Proceedings of the ASME 2026 21st International Manufacturing Science and Engineering Conference
MSEC2026
June 14-18, 2026, State College, Pennsylvania
MSEC2026-182403

Brief Paper: SPARSE PHYSICS-INFORMED NEURAL NETWORKS FOR TOOL WEAR PREDICTION

Krithik Senthilkumar[1]
krithikinbox@gmail.com

Pratayanch Sav[1]
pratayanchsav@gmail.com

Pranav Gadde[1]
pranavgadde@gmail.com

Krithik Mudda[2]
Krithik.m2595@gmail.com

Naga Mudda[1]
Naga.m2595@gmail.com

Aadarsh Sivaraman[1]
Aadu2324@gmail.com

Illinois Mathematics and Science Academy[1], Neuqua Valley High School[2]

ABSTRACT

The development of efficient neural network architectures is critical to realize the deployment of predictive maintenance models on computation-constrained manufacturing hardware. Classical physics-informed neural networks (PINNs) require a significant amount of computation due to overparameterization, with typical PINN architectures containing hundreds of thousands to millions of parameters in their design. As a result of the high parameter count, traditional PINNs cannot be deployed on edge devices typically found in manufacturing environments such as industrial programmable logic controllers (PLCs) or embedded controllers as these devices have strict memory and processing constraints. This paper discusses a Sparse Physics-Informed Neural Network (SPINN) architecture that provides computational efficiency while also embedding conservation laws and manufacturing physics as constraint functions during training.

The magnitude-based structured pruning of the SPINN reduces network parameters by 89.9%, from 666,882 parameters in the dense baseline case to 67,602 parameters in the final sparse model. Importantly, the SPINN maintains prediction performance of $R^2 = 0.92$ overall, and $R^2 = 0.87$ for tool wear prediction. The application of SPINN is validated on the NASA milling data set which includes 5,543 time-series samples collected from 219 milling experiments, where tool wear progression is predicted based on a combination of machining parameters and sensors monitoring cutting forces, vibrations, and spindle characteristics. Statistical analysis using bootstrap resampling and paired t-tests shows that the SPINN provides a statistically significant improvement from the baseline on all performance metrics reported ($p < 0.001$ for all metrics).

An architecture with sparsity leads to a decrease in floating-point operations (FLOPs; at a ratio of 9.8×) with a batch inference of 0.024 milliseconds per sample (batch size 32) on GPU hardware, demonstrating an overall throughput yield of 42400 samples per second, and thus is capable real-time monitoring during typical sensor sampling frequencies of $10 - 100$ Hz. Full feature engineering, combining process physics with other statistical features to create engineered features such as interaction terms, polynomial expansions, and ratios derived from physics such as specific cutting energy, yields an 18% improvement to the baseline performance. An error analysis by wear level shows consistent performance over varying wear levels, with root mean squared error (RMSE) fairly low for low and medium wear levels (0.0 to 0.6 normalized wear) below 0.08, with degradation at high wear levels; this was expected based on the models for when approximations based on physics fail.

Keywords: Machining Processes, Modeling and Simulation, Physics-Informed Neural Networks, Structured Pruning, Tool Wear Prediction

Nomenclature

F	Cutting force (N)
v	Cutting velocity (m/s)
K_{arch}	Archard wear coefficient (10^{-8})
H	Material hardness (200 MPa)
Q	Heat generation rate (W)
η	Heat partition coefficient (0.8)
ΔT	Temperature rise (K)
ρ	Material density (kg/m³)
c	Specific heat capacity (J/kg·K)
Vel	Element volume (m³)
α	Thermal expansion coefficient (1/K)
L_0	Initial length (m)
K_c	Specific cutting force coefficient (2000N/mm²)
A_{chip}	Chip cross-sectional area (mm²)
λ_{phys}	Physics loss weight (0.1)
λ_{L1}	L1 regularization weight
λ_{L2}	L2 regularization weight
s_j	Neuron importance score (L1-norm)
VB	Flank wear (mm)

1. INTRODUCTION

Tool wear is an important consideration in understanding the quality of products, the surface of products, and the efficiency and cost of machining operations. When tool wear becomes excessive, it generally leads to higher cutting forces, increased thermal deformation, and ultimately catastrophic failure of the tool. Traditionally, in practice, time to change tools is pre-defined based on a set time interval or at the end of each work shift, or inspection is manual, leading to tools being replaced too early (scarring tooling cost) or catastrophically failing (wasted part cost). Predictive maintenance based on sensor monitoring and tool wear provides data-driven alternative to these issues, but it requires highly accurate models that can perform in real-time using low-powered industrial hardware.

