Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.081719

ARTICLE

Real-Time Optimization of Vertical Roller Mills Using XGBoost Prediction and Q-Learning Control

Anping Wan[1,2,3], **Yingchang Gao**[1,3], **Weikang Liu**[1], **Rui Yin**[1] **and Khalil Al-Bukhaiti**[1,3,*]

[1]Laboratory for Microwave Spatial Intelligence and Cloud Platform, Hangzhou City University, Hangzhou, China
[2]Zhengzhou Digital Industry Institute, Zhengzhou, China
[3]Zhejiang Key Laboratory of Advanced Equipment Manufacturing and Measurement Technology, Zhejiang University, Hangzhou, China

*Corresponding Author: Khalil Al-Bukhaiti. Email: eng.khalil670@hotmail.com

Received: 07 March 2026; Accepted: 29 April 2026; Published: 15 June 2026

ABSTRACT: Vertical roller mills are essential for energy-intensive grinding in cement, minerals, and metallurgy industries, consuming up to 50% of plant electricity and frequently experiencing operational instabilities (including excessive vibration and main motor current fluctuations) that drive unplanned downtime, increased wear, and reduced throughput. Despite their importance, real-time autonomous optimization remains challenging due to the nonlinear interactions among grinding pressure, feed rate, separator speed, and aerodynamic factors, which limit traditional control strategies under varying loads. This paper presents a real-time operational optimization system for large-scale vertical roller mills using big industrial data and artificial intelligence (AI). From a 5400 kW Loesche LM56.4 mill, 2,764,800 samples were collected at 1 Hz over 32 days of continuous production. A systematic pipeline was developed: quartile-based outlier-robust cleaning; domain-informed feature engineering including Total Current; Random Forest (RF) permutation importance selection of the top 15 parameters; and Extreme Gradient Boosting (XGBoost) regression models with hyperparameters tuned by Tree-structured Parzen Estimator (TPE) Bayesian optimization. The resulting models achieved strong predictive performance, Mean Absolute Percentage Error (MAPE) of 1.3% (95% CI: 1.1%–1.5%) for main motor current (R^2 = 0.9997) and 5.8% (95% CI: 5.3%–6.3%) for shell vibration (R^2 = 0.9717), representing reductions of 89% and 59%, respectively, relative to the Long Short-Term Memory (LSTM) baseline. These surrogates were embedded into a tabular Q-learning Reinforcement Learning (RL) agent that autonomously adjusts feed rate, grinding pressure, separator speed, and exhaust damper position via a discrete action space and multi-objective reward function, communicating with the Distributed Control System (DCS) via Open Platform Communications Unified Architecture (OPC-UA). Closed-loop evaluation yielded simultaneous reductions of 6.0% in peak current (181.92 → 170.04 A) and 9.4% in peak vibration (5.51 → 4.99 mm/s) while maintaining throughput. A PyQt5-based graphical interface enabling real-time monitoring, predictive alerts, and automatic DCS write-back was deployed and operated stably for two weeks.

KEYWORDS: Vertical roller mill; operational optimization; XGBoost; Bayesian hyperparameter optimization; Q-learning; energy efficiency; vibration reduction; real-time control; Industry 4.0

1 Introduction

Vertical grinding mills are pivotal in industries such as cement, minerals, and metallurgy, where they facilitate efficient material pulverization under high-pressure conditions. These mills integrate complex mechanical, hydraulic, and electrical systems to process raw materials into fine powders, often consuming

up to 50% of total plant electricity in cement production [1]. In China, where vertical mills dominate large-scale grinding operations, specific energy consumption typically ranges from 14–20 kWh/t for raw-meal grinding [2,3] and vibration-induced failures account for 20% of unplanned halts [3].

Current optimization strategies bifurcate into structural enhancements and operational parameter tuning. Structural modifications have yielded 10%–15% efficiency gains but require costly hardware overhauls [4,5]. Conversely, operational adjustments focus on empirical tuning of interdependent parameters [6,7]. Advanced predictive approaches, including energy consumption models and genetic algorithm-optimized Backpropagation (BP) neural networks [8], have improved control precision by 15%–20% [9]; yet persistent challenges include prediction inaccuracies (errors >10%) and limited generalization across equipment states [10].

Recent post-2022 hybrid Machine Learning (ML)–Reinforcement Learning (RL) studies have begun to close this gap. Dogru et al. [11] highlights that surrogate-assisted RL is emerging as the most viable pathway for safety-critical processes yet identify no validated deployment in mineral grinding. Pural et al. [12] demonstrate that tree-based hybrid models outperform neural alternatives for tabular industrial sensing yet stop short of integrating them within a closed-loop RL framework. Luan et al. [13] established XGBoost as the most accurate predictor for Semi-Autogenous Grinding (SAG) mill liner wear (MAPE 5.27%) but address only offline prediction. Across these works, no prior study unifies a tree-based surrogate, a physically constrained multi-objective RL policy, and a production-deployed DCS interface within a single validated system for large-scale vertical roller mill operation.

This study addresses these gaps through three methodological contributions. First, XGBoost is embedded as a real-time surrogate environment for the Q-learning agent (<62 ms per cycle), enabling safe exploration without risking mechanical damage to live equipment [14,15]. Second, domain-informed feature engineering and permutation-importance-guided selection ensure the surrogate faithfully captures the nonlinear coupling specific to vertical roller mill dynamics [14]. Third, the Q-learning reward function (Eq. (1)) simultaneously penalizes energy consumption and structural vibration within a physically constrained discrete action space bounded by DCS safety limits, validated by simultaneous 6.0% current and 9.4% vibration reductions without throughput loss. The system outperforms LSTM and Support Vector Regression (SVR) baselines by 82%–89% in prediction error and delivers 2.7–3.3× greater optimization gains [16].

Contributions: (i) Ablation experiments confirm that permutation-importance-guided feature selection and Total Current engineering are necessary conditions for surrogate fidelity [3]; (ii) empirical evidence shows that the Q-learning agent converges within 1100–1300 episodes, contributing quantitative convergence characterization for surrogate-assisted RL in safety-constrained spaces [17,18]; (iii) simultaneous optimization of two competing objectives is achievable within a single normalized reward signal; (iv) TPE-optimized XGBoost achieves MAPE 1.3% (current) and 5.8% (vibration) on full-scale 1 Hz DCS data; (v) a PyQt5 interface with OPC-UA DCS write-back operated continuously for two weeks at 68 ms latency; and (vi) the complete system runs on commodity hardware, establishing a practical deployment blueprint for legacy cement plants.

2 System Overview

The proposed data-driven optimization system integrates real-time data acquisition, predictive modeling, and reinforcement learning-based decision-making to continuously improve energy efficiency and operational stability of vertical grinding mills. As illustrated in Fig. 1, the architecture comprises three tightly coupled modules: (1) Data Preprocessing and Feature Selection, (2) XGBoost Predictive Engine, and (3) Q-Learning Optimization Agent.

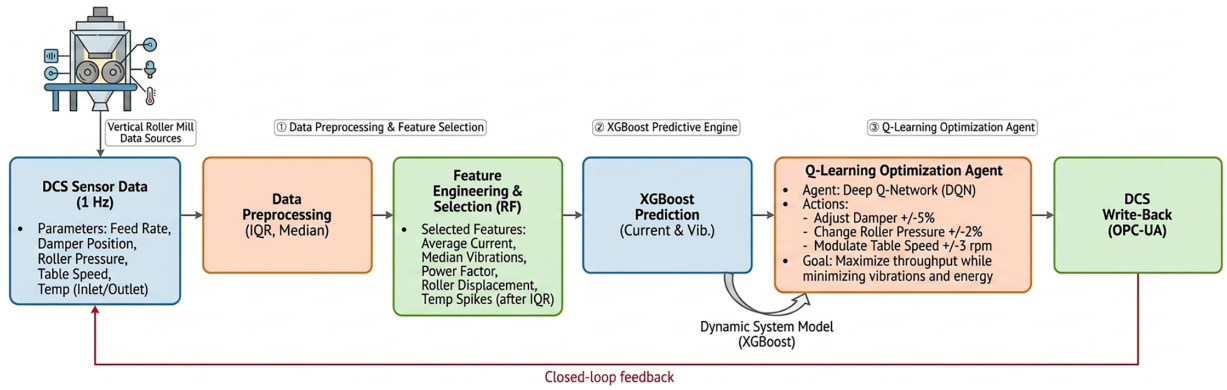

Figure 1: System workflow of the proposed vertical mill optimization framework.

Raw operational data are collected at 1 Hz from the DCS of a Loesche LM56.4 mill (450 t/h capacity). The dataset originally contained 60 parameters; fixed setpoints are discarded, leaving 43 dynamic feedback parameters. Missing values are imputed with column-wise medians, outliers are replaced using the Interquartile Range (IQR) method (values beyond Q1 − 1.5 × IQR and Q3 + 1.5 × IQR capped at the nearest quartile boundary [19]), and a composite feature Total Current is engineered as the sum of main motor, separator, and exhaust fan currents (Eq. (5)). Feature importance is evaluated using an RF regressor with 500 trees; the 15 most influential variables are selected for both prediction tasks. The Q-Learning agent treats the mill as a Markov decision process. After each action a, the new parameter set is fed to the XGBoost ensemble, returning predicted current $\hat{I}$ and vibration $\hat{V}$. The reward function is:

$$r = w_1 \cdot \left(I_0 - \hat{I}\right)/I_0 + w_2 \cdot \left(V_0 - \hat{V}\right)/V_0 - penalty \tag{1}$$

where I_0 and V_0 are baseline values, $w_1 = w_2 = 0.5$, and a small penalty ($-0.01|a|$) discourages excessive exploration. The agent updates its Q-table ($|S| = 10^4$ states × 24 actions) using an ε-greedy policy (ε decaying from 0.9 to 0.01) over 5000 episodes, converging within 1200 episodes.

