

KARADENIZ TECHNICAL UNIVERSITY
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

DEPARTMENT OF THE MINING ENGINEERING

**CUTTING PERFORMANCE OPTIMIZATION OF LABORATORY SCALE
ROTARY ROCK CUTTING HEAD BY APPLICATION OF MACHINE
LEARNING METHODS**

DOCTORATE THESIS

Abdolsattar ROUDINI

OCTOBER 2025
TRABZON



**KARADENİZ TECHNICAL UNIVERSITY
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES**

DEPARTMENT OF THE MINING ENGINEERING

**CUTTING PERFORMANCE OPTIMIZATION OF LABORATORY SCALE
ROTARY ROCK CUTTING HEAD BY APPLICATION OF MACHINE
LEARNING METHODS**

Abdolsattar ROUDINI

**This thesis is accepted to give the degree of
DOCTOR OF PHILOSOPHY
By
The Graduate School of Natural and Applied Science at
Karadeniz Technical University**

The Date of Submission : 18 / 09 / 2025

The Date of Examination : 13 / 10/ 2025

Thesis Supervisor : Prof. Dr. Kerim AYDINER

Trabzon 2025

PREFACE

This thesis was supported by the Scientific Research Projects Coordination Unit (BAP) under Project No. FBA-2020-8537, and by the Scientific and Technological Research Council of Türkiye (TÜBİTAK) under Project No. 123M879. I am grateful for this fundings, which enabled the experimental work and analysis presented here.

I extend my sincere thanks to my supervisor, Prof.Dr. Kerim AYDINER, for his guidance, patience, and high expectations throughout the study. I also thank Dr. Serkan İNAL for his steady help and support at key stages of the work.

My appreciation goes to our bachelor students, Orçun Latifoğlu and Mertcan BEŞLİ, for their diligent assistance during sample preparation and the rock-cutting tests. I also thank Dr. Melek Hanım BEŞER for her practical help during the rock mechanics tests.

I am deeply grateful to my family and friends for their unwavering encouragement and emotional support throughout this marathon.

Abdolsattar ROUDINI
Trabzon 2025

DECLARATION

I declare that this thesis entitled “Cutting Performance Optimization of Laboratory Scale Rotary Rock Cutting Head by Application of Machine Learning Methods” is my own original work. It has not been submitted, in whole or in part, to any degree or diploma at any other university or institution. All sources of information have been duly acknowledged through proper citation. The experimental program, analyses, figures, and tables are the result of my work except where explicitly stated otherwise. This research was conducted under the supervision of Prof. Kerim AYDINER. It was financially supported by the Scientific Research Projects Coordination Unit (BAP, Project No. FBA-2020-8537) and by the Scientific and Technological Research Council of Türkiye (TÜBİTAK, Project No. 123M879). No human participants or live animals were involved. All laboratory work complied with institutional health, safety, and ethical guidelines. I understand the university’s policies on academic integrity and affirm that this work complies with those policies.13.10.2025

Abdolsattar ROUDINI

CONTENTS

	<u>Page No</u>
PREFACE.....	III
DECLARATION	IV
CONTENTS	V
ÖZET	VIII
SUMMARY	IX
LIST OF FIGURES	X
LIST OF TABLES	XIV
LIST OF SYMBOLS AND ABBREVIATIONS.....	XVI
1. INTRODUCTION	1
1.1. Background & Motivation	1
1.1.1. Tunneling And Mechanical Excavation: Role of Roadheaders.	2
1.1.2. Industrial Pain Points: Productivity, Energy Use, Pick Wear, Safety.....	3
1.1.3. Variability Of Rock Mass Properties and Need for Adaptive Control	4
1.1.4. Digitalization in mining:	5
1.2. Problem Context & Statement	6
1.2.1. Scientific Gap.....	6
1.2.2. Limitations of Empirical and Physics-Only Approaches.....	6
1.3. Research Questions & Hypotheses	7
1.3.1. Research Questions	7
1.3.2. Hypotheses Aligned with Each Research Question.	7
1.4. Mechanical Excavation.....	8
1.5. Roadheader Types and Applications	9
1.6. Conical Picks and Excavation Principles	12
1.6.1. Most Important Parameters of Conical Picks	12
1.6.2. Torque And Power on a Cutterhead.....	13
1.7. Roadheader Performance Prediction Models	14
1.8. Automation in Roadheaders	16
1.8.1. Cutting Speed and Energy Management.....	16
1.8.2. Control Strategies and Implementation Pathways	17
1.8.3. Sensing, Measurement, and Environmental Limits	17
1.8.4. Precision Cutting, Dynamics, and Pose Compensation	17