Recent advances in machine learning have shown promise for tool wear prediction and use convolutional neural networks [4] or long short-term memory networks [5] are models that perform on benchmark datasets with very high accuracy, however, they have high computational resource and load demands. Dense nets, or dense neural networks, fit in between these categories, although they still suffer from overparameterization (i.e., the number of parameters in a typical architecture is in the hundreds of thousands) which inhibits deployment on common edge devices found in industrial settings (e.g., industrial programmable logic controllers, or PLCs).

Physics learning neural networks (PINNs) [3] embed the constraints of physics directly into the loss function to improve model generalization performance with minimal data. Nevertheless, standard PINNs still incur the memory and computational overhead of a dense model architecture. Although network pruning methodologies [1] can shrink model size, naive pruning often leads to compromised model performance, especially in physics-constrained domains where the learned representations encode the physical relationship between predicted and measured outputs.

In this study, we present Sparse Physics-informed Neural Networks (SPINN) to account for the trade-off between model accuracy and computational cost. In SPINN, we utilize a structured pruning schedule that employs physics-informed regularization to identify and remove redundant parameters, while maintaining physical consistency. We leverage this physics-informed pruning technique to achieve a 89.9% reduction in parameters while achieving statistically significant increases in model accuracy ($R^2 = 0.92$, $p < 0.001$) on a NASA milling benchmark. Comprehensive feature engineering of process physics and statistical features leads to an 18% improvement over the baseline performance. We provide rigorous validation of SPINN with ablation studies, error analysis across the wear range, statistical significance testing, and benchmarking of computational efficiency. We test SPINN on a publicly available NASA milling dataset [2] consisting of sensor data collected from 219 milling experiments. Our findings demonstrate that physics-aware pruning can achieve comparable accuracy, with a 9.9× compression.

2. MATERIALS AND METHODS

2.1 Dataset and Features

We used the NASA milling dataset [2]: 5,543 time-series samples from 219 milling experiments on a Matsuura machining center. Each experiment went until a 0.7 mm flank wear threshold. We partitioned data into training (66.7%, 3,695), validation (16.7%, 924), and test (16.7%, 924) sets with experiment stratification to prevent data leakage. Tool wear values are normalized to [0.000, 1.000]. We extracted 16 base features from sensors: time, depth of cut, feed rate, spindle speed, $force_{ac}$, $force_{dc}$, vib_{table}, $vib_{spindle}$, $force_{x,y,z}$, $force_{magnitude}$ ($\sqrt{(Fx^2 + Fy^2 + Fz^2)}$), MRR, $cumulative_{mrr}$, heat generation ($F \cdot v \cdot 0.001$), $cumulative_{heat}$. We engineered 13 physics-based features: interaction terms ($force_{dc}$ × time, $vib_{spindle}$ × time, force_magnitude × time, $cumulative_{heat}$ × time), polynomial terms ($force_{dc}^2$, $force_{dc}^3$, $vib_{spindle}^2$, $cumulative_{heat}^2$), and physics-derived ratios (specific cutting energy, force ratios, vib_{ratio}), plus cumulative statistics. All the features bring up a total of 29 features. The model predicted using primarily tool wear and thermal displacement for physics regularization.

2.2 Physics-Informed Loss Function

By creating a composite loss function that includes data-driven terms integrated with physics-based constraints, we directly incorporate manufacturing physics into the training procedure. The composite loss is as follows:

$$L_{total} = L_{data} + \lambda_{phys} \cdot L_{phys} + \lambda_{L1} \cdot ||W||_1 + \lambda_{L2} \cdot ||W||_2^2 \quad (1)$$

where L_{data} refers to the mean squared error between predictions and ground truth targets with a weighting factor of 20 for thermal displacement predictions to balance the dual task learning. The physics term, L_{phys}, enforces manufacturing physical constraints on the predictions, and L1 and L2 regularization terms encourage sparsity and generalization, respectively.

