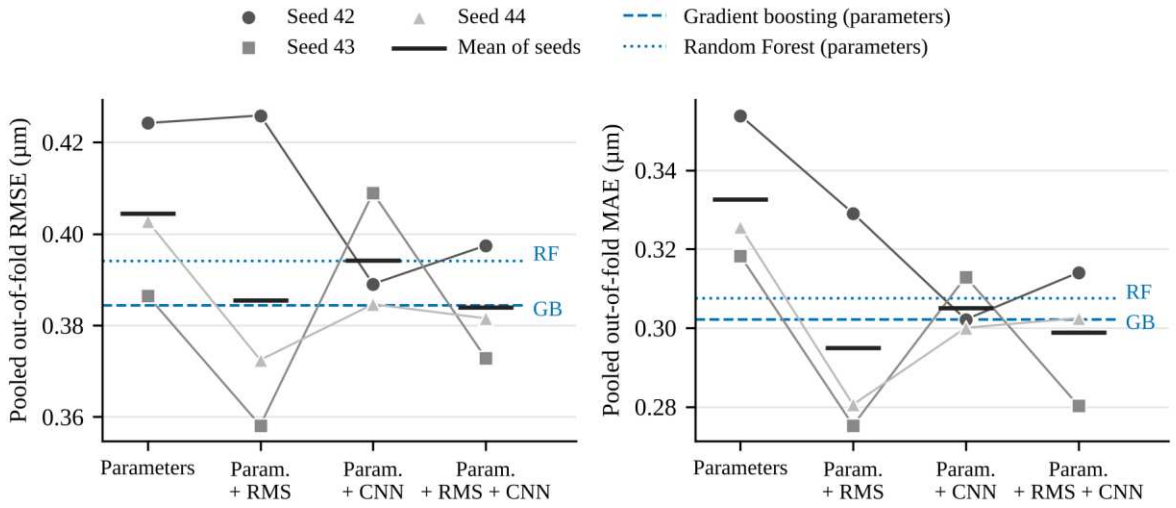


Fig. 6 Pooled out-of-fold RMSE and MAE of the four neural configurations for each training seed on the fixed Stage 1 partition. Lines connect results from the same seed; black bars are seed means. Horizontal lines mark the parameter-only gradient-boosting and Random Forest references averaged over the same seeds.

3.3. Sensitivity to run-partition assignment

The repeated-partition experiment examined whether the model comparison persisted under five new assignments of runs to training, validation and test subsets. Table 6 reports the mean and sample standard deviation of the five repetition-level pooled out-of-fold scores.

Table 6. Repeated five-fold run-level prediction performance with the neural training seed fixed at 42 (Stage 2).

Model	R^2	RMSE (µm)	MAE (µm)
Parameter-only gradient boosting	0.5929 ± 0.0597	0.3779 ± 0.0275	0.2972 ± 0.0296
Parameters + RMS	0.5936 ± 0.0375	0.3780 ± 0.0177	0.2940 ± 0.0187
Parameters + RMS + CNN	0.5734 ± 0.0252	0.3875 ± 0.0113	0.2918 ± 0.0092

The ranking depended on the evaluation metric. Gradient boosting and Parameters + RMS were practically tied on RMSE (0.3779 and 0.3780 µm) and R^2 (0.5929 and 0.5936), and the hybrid achieved the lowest mean MAE. The hybrid's mean MAE was lower than those of gradient boosting and Parameters + RMS by 0.0054 and 0.0022 µm, respectively, but its mean RMSE was higher than both. Across individual repetitions, the hybrid achieved lower pooled RMSE than gradient boosting in two of five repetitions and lower RMSE than Parameters + RMS in two of five. Its pooled R^2 ranged from 0.5315 to 0.5979, and its RMSE from 0.3763 to 0.4062 µm. These results demonstrated repeatable predictive capability under different run assignments without establishing consistent RMSE superiority.

The hybrid exhibited the smallest across-repetition standard deviations for all three metrics, indicating lower variation in its aggregate scores across the five tested partitions, although this does not establish robustness to changes in equipment or operating conditions. Because the repetitions reuse the same 54

runs, the reported standard deviations describe partition sensitivity rather than uncertainty across independent experimental datasets.

3.4. Regime-specific prediction performance

Regime-level evaluation revealed differences that were not apparent from the pooled metrics. Table 7 reports the mean of the five repetition-level metrics for dry (36 runs), MQL (nine runs) and NMQL (nine runs) conditions.