1.8.5. Path Planning and Autonomous Motion	18
1.9. Artificial Intelligence and Machine Learning.....	18
1.10. Machine Learning Application in Roadheader Automation	21
1.10.1. Perception Of Ground Conditions and Property Recognition	21
1.10.2. Cutting-Load Identification and Adaptive Speed Control	21
1.10.3. Motion Control, Rectification, and Posture Regulation.....	23
1.10.4. Trajectory Planning and Path Tracking	24
1.10.5. Performance Prediction.....	24
1.10.6. Fault Diagnosis and Anomaly Detection	24
1.11. Organization of the Thesis	25
2. MATERIAL AND METHODS.....	26
2.1. Materials	26
2.1.1. Materials Used for Samples Preparation.....	26
2.1.2. Design of Casting Molds.....	28
2.1.3. Samples Casting.....	29
2.1.4. Samples Curing.....	30
2.2. Manufacturing of Cutting Head.....	32
2.3. Cutting Head Modernization and Sensor Integration	35
2.4. Mechanical Properties of Samples.....	39
2.4.1. Uniaxial Compressive Strength (UCS).....	39
2.4.2. Modulus of Elasticity (E).....	40
2.4.3. Indirect Tensile Strength (ITS) or BTS.....	43
2.5. Experimental Design of Excavation Tests.....	44
2.5.1. Full Factorial Test Design.....	45
2.5.2. Experimental Design of Excavation Tests	45
2.6. Rock Cutting Tests	47
2.7. Data Acquisition and Cleaning.....	52
2.8. Most Important Feature Detection.....	55
2.8.1. Linear Regression	55
2.8.2. Least Absolute Shrinkage and Selection (Lasso) Regression.....	56
2.8.3. Ridge Regression (RidgeCV).....	57
2.8.4. Random Forest Regression	59
2.8.5. Gradient Boosting	61
2.9. Particle Swarm Optimization.....	63

2.10. Model Performance Evaluation.....	66
2.10.1. Coefficient of Determination (R^2).....	66
2.10.2. Mean Absolute Error (MAE)	66
2.10.3. Root Mean Squared Error (RMSE).....	67
2.10.4. Mean Absolute Percentage Error (MAPE)	67
3. RESULTS AND DISCUSSION	69
3.1. Mechanical Properties	69
3.2. Dominant Machine-Frame Vibration Axis During Cutting.....	71
3.3. Most Important Feature	79
3.3.1. Most Important Feature Recognition for Specific Energy.....	81
3.3.2. Most Important Feature Recognition for Vibration-X.....	88
3.4. Optimization of Cutting Variables.....	98
3.4.1. Optimization Results and Trade-off Analysis.....	101
3.5. Pick Wear Assessment.....	103
4. CONCLUSIONS	106
4.1. Answers to the Research Questions.....	106
4.2. Limitations.....	107
4.3. Future Research	107
5. REFERENCES	109
6. APPENDIX.....	120
RESUME	

Doktora Tezi

ÖZET

MAKİNE ÖĞRENİMİ YÖNTEMİYLE LABORATUVAR ÖLÇEKLİ KESİCİ KAFA KESME PERFORMANSI OPTİMİZASYONU

Abdolsattar ROUDINI

Karadeniz Teknik Üniversitesi
Fen Bilimleri Enstitüsü
Maden Mühendisliği Anabilim Dalı
Danışman: Prof. Dr. Kerim AYDINER
2025, 149 Sayfa