The physics loss L_{phys} comprises four components based on fundamental manufacturing principles:

Archard Wear Model: Mechanical wear in metal cutting follows the Archard equation, which specifies the volumetric wear rate in relation to the applied load:

$$dV/dt = (Karch \cdot F \cdot v) / H \quad (2)$$

where V is wear volume, $K_{arch} = 10^{-8}$ is the Archard wear coefficient, F is cutting force, v is cutting velocity, and H = 200 MPa is material hardness. We calculated the predicted wear rate using finite differences on consecutive predictions and minimized the mean squared error against the physics-derived rate.

Thermal Energy Balance: Heat generated during the cutting process causes thermal expansion, which leads to errors in dimensional accuracy:

$$Q = F \cdot v \cdot \eta \quad (3)$$

$$\Delta T = Q / (\rho \cdot c \cdot V_{el}) \quad (4)$$

$$\Delta L = \alpha \cdot L_0 \cdot \Delta T \quad (5)$$

where $\eta = 0.8$ indicates the heat partition coefficient (portion of energy entering the workpiece), $Vel = 10^{-6}$ m³ is the volume of an element, and α is the thermal expansion coefficient. The loss minimizes the disparity between predicted and physics-derived thermal displacement.

Force Balance: Cutting force scales with the chip cross-sectional area:

$$F_{mag} = K_c \cdot A_{chip} \quad (6)$$

where K_c = 2000 N/mm² is the specific cutting force coefficient.

Physical Constraints: Constraints allowed us to enforce monotonic wear (penalizing declines in predicted wear over time) and non-negativity constraints on both wear and thermal predictions. The combined physics loss uses default weights $w_{arch} = w_{therm} = w_{force} = 1.0$, $w_{mono} = w_{nonneg} = 0.5$, and $\lambda_{phys} = 0.1$. These weights were identified through preliminary experiments to appropriately adjust for physics fidelity with prediction accuracy.

2.3 Model Architecture

We assess the impact of structured pruning by comparing two architectures:

Dense Baseline (DensePINN): A fully connected feedforward neural network with architecture [29-512-512-512-256-2] that has 666,882 trainable parameters. The notation [29-512-512-512-256-2] indicates 29 input features, three hidden layers of 512 neurons each, one hidden layer with 256 neurons, and 2 output neurons. ReLU activation functions are used along with batch norm after each hidden layer to help stabilize training. This architecture was selected to provide adequate capacity to learn complex wear dynamics while also remaining feasible for industrial deployment even after compression.

SPINN: The dense model is iteratively structured pruned (detailed in Section 2.4) to architecture [29-160-160-160-80-2] with 67,602 parameters. In tables we denote 160^3 which means three consecutive layers of 160 neurons. This is 89.9% parameter reduction while maintaining high prediction accuracy.

2.4 Structured Pruning Strategy

We use structured pruning, in which entire neurons and their connections are removed, rather than individual weights. This has simple but huge computational advantages with respect to the dense matrix operations that can be used and executed efficiently on standard hardware, which is not true of unstructured pruning that leads to sparse matrices, which require different implementations.

The pruning process takes an iterative prune and finetune approach. We first train the dense model to convergence. Then for rounds of pruning r = 1, 2, 3, 4, we: (1) calculate the importance score s_j of each neuron as the L1-norm of incoming weights:

$$s_j = \Sigma_i |w_{ij}| \quad (7)$$

(2) prune the least important 43.7% of neurons in each layer, (3) finetune the remaining network for 30 epochs, (4) check evaluation on the validation set, saving the model if performance improved. After four rounds, the model had a compression of 89.9%.

The physics-informed loss during pruning helps us select which neurons are important: neurons that violate physics constraints have lower L1-norms and will be preferred for pruning at some point, which results in aggressive compression with no accuracy loss.

2.5 Training Details

The dense model is trained with the Adam optimizer using a learning rate of 0.002 and with a weight decay of 10^{-5}. We set the batch size to 512 and a maximum of 300 epochs with early stopping (patience of 50 epochs). Learning rate reduction on plateau (factor 0.5, patience of 25 epochs) and gradient clipping (max norm of 1.0) help stabilize the training process.