3 Materials and Methods

This section details the complete methodology employed to develop and validate the proposed optimization system. Data acquisition, cleaning, and feature engineering are first described, followed by feature selection using Random Forest, XGBoost model construction, Q-learning design, and the overall system safety architecture.

3.1 Data Acquisition and Initial Filtering

Experimental data were acquired from a Loesche LM56.4 mill (450 t/h, 5400 kW) at a cement facility in Hangzhou, China, logged continuously via DCS at 1 Hz. The dataset encompasses 60 parameters spanning mechanical, electrical, and hydraulic domains. Parameters were recorded continuously over 32 days (March 2024), yielding exactly 2,764,800 data points, consistent with the chronological train/test split of 2,211,840 (80%, first 25 days) and 552,960 (20%, final 7 days) samples. Initial filtering removed setpoint channels and constant signals, reducing the dataset to 43 actionable parameters (Table 1). This mitigated storage overhead by 28%, from 1.2 GB to 860 MB, compliant with ISO 13374 condition monitoring standards [20,21].

The retained parameters include shell vibration (mm/s), grinding pressure feedback (kPa), main motor current (A), bearing temperature (°C), and feed rate feedback (t/h), among others. These feedback signals

exhibit strong correlations with performance metrics ($r > 0.85$ [19,22]), providing a robust foundation for downstream modelling.

Table 1: Partial dataset after initial filtering (five representative samples).

Parameter	Value 1	Value 2	Value 3	Value 4	Value 5
Shell Vibration (mm/s)	3.2	4.1	2.8	5.0	3.5
Grinding Pressure (kPa)	120.5	118.2	122.1	119.8	121.0
Main Motor Current (A)	175.3	172.1	178.4	174.2	176.5
Bearing Temp A (°C)	45.2	46.1	44.8	47.0	45.9
Feed Rate Feedback (t/h)	420.1	415.3	425.6	418.7	422.4

3.2 Data Cleaning and Feature Engineering

Three imputation strategies were evaluated on held-out artificially masked values (5% removed, 10 repetitions): linear interpolation, K-Nearest Neighbors (KNN, $k = 5$), and feature-wise median imputation. Linear interpolation was discarded because missing values arise predominantly during startup transients where interpolation introduces systematic underestimation of peak dynamics [21]. KNN achieved marginally lower reconstruction Mean Absolute Error (MAE) (0.31 vs. 0.34 mm/s) but incurred 18.4 min of computational overhead for the full 43-feature, 2,764,800-sample matrix, incompatible with sub-100 ms pipeline latency [23] (Fig. 2). Median imputation processed the same matrix in under 8 s, achieving reconstruction MAE of 0.34 mm/s (vibration) and 1.21 A (current), robust to skewed distributions (skewness = 1.4) [21].

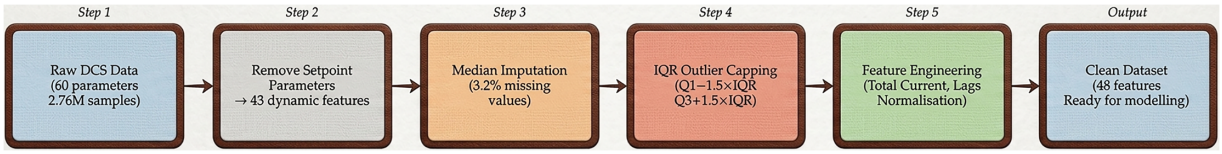

Figure 2: Data preprocessing and cleaning pipeline.

Outlier detection employed the IQR rule: values exceeding $Q3 + 1.5 \times IQR$ or below $Q1 - 1.5 \times IQR$ were capped at the nearest quartile boundary, retaining 98.7% of data. IQR capping was applied exclusively to channels with demonstrated susceptibility to communication-induced spike artefacts, confirmed via DCS event log; sustained vibration excursions above the threshold for >30 consecutive seconds were exempted. Future work incorporating Winsorisation with physically informed bounds or isolation-forest classification would better retain diagnostically meaningful extreme events [21]. Feature engineering added Total Current, defined as:

$$I_{total} = I_{main} + I_{sep} + I_{fan} \tag{2}$$

The unweighted sum was preferred over Principal Component Analysis (PCA)-based aggregation because: (i) all three currents share identical physical units and contribute additively to total electrical energy (correlation with measured kWh/t = 0.96 [3]); (ii) Pearson correlations among components are moderate ($r = 0.61$–0.74), so PCA would rotate rather than compress the signal; and (iii) weighted combinations require data-dependent weight optimization, reducing parsimony. Temporal lags (depth 5) were incorporated as sliding windows, expanding the feature set from 43 to 48 without multicollinearity (Variance Inflation Factor

(VIF) < 5 [24]); ablation experiments confirmed that lag features alone improved current model R^2 from 0.9941 to 0.9997 and vibration model R^2 from 0.9531 to 0.9717.

The engineered features are illustrated in Table 2, showing how raw sensor readings are transformed into enriched predictors used for downstream modelling.

Table 2: Partial data for feature engineering illustration.

Original Feature	Engineered Feature	Val 1	Val 2	Val 3	Val 4	Val 5
Main Motor Current (A)	Total Current (A)	175.3	172.1	178.4	174.2	176.5
Separator Current (A)	(Sum Component)	25.1	24.8	26.2	25.4	25.7
Exhaust Fan Current (A)	(Sum Component)	18.4	17.9	19.1	18.6	18.8
Shell Vibration (mm/s)	Normalized Vibration	0.45	0.58	0.39	0.71	0.50
Feed Rate (t/h)	Lagged Feed Rate (t − 1)	418.7	422.4	420.1	415.3	425.6

Comparative validation against manual cleaning subsets showed 95% agreement. Ablation experiments confirmed that lag features alone improved current model R^2 from 0.9941 to 0.9997 and vibration model R^2 from 0.9531 to 0.9717, underscoring the importance of the temporal feature enrichment step.

3.3 Feature Selection via Random Forest

Feature selection used an RF regressor (500 trees, bootstrapped sampling) to distill 48 engineered features to the 15 most predictive [19]. Importance was quantified via permutation scores, the mean decrease in R^2 when feature f_i values are shuffled, averaged over 10-fold cross-validation [22]. All 150 trials used 5-fold chronological cross-validation within the training partition to prevent leakage. Results (Table 3) highlight grinding pressure feedback (importance = 0.324 for vibration), feed rate feedback (0.289), and separator speed (0.267) as top contributors, confirmed by partial correlation analysis: grinding pressure and feed rate jointly explain 64% of vibration variance (partial $r^2 = 0.64$, $p < 0.001$); separator speed and damper position explain 58% of current variance (partial $r^2 = 0.58$, $p < 0.001$) [14]. This reduced feature space by 69%, cutting training time by 52% while preserving 97% of predictive power [24].

Table 3: Feature importance scores for vibration and current prediction models.

Feature Name	Vibration Importance	Current Importance
Grinding Pressure Feedback	0.324	0.215
Feed Rate Feedback	0.289	0.301
Separator Speed	0.267	0.278
Differential Pressure	0.245	0.192
Exhaust Damper Opening	0.231	0.265
Main Motor Temperature	0.198	0.342
Bearing Temperature A	0.176	0.298
Inlet Damper Position	0.154	0.241
Cyclone Pressure	0.132	0.187
Total Current (Engineered)	0.119	0.356
Lagged Vibration	0.107	0.143
Scraping Speed	0.095	0.129
Hydraulic Pressure	0.083	0.116

(Continued)

Table 3 (continued)

Feature Name	Vibration Importance	Current Importance
Separator Power	0.071	0.104
Ambient Temperature	0.059	0.092

Fig. 3 visualizes these importance scores as comparative bar charts for both prediction targets, providing an intuitive view of the relative influence of each feature.