Bu çalışma, özgül enerjiyi ile makine gövdesi titreşimini azaltmaya yönelik veri odaklı bir yaklaşım geliştirerek konik kesicilerle donatılmış laboratuvar ölçekli bir kesme kafasının kesme performansını incelemektedir. Yardımsız kazı yöntemi ve $3 \times 3 \times 3$ tam faktöriyel deney tasarımı uygulanarak toplamda 243 deney yapılmıştır. Deneylerde güç tüketimi, ilerleme hızı, iki eksenli titreşim (X ve Z) ve gerçek kesme derinliği kaydedilmiştir. Kesme yönüyle aynı hızda olan X eksenli en yüksek titreşimi göstermiş ve yaklaşık 86 mm/s ve 141 mm/s olmak üzere iki eşik, titreşim davranışını malzeme dayanımına uygun biçimde ayırmıştır. Çalışma parametreleri ile özgül enerji ve titreşim arasındaki ilişkiyi incelemek üzere denetimli makine öğrenmesi modelleri eğitilmiştir. Ağaç tabanlı topluluk modelleri, özgül enerjiyi yüksek doğrulukla tahmin etmiş; çapraz doğrulamalı R^2 değerleri 0,93–0,97 arasında bulunmuş ve kesme derinliği en etkili değişken olarak belirlenmiştir. Titreşim tahmininde ise R^2 değerleri yaklaşık 0,32–0,65 arasında elde edilmiştir. Kesme derinliği birincil, dönme hızı ikincil etken olarak belirlenmiştir. Son aşamada, uyarlanabilir, titreşim duyarlı bir kesme stratejisinin geliştirilmesi amacı için oluşturulan bir Kısıtlı Parçacık Sürüşü Optimizasyonu (PSO) algoritması ile uygulanabilir işlem parametreleri önerilmiştir.

Anahtar Kelimeler: Laboratuvar Ölçekli Kesici Kafa, Özgül Enerji (SE), Kazı Performansı, Makine Öğrenmesi, Parçacık Sürüşü Optimizasyonu (PSO)

PhD. Thesis

SUMMARY

CUTTING PERFORMANCE OPTIMIZATION OF LABORATORY SCALE ROTARY ROCK CUTTING HEAD BY APPLICATION OF MACHINE LEARNING METHODS

Abdolsattar ROUDINI

Karadeniz Technical University
The Graduate School of Natural and Applied Sciences
Mining Engineering Graduate Program
Supervisor: Prof. Kerim AYDINER
2025, 149 Pages

This thesis investigates the cutting performance of a laboratory-scale cutting head equipped with conical picks and develops a data-driven approach to reduce specific energy and machine-frame vibration. Unrelieved cutting was carried out under a full factorial $3 \times 3 \times 3$ design, resulting in 243 tests. Power demand, traverse speed, bi-axial vibration (X and Z), and actual depth of cut were recorded. The X direction, aligned with cutting, displayed the highest vibration, and two thresholds (≈ 86 mm/s and ≈ 141 mm/s) separated vibration behavior in line with material strength. Supervised machine learning models were trained to relate operating parameters to specific energy and vibration. Tree-based ensembles predicted specific energy with high accuracy, yielding cross-validated R^2 values of 0.93–0.97, with depth of cut identified as the most influential factor. Vibration prediction achieved R^2 values of about 0.32–0.65, with depth primary and rotational speed secondary. A constrained Particle Swarm Optimization (PSO) routine provided feasible parameter recommendations, supporting adaptive and vibration-aware selection of operating settings.

Key Words: Laboratory-Scale Cutting Head, Specific Energy (SE), Cutting Performance, Machine Learning, Particle Swarm Optimization (PSO).