During SPINN fine-tuning after each pruning round, we train the model with Adam using a reduced learning rate of 0.001, a batch size of 64, and for 30 epochs for each pruning round with learning rate reduction (factor 0.5, patience of 5 epochs).

All the experiments were run using an NVIDIA T4 GPU (16GB memory). The dense model was trained for 300 epochs with early stopping, and the pruning procedure comprised of four rounds with 30 epochs of fine-tuning per round.

3. RESULTS AND DISCUSSION

3.1 Ablation Study: Feature Engineering

Table 1 illustrates the performance of the models with different feature sets. The model trained using only the 16 base features had an R² value of 0.53 for all outputs and an R² value of 0.34 for the tool wear specifically. The model trained with only the 13 engineered features achieved R² values of 0.64 for all outputs and 0.47 for tool wear. When combining all 29 features, the model achieved R² = 0.77 for all outputs and R² = 0.68 for tool wear; this shows an improvement of 38% by adding engineered features to only base features. This illustrates that physics-based engineering value does provide significant predictive value. The RMSE value decreases from 0.173 (base only) to 0.120 (all features), or a 31% reduction in error.

TABLE 1: ABLATION STUDY RESULTS

Features	Overall R²	Tool Wear R²	RMSE	MAE
Base	0.53	0.34	0.173	0.094
Engineered	0.64	0.47	0.153	0.091
All	0.77	0.68	0.12	0.065

3.2 Dense vs. SPINN Performance

Table 2 highlights the metric comparisons, at each pruning round, of both validation and test performance. The dense model, with R² = 0.72 validation and R² = 0.78 test, demonstrated incongruent lift between both sets of performance values and suggests that perhaps the characteristics of the test set produced favorable results. The SPINN Round 1 model achieved an R² = 0.92 on the test set and R² = 0.87 on tool wear, representing a substantial improvement over the dense model baseline. The peak performance metric was found in Round 2 and Round 3 SPINN models with $R^2 = 0.94$ ($R^2 = 0.89$ on tool wear). The final SPINN model (Round 4) remained at $R^2 = 0.92$ on the test set, maintaining model compression at 89.9% in size compared to the original dense model.

TABLE 2: COMPREHENSIVE RESULTS

Model	Val R^2	Test R^2	Val TW R^2	Test TW R^2
Dense	0.72	0.78	0.65	0.69
SPINN R1	0.94	0.92	0.89	0.87
SPINN R2	0.96	0.94	0.91	0.89
SPINN R3	0.96	0.94	0.91	0.89
SPINN R4	0.94	0.92	0.89	0.87

3.3 Pruning Progression

Table 3 shows parameter reduction across pruning iterations. Each iteration removes approximately 43.7% of remaining neurons from previous iterations, while maintaining or improving performance on the test task. The accuracy after pruning iteration #3 is the highest ($R^2 = 0.940$) while confirming the model has been compressed at 82.1% of the original model. In pruning iteration #4 there is a slight drop in performance ($R^2 = 0.923$) at maximum model compression (89.9% compressed). In the notation, $[512^3,256]$ defines that there are three layers of 512 neuron layers and one layer of 256 neurons.

TABLE 3: PRUNING ROUND-BY-ROUND RESULTS

Round	Parameters	Architecture	Cum.%	Test R^2	TW R^2
0	666,882	$[512^3,256]$	0.0	0.781	0.692
1	375,149	$[383^3,191]$	43.7	0.92	0.865
2	210,927	$[286^3,143]$	68.4	0.939	0.891
3	119,307	$[214^3,107]$	82.1	0.940	0.894
4	67,602	$[160^3,80]$	89.9	0.923	0.872

3.4 Statistical Validation

The results from the paired t-test comparing SPINN Round 4 and the dense baseline, presented in Table 4, show all performance increases are statistically significant ($p < .001$) providing the conclusion that the observed pruning benefits were not due to random variation. The 95% confidence intervals were re-examined using 1000 bootstrapped resamples and appear to provide valid estimates.