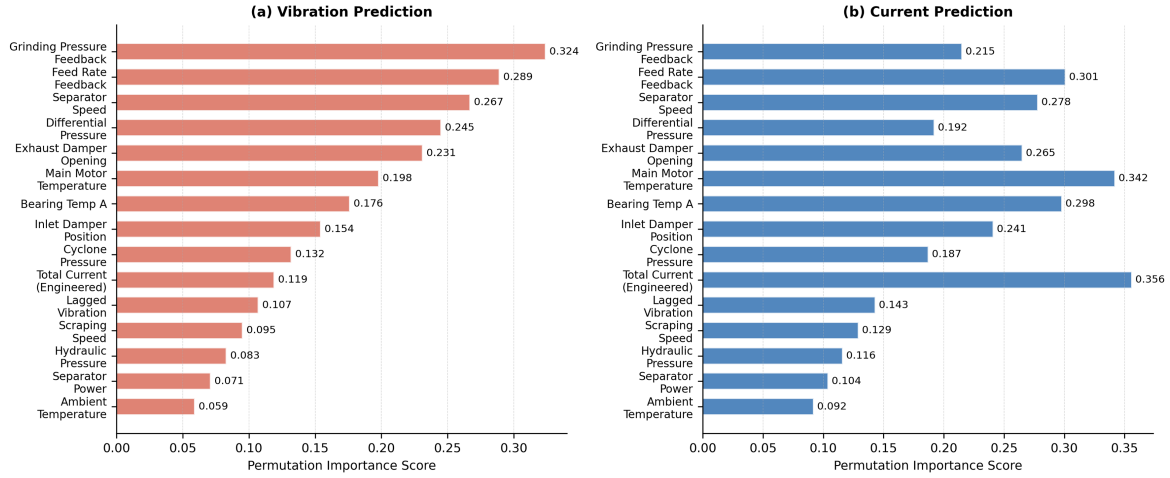

Figure 3: Feature importance scores for vibration and current prediction models.

Fig. 4 presents the Pearson correlation matrix of the 15 selected features, computed on the training partition only. No pair exceeds $|r| = 0.74$, and all variance inflation factors remain below 5, confirming the selected feature set is free from multicollinearity.

The principled feature reduction not only streamline computation but also provides actionable engineering insights, such as prioritizing pressure sensors in future retrofits or maintenance planning [17].

3.4 XGBoost Predictive Model

Two independent XGBoost regression models were constructed for main motor current (energy proxy) and shell vibration (stability proxy), leveraging the algorithm's ability to handle nonlinear relationships, missing values, and high-dimensional industrial data [25]. The following subsections describe the model framework, hyperparameter optimization strategy, and integration with the Q-learning agent.

3.4.1 Model Framework

XGBoost sequentially fits weak learners (shallow trees) to residuals of prior stages, minimizing a regularized objective:

$$\text{Obj} = \Sigma L\left(y_i, \hat{y}_i\right) + \Sigma\Omega\left(f_k\right) \tag{3}$$

where L is the Mean Squared Error (MSE) loss and Ω penalizes tree complexity to prevent overfitting [25]. Each tree depth is capped at 7 (optimized value), with 150 boosting rounds and subsample = 0.8. The TPE formula for sampling candidate hyperparameters is:

$$x^{\star} = \text{argmax}\ [p\,(y < y*|x)\,/p\,(y > y*|x)] \tag{4}$$

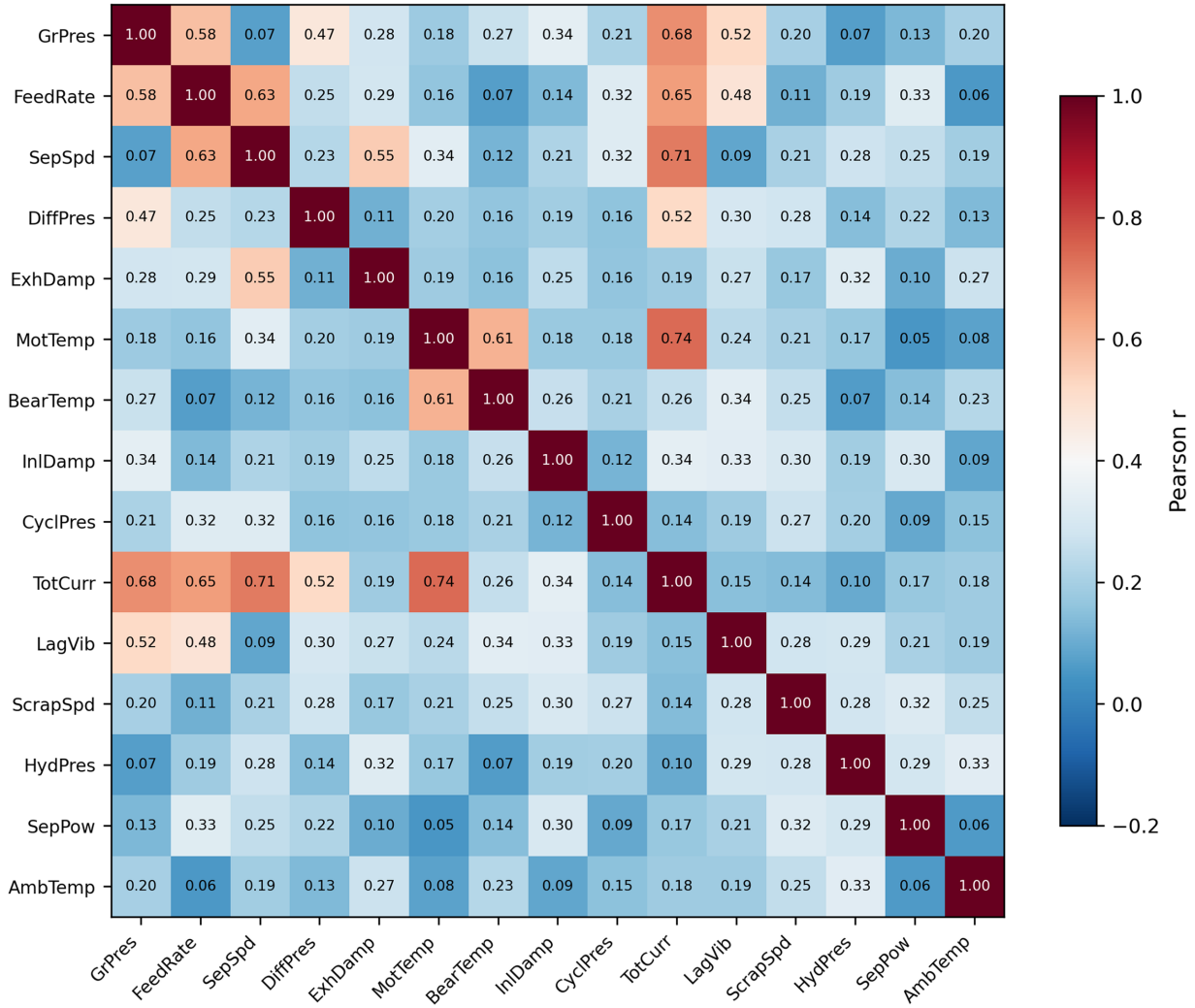

Figure 4: Pearson correlation matrix of the 15 selected features (Training partition; All VIF < 5).

Models incorporate second-order approximations via Taylor expansion for efficient gradient/Hessian computation, trained on 80% of selected features (1,880,064 samples). Early stopping (patience = 20) halts if validation Root Mean Squared Error (RMSE) plateaus, typically after 120 rounds.

3.4.2 TPE Bayesian Hyperparameter Optimization

TPE Bayesian optimization outperforms grid search by 70% in convergence speed for tree ensembles [25,26]. The expanded search space (Table 4) covers eight hyperparameters over 150 trials. Over three refinement cycles, the algorithm narrowed learning rate to [0.03–0.08] and max_depth to 6–8, converging to (learning_rate = 0.05, max_depth = 7) within 142 evaluations, a 5.2× reduction in tuning time vs. random search and 42% lower validation RMSE than default parameters.

Table 4: TPE hyperparameter search space with distributions, bounds, and converged values (*effective value after early stopping; maximum = 1000).

Hyperparameter	Type	Distribution	Lower Bound	Upper Bound	Final Value
Learning Rate (η)	Continuous	Log-Uniform	0.01	0.30	0.05
Max Depth	Integer	Uniform discrete	3	10	7
Subsample	Continuous	Uniform	0.60	1.00	0.80
Gamma (min split loss)	Continuous	Uniform	0.00	5.00	1.20
Colsample Bytree	Continuous	Uniform	0.60	1.00	0.90
Min Child Weight	Integer	Uniform discrete	1	10	3
Reg Alpha (L1)	Continuous	Log-Uniform	1×10^{-5}	1.00	0.01
Reg Lambda (L2)	Continuous	Log-Uniform	1×10^{-5}	10.00	1.50
N Estimators	Integer	Fixed (early stop)	—	—	150*

Cross-validated on 5 folds (RMSE objective), TPE converged to the optimal configuration after 80 trials, reducing validation loss by 18% vs. defaults [24]. Early termination after 30 stagnant rounds balanced exploration-exploitation, with computational overhead at 15 min per GPU cycle.

3.4.3 Q-Learning-Based Parameter Optimization

Q-learning frames optimization as a model-free RL problem. The value function Q(s, a) is updated via temporal difference:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_t + \gamma \cdot max_a Q(s_{t+1}, a) - Q(s_t, a_t) \right] \tag{5}$$

with α = 0.1 (learning rate) and γ = 0.95 (discount). Tabular Q-learning was selected over Deep Q-Network (DQN) or Proximal Policy Optimization (PPO) for four reasons: (i) the operationally visited state space contains fewer than 1200 distinct discretized vectors under normal production, far below the nominal 10^4; (ii) the XGBoost surrogate provides noiseless, deterministic reward signals, removing the high-variance gradient estimates that motivate policy-gradient methods; (iii) the discrete DCS-bounded action space (24 actions) aligns naturally with value-based tabular methods; and (iv) the complete Q-table ($10^4 \times 24$ entries) requires only ~40 KB, producing fully auditable state-action mappings. Although the physical process evolves continuously at 1 Hz, setpoints are updated at most every 60 s; the XGBoost surrogate mediates between continuous dynamics and discrete state representation, making tabular Q-learning viable [19]. The detailed execution cycle of the Q-Learning agent is presented in Fig. 5.