Table 7. Mean regime-specific out-of-fold performance across five partition repetitions (Stage 2).

Regime	Model	R^2	RMSE (µm)	MAE (µm)
Dry	Parameter-only gradient boosting	0.6054	0.3694	0.2874
	Parameters + RMS	0.5073	0.4135	0.3210
	Parameters + RMS + CNN	0.4704	0.4284	0.3293
MQL	Parameter-only gradient boosting	0.4841	0.3210	0.2668
	Parameters + RMS	0.7114	0.2409	0.2059
	Parameters + RMS + CNN	0.8293	0.1779	0.1448
NMQL	Parameter-only gradient boosting	−0.7166	0.4536	0.3668
	Parameters + RMS	0.0432	0.3354	0.2741
	Parameters + RMS + CNN	−0.0844	0.3584	0.2889

3.4.1. MQL conditions

The clearest incremental benefit from the temporal branch occurred under MQL. The hybrid achieved a mean R^2 of 0.8293, RMSE of 0.1779 µm and MAE of 0.1448 µm; relative to gradient boosting and Parameters + RMS, its mean RMSE was lower by 44.6% and 26.1%, respectively (Eq. 9). The hybrid outperformed each comparator on MQL RMSE in four of five repetitions and achieved lower MAE than Parameters + RMS in all five. Nevertheless, the advantage was not universal: in the fifth repetition, hybrid RMSE was 0.2853 µm, compared with 0.2493 µm for gradient boosting and 0.2664 µm for Parameters + RMS.

Stage 1 gives complementary support for this result at the level of individual runs (Fig. 7). Relative to Parameters + RMS, adding the CNN reduced the seed-averaged absolute error for seven of nine MQL runs, with a mean change of −0.083 µm. For the other blocks the changes were mixed and close to zero on average: 13 of 27 B1 runs (mean change +0.025 µm), two of nine L9 dry runs (+0.029 µm) and four of nine NMQL runs (+0.004 µm) improved. The temporal branch therefore helped consistently only under MQL, in both evaluation stages.

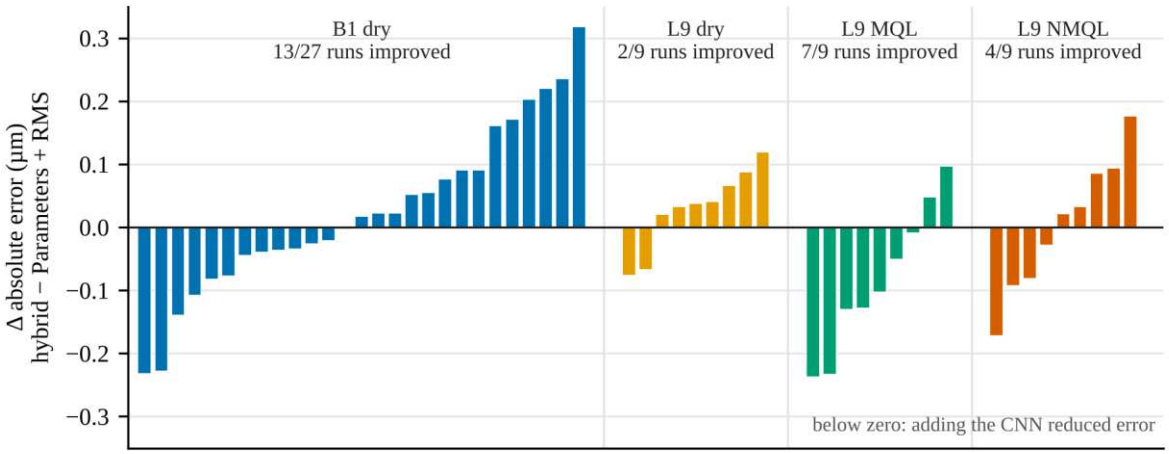


Fig. 7 Change in seed-averaged absolute error for each run when the CNN branch is added to Parameters + RMS (Stage 1, seeds 42-44). Bars below zero indicate runs for which the hybrid was more accurate; runs are sorted within each block.

3.4.2. NMQL conditions

Under NMQL, both signal-informed neural models achieved lower mean RMSE and MAE than parameter-only gradient boosting. The lowest mean errors were obtained with Parameters + RMS (RMSE 0.3354 µm, MAE 0.2741 µm); adding the CNN increased these values to 0.3584 and 0.2889 µm, respectively. The hybrid outperformed gradient boosting on both RMSE and MAE in all five repetitions, but Parameters + RMS achieved lower RMSE than the hybrid in every repetition. The NMQL results therefore supported the signal-informed neural alternatives over the gradient-boosting comparator but did not support an incremental RMSE benefit from the temporal branch. Absolute prediction performance remained limited, as reflected in the near-zero mean R^2 for Parameters + RMS and the negative mean R^2 for the hybrid.