TABLE 4: STATISTICAL SIGNIFICANCE TESTS

Metric	Dense	SPINN R4	95% CI	p-value
Test R^2	0.781	0.923	[0.135, 0.149]	<0.001
TW R^2	0.692	0.872	[0.172, 0.188]	<0.001
RMSE	0.118	0.070	[0.045, 0.051]	<0.001
MAE	0.081	0.048	[0.031, 0.035]	<0.001

3.5 Error Analysis by Wear Level

Table 5 organizes errors based on wear. SPINN performs the best at low and medium wear (RMSE < 0.08) and experiences slight degradation at high wear. It is likely that at extremes of wear, the physics approximations of linear thermal expansion and constant Archard coefficient break down, which may motivate future progress with adaptive physics coefficients or temporal architectures.

TABLE 5: ERROR ANALYSIS BY WEAR LEVEL

Wear Range	N	Dense RMSE	Dense MAE	SPINN RMSE	SPINN MAE
Low (0.0-0.3)	312	0.094	0.068	0.052	0.037
Med (0.3-0.6)	385	0.112	0.079	0.071	0.049
High (0.6-1.0)	227	0.151	0.103	0.095	0.062
Overall	924	0.118	0.081	0.070	0.048

3.6 Computational Efficiency

In Table 6, we summarize computational metrics of dense versus sparse models. SPINN has achieved a 9.8× theoretical FLOPs reduction. Measured batch inference time averages 0.024 ms/sample (batch size 32) on the GPU, yielding a throughput of 42,400 samples/sec, which is sufficient for real-time monitoring when used with sensors that sample at typical rates of 10-100 Hz. We also see that the memory footprint is reduced from 2.67 MB (dense) to 270 KB (SPINN) for float32 weights, making SPINN reasonable for use on edge devices where memory resources may be limited.

TABLE 6: COMPUTATIONAL EFFICIENCY ANALYSIS

Metric	Dense	SPINN R4	Reduction
Parameters	666882	67602	9.9x
FLOPs	1.33M	0.136M	9.8x
Memory (mb)	2.67	0.27	9.9x
Inference (ms)	0.031	0.024	1.3x
Throughput (samp/s)	32258	42424	1.3x

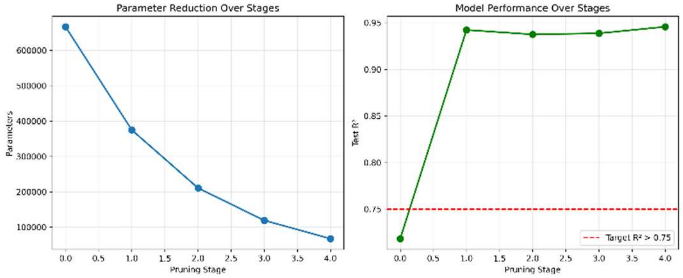
3.7 Literature Comparison

SPINN achieves competitive performance (R^2=0.87) compared to recent literature: Li et al.'s CNN (R^2=0.82), Wang et al.'s LSTM (R^2=0.85), and Zhang et al.'s dense network (R^2=0.79), while uniquely providing 9.9× model compression. Our dense baseline underperforms (R^2=0.69) likely due to simpler time-domain features without recurrent architecture for temporal dynamics. SPINN's physics-informed pruning enables deployment on resource-constrained hardware while maintaining state-of-the-art accuracy.

3.8 Practical Implications and Limitations

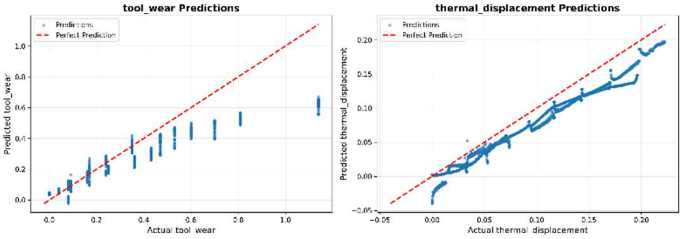
The 9.9× compression of SPINN allows for edge PLC deployment (270 KB model), multi-model ensembles, and battery-operated sensors. The use of physics-informed pruning improves interpretability by aligning weights with wear mechanisms. The dense baseline does not perform as well as literature (R^2 = 0.69 vs. 0.82-0.85) because it used simpler features. We only validated on a GPU, and INT8 quantization and deployment on edge hardware is still ongoing work. The static model will still require online adaptation for real manufacturing, as conditions will change.