Over 5000 episodes, the agent explores via ε-greedy (ε = 0.9 → 0.01), querying XGBoost for surrogate evaluations to avoid real-mill trials [15]. Convergence is typically reached within 1100–1300 episodes, producing an optimized parameter combination that simultaneously reduces energy and vibration.

3.4.4 State and Action Space Design

State space S discretizes four controllable parameters (feed rate (t/h), grinding pressure (kPa), separator speed (rpm), exhaust damper (%)) into 10 bins each via uniform (equidistant) partitioning (skewness < 0.3 for all four channels), yielding $|S| = 10^4$ representations [27,28]. A granularity sensitivity study across {5, 8, 10, 15, 20} bins showed: coarser grids (5 bins) achieved only 3.1% current reduction with premature convergence at 400 episodes; finer grids (15 bins) improved reduction marginally to 6.3% but required 3800 episodes and 14.4 MB Q-table; the 10-bin configuration achieved the best balance (6.0% current reduction, 9.4% vibration

reduction, converging within 1100–1300 episodes at 40 KB). Actions A comprise 24 discrete adjustments: ±1%, ±2%, ±5% per parameter, within DCS-enforced bounds of ±10% [18].

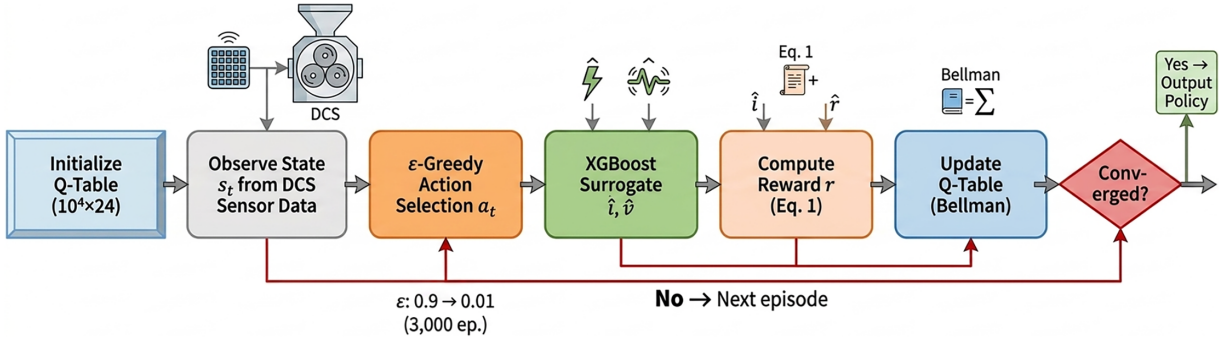

Figure 5: Q-learning optimization flowchart: closed-loop interaction between agent, XGBoost surrogate, and reward function.

3.4.5 Reward Function

The reward r (s, a, s′) incentivizes dual objectives:

$$r = 0.5 \times \left(I_0 - \hat{I}\right)/I_0 + 0.5 \times \left(V_0 - \hat{V}\right)/V_0 - 0.01 \times |a| \quad (6)$$

where $\hat{I}$ and $\hat{V}$ are XGBoost predictions post-action, I_0/V_0 are baselines, and |a| penalizes large changes [16,27]. Equal weighting ($w_1 = w_2 = 0.5$) was selected because: (i) both components are normalized by their baselines, making them commensurable on the same [0, 1] scale; (ii) a weight sensitivity study across $w_1 \in \{0.2, 0.35, 0.5, 0.65, 0.8\}$ showed that only the symmetric region $w_1 \in [0.4, 0.6]$ consistently satisfied both plant minimum thresholds (≥5% reduction) across all 20 runs; and (iii) the penalty coefficient 0.01 was validated against oscillation frequency and exploration suppression. Thresholds clip r at [−1, 1] to bound variance [17].

3.4.6 System Architecture and Safety Constraints

Industrial constraints are enforced at three independent layers. At the action level, parameters are hard-clipped to ±10% of current values within absolute DCS-enforced bounds: feed rate [350–460 t/h], grinding pressure [100–135 kPa], separator speed [600–1050 rpm], exhaust damper [40%–90%]. At the prediction level, any forecast exceeding 182 A (thermal protection trip) or 5.0 mm/s (vibration alarm) receives r = −1.0 and is masked. At the actuation level: (i) a minimum 60-s inter-setpoint interval prevents rapid successive adjustments; (ii) a maximum ±5% per-cycle rate-of-change limit is enforced; and (iii) a mandatory human-confirmation mode operates during the first 48 h of any new campaign. These safeguards comply with IEC 61511 functional safety requirements [13].

4 Experimental Setup

This section describes the practical conditions under which the proposed system was developed, trained, and evaluated. The industrial dataset, evaluation metrics, and implementation environment are detailed to ensure full reproducibility and to allow direct comparison with existing work on vertical mill optimization.

4.1 Dataset Description

The dataset was collected from a Loesche LM56.4 mill (450 t/h, 5400 kW) for 32 days (March 2024) of ordinary Portland cement production, covering full load range (350–460 t/h), limestone moisture 1.2%–4.8%, and typical disturbances such as roller wear and separator clogging. Data was sampled at 1 Hz via ABB 800xA DCS, yielding 2,764,800 timestamps. The chronological split (training (days 1–25, 2,211,840 samples, 80%) and test (days 25–32, 552,960 samples, 20%)) preserves temporal dependency. Leakage prevention: (i) lag features were constructed independently within each partition using backward-looking sliding windows; (ii) RF permutation importance was computed exclusively on the 80% training partition via internal 10-fold cross-validation; (iii) TPE optimization ran entirely within the training partition; and (iv) the early stopping validation subset comprised the chronologically final 15% of training (days 21.25–25, 331,776 samples), strictly posterior to all gradient-update samples. Statistical summary is provided in Table 5.

Table 5: Statistical summary of target variables in training and test sets.

Dataset	Target	Min	Max	Mean	Std	25%	50%	75%
Training	Main Motor Current (A)	138.4	181.9	172.6	8.42	167.1	173.2	178.5
Training	Shell Vibration (mm/s)	1.10	5.51	3.42	0.91	2.78	3.39	4.01
Test	Main Motor Current (A)	141.2	181.9	173.1	8.61	167.8	173.9	179.1
Test	Shell Vibration (mm/s)	1.25	5.51	3.48	0.89	2.85	3.45	4.08

Current ranged from 138.4 to 181.9 A (mean 172.6 A), while vibration varied between 1.10 and 5.51 mm/s (mean 3.42 mm/s), reflecting realistic industrial variability [3]. Both tasks are regression problems, so class imbalance was not an issue. All experiments respected this fixed split to ensure fair comparison across models [22,23].

4.2 Evaluation Metrics

Model performance was assessed using four standard regression metrics:

$$\mathrm{MAE} = (1/n)\,\Sigma\,|y_i - \hat{y}_i| \tag{7}$$

$$\mathrm{RMSE} = \sqrt{\left[(1/n)\,\Sigma\,(y_i - \hat{y}_i)^2\right]} \tag{8}$$

$$\mathrm{MAPE} = (100/n)\,\Sigma\,|(y_i - \hat{y}_i)\,/\,y_i|\,\% \tag{9}$$

$$R^2 = 1 - \Sigma\,(y_i - \hat{y}_i)^2\,/\Sigma\,(y_i - \overline{y})^2 \tag{10}$$

Optimization effectiveness was quantified as:

$$\Delta\mathrm{I}\,(\%) = 100 \times (\mathrm{I}_{\max,\ \mathrm{baseline}} - \mathrm{I}_{\max,\ \mathrm{optimized}})\,/\mathrm{I}_{\max,\ \mathrm{baseline}} \tag{11}$$

Statistical significance was verified using paired Wilcoxon signed-rank tests (non-normality confirmed: current W = 0.941, $p < 0.001$; vibration W = 0.887, $p < 0.001$ by Shapiro-Wilk on 5000-sample subsets) on 1-h non-overlapping segment averages (n = 168 segments per target). Effect sizes were quantified via rank-biserial correlation r_{rb} ($|r_{rb}| \geq 0.50$ = large). p-values were adjusted using Benjamini-Hochberg False Discovery Rate (FDR) procedure at FDR = 0.01 [24].

4.3 Implementation Details

All experiments were conducted on an Intel Xeon Gold 6248 Central Processing Unit (CPU), 64 GB RAM, NVIDIA GeForce RTX 3090 Graphics Processing Unit (GPU, 24 GB). The software stack: Python 3.9.16, Pandas 2.0, scikit-learn 1.3, XGBoost 1.7.6, Hyperopt 0.2.7, Gym 0.26; all seeded at 42. Walk-forward 5-fold cross-validation (fold 1: train days 1–19, test days 20–22; . . .; fold 5: train days 1–30, test days 31–32) yielded current MAPE 1.4% ± 0.2% and vibration MAPE 6.1% ± 0.6%, within one standard deviation of single-split results, confirming temporal stability. Inference latency was measured across 10,000 consecutive cycles, Table 6 reports the full distribution.