3.4.3. Dry conditions

For dry cutting, gradient boosting achieved the lowest mean RMSE and MAE and the highest mean R^2, and its RMSE was lower than that of the hybrid in all five repetitions. The hybrid achieved a mean RMSE of 0.4284 µm, compared with 0.3694 µm for gradient boosting and 0.4135 µm for Parameters + RMS. Dry runs constituted two-thirds of the dataset and consequently contributed most of the observations to the pooled error metrics, so the poorer hybrid performance within this subset offset its favourable MQL results. The dry subset also combined the B1 and L9 dry blocks; these regime summaries should therefore not be interpreted as a matched L9-only comparison of lubrication conditions.

3.5. Summary of predictive findings

The results demonstrate the feasibility of run-level Ra prediction using temporal dynamometer signals combined with available machining parameters. However, the predictive benefit of the temporal branch depended on the subset and metric considered. The hybrid achieved the lowest overall mean MAE in repeated-partition evaluation but did not consistently improve RMSE over parameter-only gradient boosting or Parameters + RMS. Its strongest regime-specific advantage occurred under MQL, whereas RMS features without the CNN performed best under NMQL and gradient boosting performed best under dry conditions.

Figs. 5-7 explain why the pooled metrics changed little. Adding the CNN altered individual run predictions substantially, but the improvements for MQL runs were offset by deteriorations among the dry runs, which make up two-thirds of the dataset (Fig. 7). All configurations also compressed the range of Ra (Fig. 5), and their rankings were sensitive to the training seed (Fig. 6).

The regime-specific findings remain exploratory because the MQL and NMQL subsets each contained nine unique runs, reused across the five repetitions, and no statistical superiority is claimed from the reported mean differences. Furthermore, the evaluated outputs are run-level Ra estimates obtained from recorded signals; the results do not establish time-local roughness accuracy or demonstrated real-time deployment.

4. Conclusions

The present study assessed run-level surface-roughness (Ra) prediction for Ti-6Al-4V turning under dry, MQL and NMQL conditions by combining machining parameters with dynamometer signals. Four neural input configurations were compared with parameter-only tree models under strictly run-disjoint evaluation; the hybrid model showed a temporal-branch benefit only in the MQL subset. The following conclusions can be drawn:

- Run-level Ra can be predicted from machining parameters and dynamometer signals with mean errors on the order of 0.3-0.4 μm under rigorous run-disjoint evaluation.
- Parameter-only gradient boosting is already competitive on pooled metrics; adding dynamic RMS and a 1D CNN does not produce a consistent overall RMSE reduction.
- The temporal CNN branch yields its clearest benefit under MQL (≈45 % RMSE reduction relative to gradient boosting), while gradient boosting is strongest under dry conditions and parameters + RMS is strongest under NMQL.
- Learned temporal features therefore provide a regime-dependent rather than uniform advantage, underscoring the need for regime-aware model selection and larger lubrication-specific data sets.
- The two-branch framework supplies a transparent ablation template that can be scaled to other machining processes and richer sensor suites while preserving run-independent evaluation.
- Future work will extend the framework towards big-data machining analytics, using larger multi-sensor datasets for real-time surface-quality monitoring for different machining processes.

Statements and Declarations

Funding: This research received no specific grant from any funding agency in the public, commercial or not-for-profit sectors.

Competing interests: The authors have no relevant financial or non-financial interests to disclose.

Data availability: The datasets generated and analysed during the current study are available from the corresponding author on reasonable request.

Ethics approval: Not applicable. This study did not involve human participants or animals.