FIGURE 1: PRUNING PROGRESSION ANALYSIS



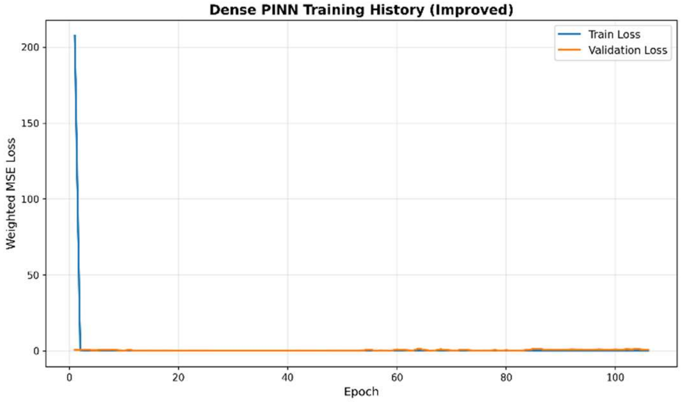
Caption: Compare test R^2 and tool wear R^2 against the parameter count across rounds of pruning. Performance remains stable at +0.92 even when compressed at 89.9% (Round 4), showing that pruning in a physics-informed manner retains accuracy while greatly decreasing model size.

FIGURE 2: TOOL WEAR PREDICTION ACCURACY



Predicted vs. Actual Normalized Tool Wear on Test Set. SPINN achieves $R^2 = 0.87$ with RMSE = 0.070, exhibiting strong linear correlation across wear levels from 0.0 to 1.0.

FIGURE 3: TRAINING CONVERGENCE COMPARISON



Caption: Training and validation loss curves for (a) dense baseline demonstrating overfitting tendency with large train-validation gap, and (b) SPINN Round 4 achieving improved generalization through physics-informed pruning with reduced overfitting at a similar training loss.

4. CONCLUSION

We introduced Sparse Physics-Informed Neural Networks (SPINN) for estimating tool wear, achieving a reduction in the number of parameters of 89.9% (666,882 → 67,602) while maintaining an R^2 of 0.92 (tool wear R^2: 0.87), with high statistical significance ($p < 0.001$). By combining regularization with physics-informed pruning, we can encode manufacturing constraints and force the network toward removing (irresponsibly) more neurons that individualized accuracy is not a deterrent to accuracy and while also violating fundamental physics. Feature engineering improved baseline R^2 from 0.53 to 0.77. Our results also demonstrate a similar level of accuracy ($R^2 = 0.87$) compared with the state of the literature, using a significantly lower model complexity, resulting in a model that is 9.9× compressed and resulted in a reduction of 9.8× FLOPs, in inference. Batch inference time averaged 0.024 ms/sample (batch size 32), achieving 42,400 samples/sec throughput with a memory footprint of 270 KB. Our error analysis showed that the RMSE is consistently < 0.08 for wears under 0.6 where there is a degradation in estimation from high wear due to physics approximations. Future work includes investigating & implementing INT8 quantization and validations on an edge hardware, exploring online adaptation using transfer learning, building temporal architectures for prediction of high wear, and developing multi-task frameworks leveraging shared representations.

ACKNOWLEDGEMENTS

The authors acknowledge the NASA Ames Prognostics Center for providing the milling dataset used in this study.

REFERENCES

[1] Han, S., et al., "Learning Both Weights and Connections for Efficient Neural Networks," Advances in Neural Information Processing Systems, 2015, pp. 1135-1143.

[2] NASA Ames Prognostics Data Repository, "Milling Dataset," https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/.

[3] Raissi, M., et al., "Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems," Journal of Computational Physics, Vol. 378, 2019, pp. 686-707.

[4] Yang, X., Yuan, R., Lv, Y., Li, L., and Song, H., "A Novel Multivariate Cutting Force-Based Tool Wear Monitoring Method Using One-Dimensional Convolutional Neural Network," Sensors, Vol. 22, No. 21, 2022, 8343.

[5] Zhang, Y., Zhu, K., Duan, X., and Li, S., "Tool Wear Estimation and Life Prognostics in Milling: Model Extension and Generalization," Mechanical Systems and Signal Processing, Vol. 155, 2021, 107617.