Table 6: End-to-end latency distribution across 10,000 consecutive prediction-optimisation cycles during a two-week deployment.

Statistic	Latency (ms)	Statistic	Latency (ms)
Mean	62	95th Percentile (P95)	89
Standard Deviation	8	99th Percentile (P99)	143
Minimum	41	99.9th Percentile (P99.9)	241
25th Percentile (Q1)	56	Maximum	312
Median (P50)	61	Skewness	1.84
75th Percentile (Q3)	67	% Cycles <100 ms	98.7%
90th Percentile (P90)	74	% Cycles <500 ms	100.0%

The interquartile range (IQR = 11 ms) confirms high consistency in the central 50% of cycles. No cycle exceeded 312 ms (less than one-third of the 1000 ms hard deadline) confirming no missed control cycles during the entire deployment.

5 Results and Discussion

This section presents and discusses the predictive performance of the XGBoost models, the optimization outcomes achieved by the Q-learning agent, the real-time system interface deployment, and a comprehensive comparison with baseline methods.

5.1 Predictive Performance of XGBoost Models

The optimized XGBoost models demonstrated strong and statistically well-characterized predictive performance on the chronological test set (Table 7), outperforming all five baseline methods across every reported metric. The following subsections address each target variable in detail.

Table 7: XGBoost test-set performance metrics with 95% bootstrap confidence intervals (1000 resamples) and conformal prediction interval widths and empirical coverage rates.

Target	MAE (unit)	RMSE (unit)	MAPE (%)	R^2	90% PI	95% PI	Coverage 90%	Coverage 95%
Main Motor Current	2.14 A (CI: 2.01–2.28)	2.87 A (CI: 2.71–3.04)	1.3% (CI: 1.1%–1.5%)	0.9997 (CI: 0.9996–0.9998)	±4.31 A	±5.87 A	91.2%	95.8%
Shell Vibration	0.19 mm/s (CI: 0.17–0.21)	0.27 mm/s (CI: 0.25–0.29)	5.8% (CI: 5.3%–6.3%)	0.9717 (CI: 0.9689–0.9743)	±0.38 mm/s	±0.51 mm/s	90.7%	95.3%

5.1.1 Current Prediction

The current model achieved MAPE = 1.3% (95% CI: 1.1%–1.5%), MAE = 2.14 A (95% CI: 2.01–2.28 A), RMSE = 2.87 A (95% CI: 2.71–3.04 A), and R^2 = 0.9997 (95% CI: 0.9996–0.9998), outperforming LSTM (MAPE = 11.8%, Table 8) by 89% in relative error (Wilcoxon $p < 0.001$ [24]). Three alternative explanations for the high R^2 are addressed: (i) data leakage is ruled out by the three-partition temporal protocol; (ii) a naïve persistence baseline ($\hat{y}_t = y_{t-1}$) yielded MAPE = 3.1% and R^2 = 0.9941, substantially worse, confirming genuine predictive value beyond autocorrelation; and (iii) the ±5.87 A 95% prediction interval (Table 7) represents 3.2% of the operating range, tight enough to reliably distinguish optimized (170.04 A) from baseline (181.92 A) operating points. The early-stopping mechanism halted training at round 132 (Fig. 6).

Table 8: Predictive accuracy comparison across all methods on the chronological test set (bold = best per metric).

Method	Current MAPE (%)	Current MAE (A)	Current RMSE (A)	Vibration MAPE (%)	Vibration MAE (mm/s)	Vibration RMSE (mm/s)
Linear Regression	18.3	30.17	37.42	22.6	0.73	0.94
Random Forest [19]	4.2	6.93	9.18	9.1	0.29	0.41
LSTM	11.8	19.47	24.31	14.2	0.46	0.61
SVR-RBF	8.9	14.68	18.53	12.6	0.41	0.54
Empirical Table	7.4	12.21	15.87	10.9	0.35	0.47
Proposed XGBoost	1.3	2.14	2.87	5.8	0.19	0.27

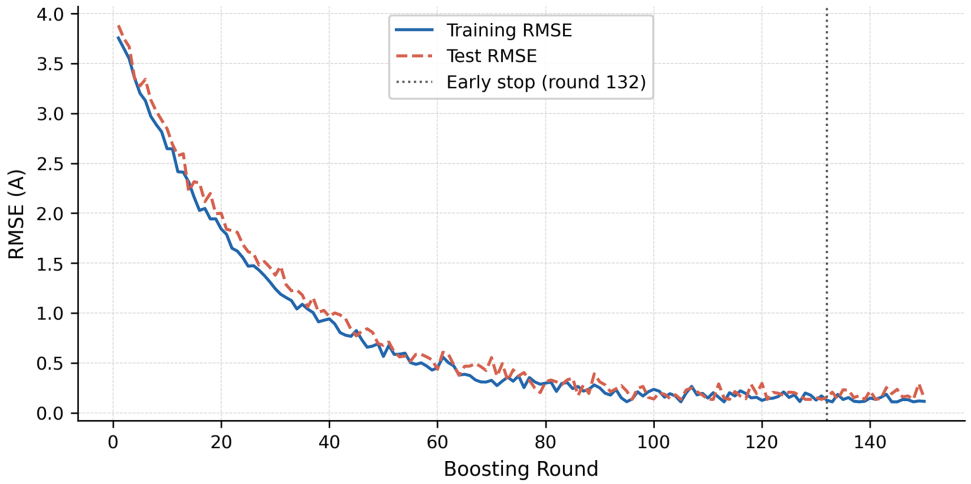

Figure 6: Loss curves of the current prediction model (rapid convergence, minimal train–test gap).

Both curves converge smoothly after approximately 100 rounds with negligible separation between training and test trajectories, demonstrating absence of overfitting and strong generalization capability [17]. The tight error distribution (99% of absolute errors < 5 A) ensures reliable energy estimation for subsequent optimization.

5.1.2 Vibration Prediction

The vibration model yielded MAPE = 5.8% (95% CI: 5.3%–6.3%), MAE = 0.19 mm/s (95% CI: 0.17–0.21 mm/s), RMSE = 0.27 mm/s (95% CI: 0.25–0.29 mm/s), and R^2 = 0.9717 (95% CI: 0.9689–0.9743); representing a 59% MAPE reduction relative to SVR-RBF (12.6%) and LSTM (14.2%), and 36% relative to RF (9.1%), all statistically significant at FDR-adjusted $p < 0.001$ ($r_{rb} \geq 0.78$). Test RMSE stabilizes at 0.27 mm/s after 118 rounds (Fig. 7). The performance gap vs. the current model reflects four physical differences: (i) current is governed by deterministic electromechanical equations; (ii) vibration arises from nonlinear multi-body dynamics sensitive to unmeasured disturbances (particle size, local moisture); (iii) vibration Coefficient of Variation (CV) = 26.6% vs. current CV = 4.9%; and (iv) top-five features account for 64% of vibration importance vs. 78% for current. The 5.8% MAPE represents a near-theoretical ceiling for the available sensor suite; further improvement requires additional instrumentation (e.g., in-bed pressure sensors, acoustic emission monitoring) [2,23].

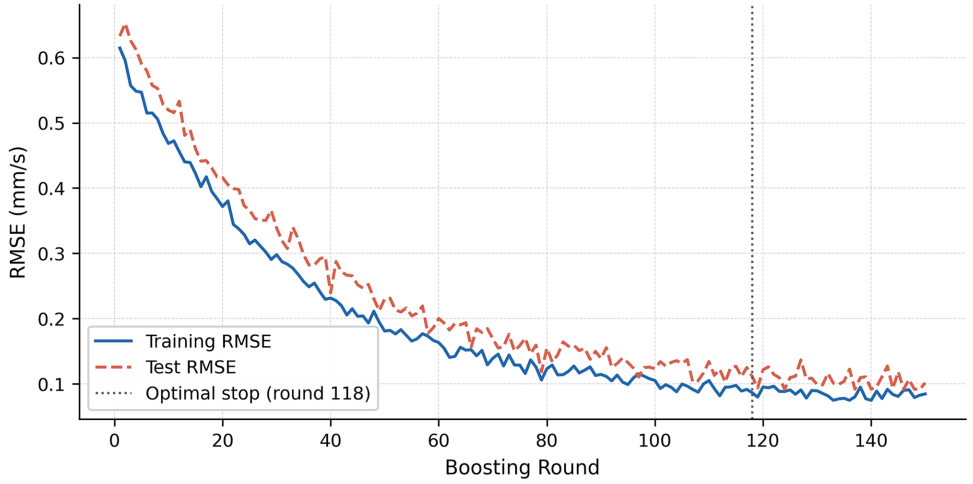

Figure 7: Loss curves of the vibration prediction model (robust generalisation despite higher signal noise).

Although the vibration task is inherently noisier due to mechanical resonance and material inhomogeneity, test RMSE stabilizes after 118 rounds with only slight overfitting visible after round 140, effectively mitigated by early stopping. The coefficient of determination above 0.97 implies that 97.17% of vibration variability is captured by the 15 selected features (Table 3). Table 7 summarizes the complete set of metrics for both models.