References

[1] E.O. Ezugwu, Z.M. Wang, Titanium alloys and their machinability—a review, J. Mater. Process. Technol. 68 (1997) 262-274. https://doi.org/10.1016/S0924-0136(96)00030-1

[2] D. Ulutan, T. Ozel, Machining induced surface integrity in titanium and nickel alloys: a review, Int. J. Mach. Tools Manuf. 51 (2011) 250-280. https://doi.org/10.1016/j.ijmachtools.2010.11.003

[3] P.G. Benardos, G.-C. Vosniakos, Predicting surface roughness in machining: a review, Int. J. Mach. Tools Manuf. 43 (2003) 833-844. https://doi.org/10.1016/S0890-6955(03)00059-2

[4] S. Ramesh, L. Karunamoorthy, K. Palanikumar, Measurement and analysis of surface roughness in turning of aerospace titanium alloy (gr5), Measurement 45 (2012) 1266-1276. https://doi.org/10.1016/j.measurement.2012.01.010

[5] G. D'Mello, P. Srinivasa Pai, N.P. Puneet, Surface roughness analysis in high speed turning of Ti-6Al-4V using coated carbide inserts: experimental and modeling studies, Tribol. Ind. 40 (2018) 457-476. https://doi.org/10.24874/ti.2018.40.03.12

[6] M.K. Gupta, Q. Song, Z. Liu, M. Sarikaya, M. Jamil, M. Mia, N. Khanna, G.M. Krolczyk, Experimental characterisation of the performance of hybrid cryo-lubrication assisted turning of Ti-6Al-4V alloy, Tribol. Int. 153 (2021) 106582. https://doi.org/10.1016/j.triboint.2020.106582

[7] H. Hegab, H.A. Kishawy, et al., On machining of Ti-6Al-4V using multi-walled carbon nanotubes-based nano-fluid under minimum quantity lubrication, Int. J. Adv. Manuf. Technol. 97 (2018) 1593-1603. https://doi.org/10.1007/s00170-018-2028-4

[8] T. Özel, Y. Karpat, Predictive modeling of surface roughness and tool wear in hard turning using regression and neural networks, Int. J. Mach. Tools Manuf. 45 (2005) 467-479. https://doi.org/10.1016/j.ijmachtools.2004.09.007

[9] I. Asiltürk, M. Çunkaş, Modeling and prediction of surface roughness in turning operations using artificial neural network and multiple regression method, Expert Syst. Appl. 38 (2011) 5826-5832. https://doi.org/10.1016/j.eswa.2010.11.041

[10] A.M. Zain, H. Haron, S. Sharif, Prediction of surface roughness in the end milling machining using artificial neural network, Expert Syst. Appl. 37 (2010) 1755-1768. https://doi.org/10.1016/j.eswa.2009.07.033

[11] M. Mia, N.R. Dhar, Prediction of surface roughness in hard turning under high pressure coolant using artificial neural network, Measurement 92 (2016) 464-474. https://doi.org/10.1016/j.measurement.2016.06.048

[12] D. Wu, C. Jennings, J. Terpenny, R.X. Gao, S. Kumara, A comparative study on machine learning algorithms for smart manufacturing: tool wear prediction using random forests, J. Manuf. Sci. Eng. 139 (2017) 071018. https://doi.org/10.1115/1.4036350

[13] A. Košarac, S. Tabaković, C. Mlađenović, M. Zeljković, G. Orašanin, Next-gen manufacturing: machine learning for surface roughness prediction in Ti-6Al-4V biocompatible alloy machining, J. Manuf. Mater. Process. 7 (2023) 202. https://doi.org/10.3390/jmmp7060202

[14] R. Teti, K. Jemielniak, G. O'Donnell, D. Dornfeld, Advanced monitoring of machining operations, CIRP Ann. 59 (2010) 717-739. https://doi.org/10.1016/j.cirp.2010.05.010

[15] K.A. Risbood, U.S. Dixit, A.D. Sahasrabudhe, Prediction of surface roughness and dimensional deviation by measuring cutting forces and vibrations in turning process, J. Mater. Process. Technol. 132 (2003) 203-214. https://doi.org/10.1016/S0924-0136(02)00920-2

[16] E. García Plaza, P.J. Núñez López, Analysis of cutting force signals by wavelet packet transform for surface roughness monitoring in CNC turning, Mech. Syst. Signal Process. 98 (2018) 634-651. https://doi.org/10.1016/j.ymssp.2017.05.006

[17] O.B. Abouelatta, J. Mádl, Surface roughness prediction based on cutting parameters and tool vibrations in turning operations, J. Mater. Process. Technol. 118 (2001) 269-277. https://doi.org/10.1016/S0924-0136(01)00959-1

[18] D.R. Salgado, F.J. Alonso, I. Cambero, A. Marcelo, In-process surface roughness prediction system using cutting vibrations in turning, Int. J. Adv. Manuf. Technol. 43 (2009) 40-51. https://doi.org/10.1007/s00170-008-1698-8

[19] V. Upadhyay, P.K. Jain, N.K. Mehta, In-process prediction of surface roughness in turning of Ti-6Al-4V alloy using cutting parameters and vibration signals, Measurement 46 (2013) 154-160. https://doi.org/10.1016/j.measurement.2012.06.002