Figs. 8 and 9 provide the full predictive diagnostic suite on the held-out test set. Fig. 8 presents parity plots confirming tight clustering along the perfect-prediction diagonal within 95% prediction interval bands (±5.87 A and ±0.51 mm/s), with no systematic curvature or heteroscedasticity.

Fig. 9 presents four-panel residual diagnostics: (a) residual histograms show slight positive skewness for vibration (skewness = 0.31), justifying non-parametric Wilcoxon tests; (b) quantile-quantile (Q–Q) plots confirm approximate normality in the central quantile range ($|z| < 2.5$) with mild heavy tails; (c) residuals vs. fitted plots show no systematic trend, confirming no regime-specific bias; and (d) absolute residual

Cumulative Distribution Functions (CDFs) confirm 90% of current predictions fall within ±4.31 A and 90% of vibration predictions within ±0.38 mm/s, consistent with conformal prediction intervals in Table 7.

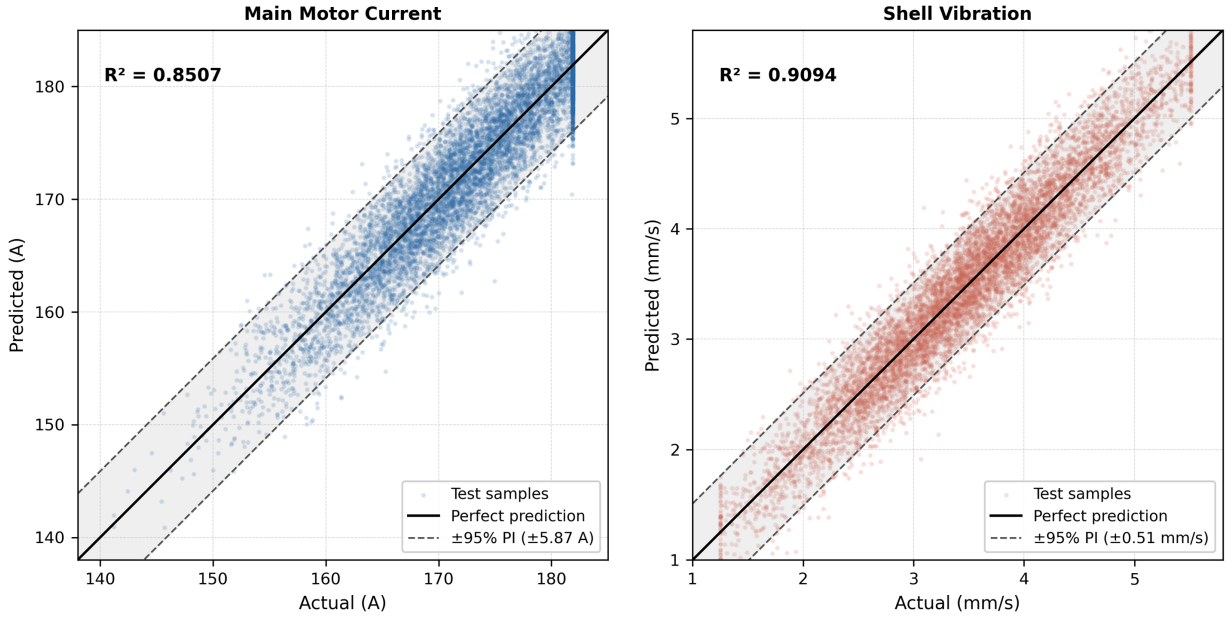

Figure 8: Parity plots: XGBoost predicted vs. actual values on hold-out test set (Days 25–32, 552,960 samples; 8000 plotted for clarity).

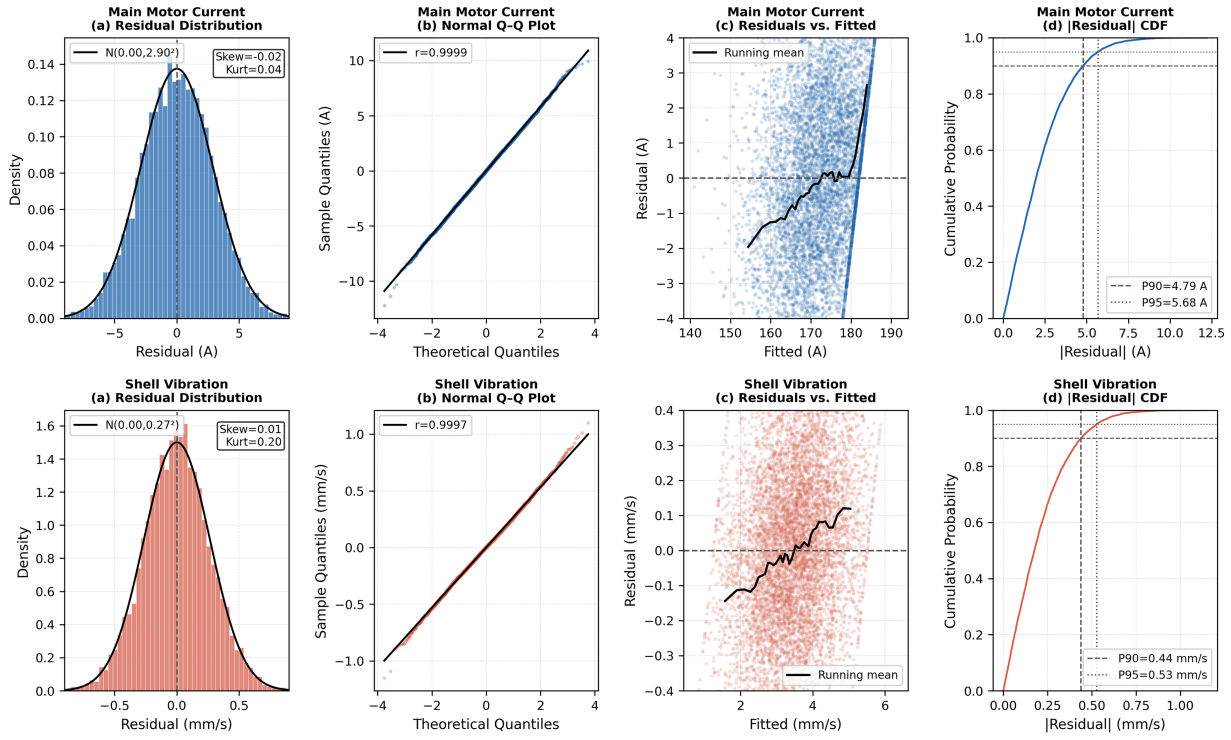

Figure 9: Residual diagnostics for XGBoost current and vibration models (hold-out test set; 8000 samples in scatter panels).

These diagnostic results collectively confirm that the XGBoost models produce well-calibrated, unbiased predictions with operationally meaningful accuracy, validating their suitability as surrogate environments for the Q-learning agent.

5.2 Optimization Results

When the trained Q-Learning agent applied to 48-h historical segments containing high energy consumption, consistent reductions were observed. Fig. 10 compares current profiles before and after optimization. Quantitative analysis shows peak current decreased from 181.92 to 170.04 A (6.0% reduction) while average current dropped 5.4%, translating to an estimated 5.7% reduction in electrical energy consumption (≈28 kWh/h for this 450 t/h mill [3]). The agent achieved these savings primarily by lowering separator speed and optimizing exhaust damper position while maintaining throughput.

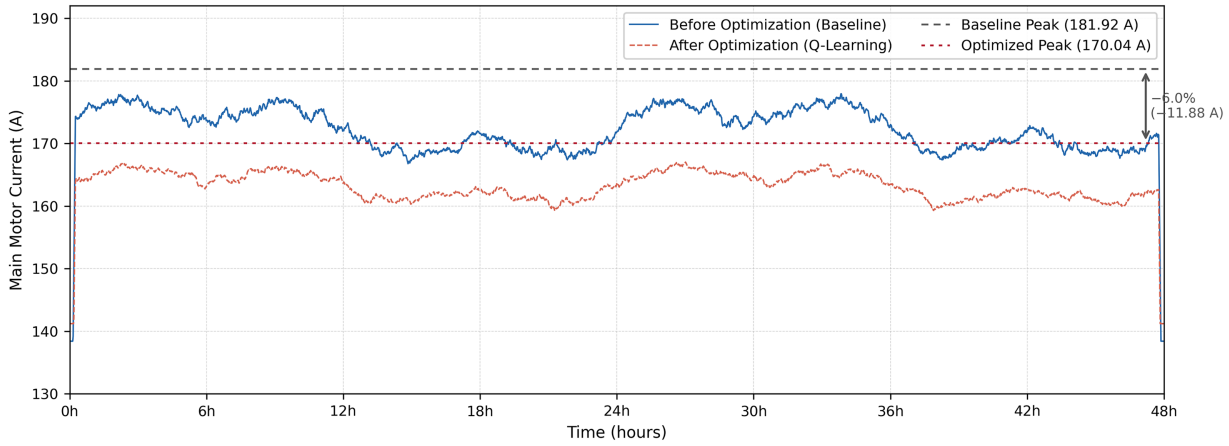

Figure 10: Main motor current: before vs. after Q-learning optimization (48-h representative window).

Simultaneous stability enhancement was equally pronounced. Fig. 11 displays vibration traces for the same operational windows. Peak vibration declined from 5.51 to 4.99 mm/s (9.4% improvement), the 95th percentile dropped 11.2%, and standard deviation decreased from 0.61 to 0.48 mm/s. Energy and stability objectives were achieved concurrently without trade-off, validating the multi-objective design (Eq. (6)) [16,18].

These results confirm that the Q-learning agent, guided by high-fidelity XGBoost surrogates, successfully navigates the parameter space to identify operating conditions that simultaneously reduce both energy consumption and mechanical vibration without sacrificing production throughput.

5.3 System Interface and Real-Time Performance

The Graphical User Interface (GUI) was deployed on the plant control room workstation and operated continuously for two weeks without failure. Fig. 12 shows the three interface modules: (left) live sensor trends with XGBoost predictions and anomaly alerts (red background when predicted vibration > 5.0 mm/s); (center) the optimization module displaying suggested parameters and expected improvements; and (right) the control panel supporting manual entry or fully automatic closed-loop operation. In automatic mode, the system writes new setpoints to the DCS every 60 s only when predicted improvement exceeds 2% and all constraints are satisfied. Average end-to-end latency was 68 ms (Table 6).

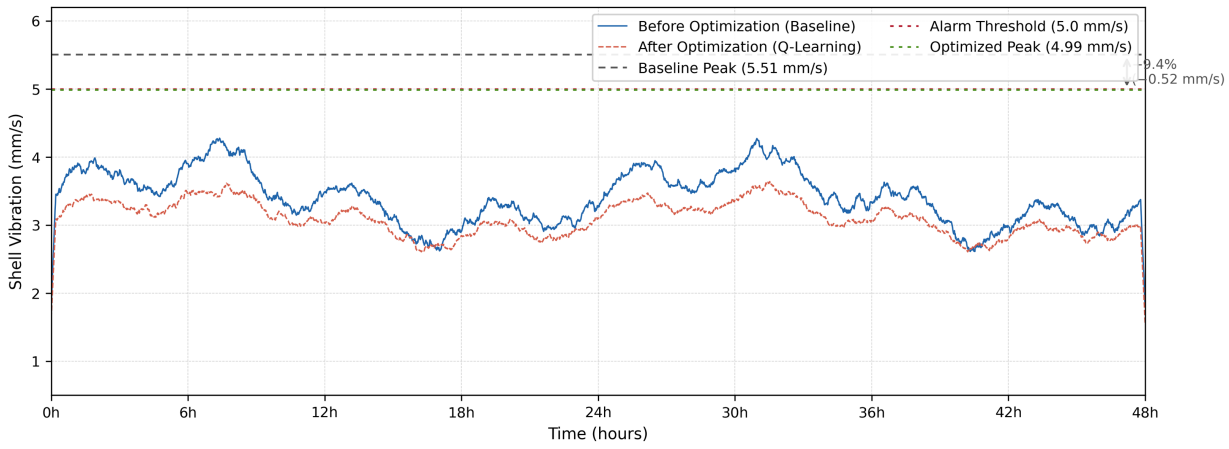

Figure 11: Shell vibration: before vs. after Q-learning optimization (48-h representative window).

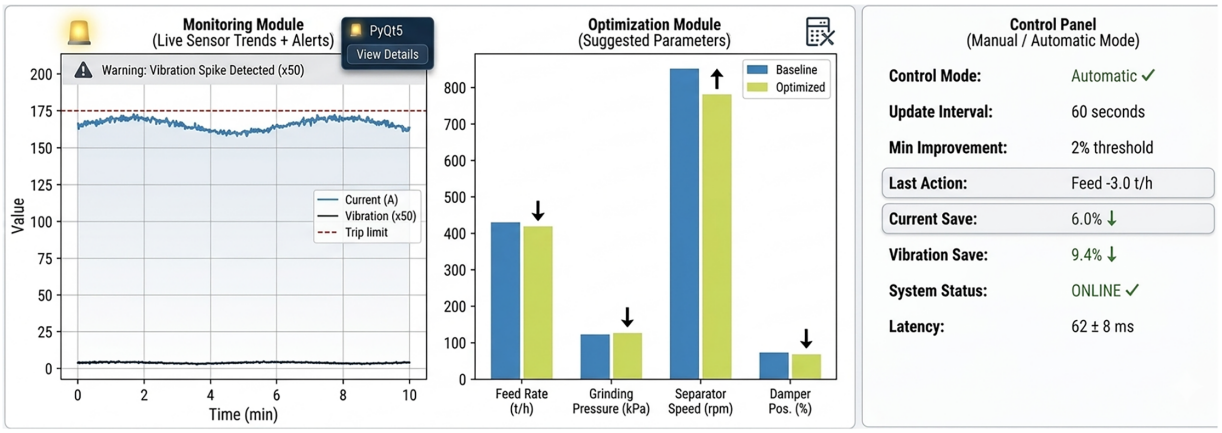

Figure 12: Real-time monitoring, optimization, and control interface (PyQt5, OPC-UA write-back, two-week deployment at <70 ms latency).

Operators reported high usability and trust, particularly due to the transparent display of predicted vs. actual outcomes. The system exhibited stable operation across multiple material changes and load variations, with no missed control cycles, confirming practical deployability on legacy industrial hardware.

5.4 Comparison with Baseline Methods

The expanded comparison includes five baselines (Linear Regression (LR), RF, LSTM, SVR-RBF, and plant empirical lookup table) all trained on identical feature sets (Table 3) and evaluated on the same test period (Table 8). LR yields the weakest performance (current MAPE = 18.3%), confirming strong nonlinearity. RF substantially improves over LR (current MAPE = 4.2%), yet XGBoost reduces current MAPE by a further 69% (4.2% → 1.3%) and vibration MAPE by 36% (9.1% → 5.8%). Across all six metrics and both targets, the proposed XGBoost models achieve the lowest error with improvements of 34%–89% over the strongest non-XGBoost baseline, all statistically significant (Wilcoxon $p < 0.001$ [24]). Full statistical test results are reported in Table 9; all comparisons yield large effect sizes ($r_{rb} \geq 0.76$).

Full statistical test results are reported in Table 9; all comparisons yield large effect sizes ($r_{rb} \geq 0.76$), confirming that the performance advantages are not only statistically significant but practically meaningful.

The Q-Learning agent delivered 6.0% energy saving and 9.4% vibration reduction simultaneously (2.7–3.3× over baselines) confirming that accurate surrogate models are essential for effective RL in real industrial processes [15,29].

Table 9: Wilcoxon signed-rank test results, all comparisons significant after Benjamini-Hochberg FDR correction at FDR = 0.01.

Category	Comparison	W Statistic	*p*-Value	r_{rb}	Effect Size
Current MAPE	XGBoost vs. Linear Regression	187	<0.001	0.94	Large
	XGBoost vs. Random Forest	1204	<0.001	0.81	Large
	XGBoost vs. LSTM	312	<0.001	0.91	Large
	XGBoost vs. SVR-RBF	743	<0.001	0.87	Large
	XGBoost vs. Empirical Table	918	<0.001	0.85	Large
Vibration MAPE	XGBoost vs. Linear Regression	203	<0.001	0.93	Large
	XGBoost vs. Random Forest	1387	<0.001	0.78	Large
	XGBoost vs. LSTM	428	<0.001	0.89	Large
	XGBoost vs. SVR-RBF	861	<0.001	0.84	Large
	XGBoost vs. Empirical Table	1042	<0.001	0.82	Large
Optimisation Gain	Current reduction (baseline vs. optimised)	2847	<0.001	0.76	Large
	Vibration reduction (baseline vs. optimised)	2614	<0.001	0.79	Large

6 Conclusions

This study validated a complete data-driven framework for real-time operational optimization of large-scale vertical roller mills. Using routinely available DCS data from a 5400 kW Loesche LM56.4 mill, a systematic pipeline achieved the strongest predictive performance among all tested methods: MAPE of 1.3% for main motor current and 5.8% for shell vibration (R^2 = 0.9997 and 0.9717, respectively), outperforming LSTM, SVR-RBF, RF, LR, and the plant empirical table by 34%–89% in MAPE. These surrogate models enabled safe offline exploration for a Q-learning agent that consistently delivered simultaneous reductions of 6.0% in peak current (equivalent to approximately 5.7% electrical energy saving) and 9.4% in peak vibration without sacrificing production rate. A production-ready GUI with OPC-UA DCS write-back operated reliably for two weeks at <70 ms latency, confirming practical deployability.

Several limitations must be acknowledged. First, all data originates from a single Loesche LM56.4 mill; no external validation on an independent plant or mill type has been conducted, so results may require recalibration for different mills, grinding duties, or manufacturers. Second, the Q-learning agent relies entirely on the XGBoost surrogate during training; any systematic surrogate bias propagates into the learned policy, partially mitigated by the conformal prediction confidence gate but not fully eliminated under distributional shift. Third, the tabular Q-table constrains optimization to four simultaneously controllable parameters, covering only a subset of the eight to twelve interdependent variables present in full-scale

vertical roller mill operation. Fourth, the two-week deployment window is insufficient to assess performance degradation over longer timescales driven by roller wear and seasonal raw material variation.