[20] W. Guo, C. Wu, Z. Ding, Q. Zhou, Prediction of surface roughness based on a hybrid feature selection method and long short-term memory network in grinding, Int. J. Adv. Manuf. Technol. 112 (2021) 2853-2871. https://doi.org/10.1007/s00170-020-06523-z

[21] P.-M. Huang, C.-H. Lee, Estimation of tool wear and surface roughness development using deep learning and sensors fusion, Sensors 21 (2021) 5338. https://doi.org/10.3390/s21165338

[22] F. Hojati, B. Azarhoushang, A. Daneshi, R.H. Khiabani, Prediction of machining condition using time series imaging and deep learning in slot milling of titanium alloy, J. Manuf. Mater. Process. 6 (2022) 145. https://doi.org/10.3390/jmmp6060145

[23] E. García Plaza, P.J. Núñez López, Surface roughness monitoring by singular spectrum analysis of vibration signals, Mech. Syst. Signal Process. 84 (2017) 516-530. https://doi.org/10.1016/j.ymssp.2016.06.039

[24] M. Guo, J. Zhou, X. Li, Z. Lin, W. Guo, Prediction of surface roughness based on fused features and ISSA-DBN in milling of die steel P20, Sci. Rep. 13 (2023). https://doi.org/10.1038/s41598-023-42968-4

[25] S. Shang, C. Wang, X. Liang, C.F. Cheung, P. Zheng, Surface roughness prediction in ultra-precision milling: an extreme learning machine method with data fusion, Micromachines 14 (2023) 2016. https://doi.org/10.3390/mi14112016

[26] Y.-C. Lin, K.-D. Wu, W.-C. Shih, P.-K. Hsu, J.-P. Hung, Prediction of surface roughness based on cutting parameters and machining vibration in end milling using regression method and artificial neural network, Appl. Sci. 10 (2020) 3941. https://doi.org/10.3390/app10113941

[27] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, D.J. Inman, 1D convolutional neural networks and applications: a survey, Mech. Syst. Signal Process. 151 (2021) 107398. https://doi.org/10.1016/j.ymssp.2020.107398

[28] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, R.X. Gao, Deep learning and its applications to machine health monitoring, Mech. Syst. Signal Process. 115 (2019) 213-237. https://doi.org/10.1016/j.ymssp.2018.05.050

[29] W.-J. Lin, S.-H. Lo, H.-T. Young, C.-L. Hung, Evaluation of deep learning neural networks for surface roughness prediction using vibration signal analysis, Appl. Sci. 9 (2019) 1462. https://doi.org/10.3390/app9071462

[30] H.-Y. Chen, C.-H. Lee, Deep learning approach for vibration signals applications, Sensors 21 (2021) 3929. https://doi.org/10.3390/s21113929

[31] E. Stathatos, E. Tzimas, P. Benardos, G.-C. Vosniakos, Convolutional neural networks for raw signal classification in CNC turning process monitoring, Sensors 24 (2024) 1390. https://doi.org/10.3390/s24051390

[32] X. Zhang, S. Wang, F. Gao, H. Wang, H. Wu, Y. Liu, An intelligent surface roughness prediction method based on automatic feature extraction and adaptive data fusion, Auton. Intell. Syst. 4 (2024) 27. https://doi.org/10.1007/s43684-024-00083-9

[33] A.P. Rifai, H. Aoyama, N.H. Tho, S.Z. Md Dawal, N.A. Masruroh, Evaluation of turned and milled surfaces roughness using convolutional neural network, Measurement 161 (2020) 107860. https://doi.org/10.1016/j.measurement.2020.107860

[34] V. Upadhyay, P.K. Jain, N.K. Mehta, Prediction of surface roughness using cutting parameters and vibration signals in minimum quantity coolant assisted turning of Ti-6Al-4V alloy, Int. J. Manuf. Technol. Manag. 27 (2013) 33-46. https://doi.org/10.1504/IJMTM.2013.058636

[35] S. Kapoor, A. Narayanan, Leakage and the reproducibility crisis in machine-learning-based science, Patterns 4 (2023) 100804. https://doi.org/10.1016/j.patter.2023.100804

[36] D.R. Roberts, V. Bahn, S. Ciuti, M.S. Boyce, J. Elith, G. Guillera-Arroita, et al., Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure, Ecography 40 (2017) 913-929. https://doi.org/10.1111/ecog.0288.