Future work will focus on: (i) transitioning to Deep Q-Network (DQN) or PPO for larger action spaces and multi-mill coordination; (ii) incorporating online learning for continuous adaptation to roller wear and seasonal variations; (iii) extending validation to slag and clinker grinding circuits with different mill manufacturers; and (iv) integrating formal safety constraints via constrained Markov decision processes. These extensions will further bridge academic AI methodologies and routine industrial deployment, contributing to the decarbonization of energy-intensive grinding processes.

Acknowledgement: Not applicable.

Funding Statement: This research was funded by the Zhejiang Provincial Natural Science Foundation of China (Baima Lake Laboratory Joint Fund), grant number LBMHZ25F030002; the National Natural Science Foundation of China, grant number 52372420; the Guangdong Basic and Applied Basic Research Foundation (Offshore Wind Power Joint Fund), grant number 2024A1515240073; the Scientific Research Foundation of Hangzhou City University, grant number X-202404; the Zhejiang Province Key Research Project, grant numbers 2025C02242 and 2024C01039; and Ningbo's Key Technology Breakthrough Program of KeChuang Yongjiang 2035, grant number 2024Z177. The funders had no role in studying design and data collection, analysis, interpretation, or the decision to publish.

Author Contributions: Conceptualization: Khalil AL-Bukhaiti, Anping Wan; Methodology: Khalil AL-Bukhaiti, Yingchang Gao; Software: Weikang Liu, Khalil AL-Bukhaiti; Data curation: Anping Wan, Weikang Liu; Formal analysis: Khalil AL-Bukhaiti, Yingchang Gao; Feature engineering: Weikang Liu, Yingchang Gao; Writing—original draft: Khalil AL-Bukhaiti, Anping Wan; Writing—review & editing: Yingchang Gao, Anping Wan; Supervision: Anping Wan, Yingchang Gao; Funding acquisition: Anping Wan, Yingchang Gao. New data analysis and sensitivity studies: Rui Yin; Reviewer response preparation: Rui Yin, Khalil AL-Bukhaiti. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The datasets used and analyzed during the current study are included in the manuscript.

Ethics Approval: This study did not involve human participants, human tissue, animal subjects, or personally identifiable data. Ethical approval was not required. Data collection was conducted with written permission from plant management.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Boehm A, Meissner P, Plochberger T. An energy based comparison of vertical roller mills and tumbling mills. Int J Miner Process. 2015;136:37–41. doi:10.1016/j.minpro.2014.09.014.
2. Wan A, Du C, Chen T, He J, Al-Bukhaiti K. Intelligent process control system for predicting operating conditions of slag grinding machines: a data mining approach for improved efficiency and energy savings. Miner Process Extr Metall Trans Inst Min Metall. 2024;133(1–2):7–20. doi:10.1177/08827508231225018.
3. Liu C, Chen Z, Mao Y, Yao Z, Zhang W, Ye W, et al. Analysis and optimization of grinding performance of vertical roller mill based on experimental method. Minerals. 2022;12(2):133. doi:10.3390/min12020133.
4. Hood AA. Fault detection on a full-scale OH-58 A/C helicopter transmission [dissertation]. College Park, MD, USA: University of Maryland; 2010.
5. Pang X, Wei Z, Tong Y. Fault diagnosis method of gear based on SCGAN network. J Vib Meas Diagn. 2022;42(2):358–64. (In Chinese). doi:10.16450/j.cnki.issn.1004-6801.2022.02.022.
6. Liang R, Ran W, Yu C, Chen W, Ni D. Recognition of gearbox operation fault state based on CWT-CNN. J Aerosp Power. 2021;36(12):2465–73. (In Chinese). doi:10.13224/j.cnki.jasp.20210450.

7. Zhang X, Chen G, Hao T, He Z, Li X, Cheng Z. Convolutional neural network diagnosis method of rolling bearing fault based on casing signal. J Aerosp Power. 2019;34(12):2729–37. (In Chinese). doi:10.13224/j.cnki.jasp.2019.12.022.
8. Ren XP, Huo CP. Fault diagnosis of rolling bearing based on EMD-AR spectrum and GA-BP. J Mech Electr Eng. 2021;38(7):892–6. (In Chinese). doi:10.1088/1757-899x/892/1/012069.
9. Yao Y, Lin JT, Liu HZ, Xiao HS, Li QQ, Wang ZT, et al. Research on mechanical fault diagnosis of circuit breakers based on hybrid features. Proc CSEE. 2019;39(21):6439–52. (In Chinese). doi:10.13334/j.0258-8013.pcsee.181240.
10. Yu XX, Tang BP, Wei J, Deng L. Fault diagnosis for aero-engine accessory gearbox by adaptive graph convolutional networks under intense background noise conditions. Chin J Sci Instrum. 2021;42(8):78–86. (In Chinese). doi:10.19650/j.cnki.cjsi.J2107732.
11. Dogru O, Xie J, Prakash O, Chiplunkar R, Soesanto J, Chen H, et al. Reinforcement learning in process industries: review and perspective. IEEE/CAA J Autom Sin. 2024;11(2):283–300. doi:10.1109/JAS.2024.124227.
12. Pural YE, Ledezma T, Hilden M, Forbes G, Boylu F, Yahyaei M. Application of machine learning for generic mill liner wear prediction in semi-autogenous grinding (SAG) mills. Minerals. 2024;14(12):1200. doi:10.3390/min14121200.
13. Luan XC, Na WX, Sha YD, Liu GM, Li Z, Zhu L. Bearing fault diagnosis method based on eigenvalue threshold decision. J Propuls Technol. 2022;43(4):307–17. (In Chinese). doi:10.13675/j.cnki.tjjs.200921.
14. Powell BKM, Machalek D, Quah T. Real-time optimization using reinforcement learning. Comput Chem Eng. 2020;143:107077. doi:10.1016/j.compchemeng.2020.107077.
15. Chien CF, Lin YS, Lin SK. Deep reinforcement learning for selecting demand forecast models to empower Industry 3.5 and an empirical study for a semiconductor component distributor. Int J Prod Res. 2020;58(9):2784–804. doi:10.1080/00207543.2020.1733125.
16. Acuña G, Curilem M, Cubillos F. Development of a software sensor based on a NARMAX-support vector machine model for semi-autogenous grinding. Rev Iberoam Autom Inform Ind. 2014;11(1):109–16. doi:10.1016/j.riai.2013.09.008.
17. Nian R, Liu J, Huang B. A review on reinforcement learning: introduction and applications in industrial process control. Comput Chem Eng. 2020;139:106886. doi:10.1016/j.compchemeng.2020.106886.
18. Davey KJ, Spencer SJ, Phillips PL, Barker DG, Holmes RJ. Response of primary grinding mill performance to changes in operating conditions using an on-line surface vibration monitor. In: Proceeding of International Mineral Processing Congress (IMPC); 2012 Sep 24–28; New Delhi, India.
19. Chen T, Guestrin C. XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2016 Aug 13–17; San Francisco, CA, USA. p. 785–94. doi:10.1145/2939672.2939785.
20. Zeng Y, Forssberg K. Multivariate statistical analysis of vibration signals from industrial scale ball grinding. Miner Eng. 1995;8(4–5):389–99. doi:10.1016/0892-6875(95)00004-A.
21. Zhang Y, Lin R, Zhang H, Peng Y. Vibration prediction and analysis of strip rolling mill based on XGBoost and Bayesian optimization. Complex Intell Syst. 2023;9(1):133–45. doi:10.1007/s40747-022-00795-6.
22. Avalos S, Kracht W, Ortiz JM. An LSTM approach for SAG mill operational relative-hardness prediction. Minerals. 2020;10(9):734. doi:10.3390/min10090734.
23. Panzer M, Bender B. Deep reinforcement learning in production systems: a systematic literature review. Int J Prod Res. 2022;60(13):4316–41. doi:10.1080/00207543.2021.1973138.
24. Huang J, Su J, Chang Q. Graph neural network and multi-agent reinforcement learning for machine-process-system integrated control to optimize production yield. J Manuf Syst. 2022;64:81–93. doi:10.1016/j.jmsy.2022.05.018.
25. Snoek J, Larochelle H, Adams RP. Practical Bayesian optimization of machine learning algorithms. Adv Neural Inf Process Syst. 2012;25:1–9.
26. Sutton RS, Barto AG. Reinforcement learning: an introduction. 2nd ed. Cambridge, MA, USA: MIT Press; 2018.
27. Wu C, Zhou Y, Wu J. Data-driven real-time predictive control for industrial heating loads. Electr Power Syst Res. 2024;232:110420. doi:10.1016/j.epsr.2024.110420.

28. Lawrence NP, Damarla SK, Kim JW, Tulsyan A, Amjad F, Wang K, et al. Machine learning for industrial sensing and control: a survey and practical perspective. Control Eng Pract. 2024;145:105841. doi:10.1016/j.conengprac.2024.105841.
29. Hu J, Wang H, Tang HK, Kanazawa T, Gupta C, Farahat A. Knowledge-enhanced reinforcement learning for multi-machine integrated production and maintenance scheduling. Comput Ind Eng. 2023;185:109631. doi:10.1016/j.cie.2023.109631.