LIST OF FIGURES

	Page No
Figure 1.1. Classification of Mechanical excavation machines modified after (Tatiya, 2017).....	9
Figure 1.2. Transvers roadheader- Key components and functions of a typical Mt521 Sandvik tunneling roadheader (SANDVIK Group).....	10
Figure 1.3. Axial roadheader- SANY SCR280 High-Gradient axial roadheader (SANY).....	10
Figure 1.4. The economic cuttability envelope for roadheader pick tools (Sifferlinger et al., 2017).....	11
Figure 1.5. Relation of roadheader weight against maximum operation area regarding rock UCS as stated by the manufacturers in 2015 (Sifferlinger et al., 2017).....	12
Figure 1.6. Some of the cutting parameters acting on a conical pick cutter configuration modified after (Zhou et al., 2023), where FC is the cutting force, FN is the normal force, and RC is total force.	13
Figure 1.7. Relationship between artificial intelligence, machine learning and deep learning (Arooj et al., 2022)	20
Figure 1.8. Machine-learning taxonomy mind map illustrating core paradigms and example algorithms; the diagram is illustrative rather than exhaustive.	20
Figure 2.1. Sand supplied from the Güsey Concrete Plant in Yomra, Trabzon, used in sample preparation.....	26
Figure 2.2. Portland cement type CEM I 42.5 R manufactured by the Akçakale Manufacturing Group, Trabzon, used as the binder in sample preparation	27
Figure 2.3. Wooden mold designed for specimen preparation: (a) schematic drawing with dimensions in cm; (b) fabricated mold used in the laboratory	28
Figure 2.4. Sequence of specimen casting steps	30
Figure 2.5. Equipment used for casting: (a) Vibration table, (b) Steel compaction rods, and (c) Mechanical mixer	31
Figure 2.6. Freshly cast specimens kept in their molds for 24 hours prior to demolding	31
Figure 2.7. Concrete specimens submerged in a water tank during the curing stage	32
Figure 2.8. Marble cutting experimental setup at the Mining Engineering Department of Karadeniz Technical University	33

Figure 2.9.	Modification of the marble cutting machine: (a) disassembled disc cutter housing, and (b) mounted laboratory-scale rotary cutting head with conical picks	34
Figure 2.10.	(a) Used conical pick (Schematic and real one) (b) Rotary cutting head schematic	34
Figure 2.11.	Schematic views of the machine set up (a) front, (b) back, and (c) side, indicating the locations of sensors, motors, and key structural members.....	35
Figure 2.12.	View of the full setup of rotary cutting head machine.....	36
Figure 2.13.	Actual installation of the vibration sensors on the cutting head	36
Figure 2.14.	Placement of the linear displacement transducers on the machine.....	37
Figure 2.15.	Casting, demolding, and curing of $100 \times 100 \times 100$ mm concrete cubes: (a) placement and compaction; (b) demolding after initial set; (c) water curing at 18 ± 2 °C for 7 days.....	39
Figure 2.16.	UCS failure modes of concrete cubes for three mixes: (a) 350-3, (b) 250-2, (c) 175	40
Figure 2.17.	Coring procedure and recovered specimens: (a) laboratory coring rig used to extract cylindrical; (b) obtained cores after coring, prior to trimming.....	41
Figure 2.18.	Trimmed cores prepared for modulus testing (companion cores also used for BTS).....	41
Figure 2.19.	LVDT layout for modulus test; two vertical and one horizontal on the specimen, mounted over the gauge length per ISRM	42
Figure 2.20.	Screenshot of the elasticity test interface (BESMAK/T&M): real-time plots of axial stress–axial strain (“Young Modulus1”) and lateral stress–lateral strain (“Young Modulus 2”)	43
Figure 2.21.	Brazilian (splitting) tensile test setup and representative failures for the 250-2 (left) and 350-3 (right) mixes.	44
Figure 2.22.	Relieved and unrelieved cutting modes with conical picks as a function of groove spacing s relative to depth of cut d . Modified after (Jeong et al., 2020).....	48
Figure 2.23.	Post-cut verification of the real depth of cut using a digital vernier caliper on grooves	48
Figure 2.24.	Representative specimens after rotary unrelieved cutting: (a) Mix 1, (b) Mix 2, (c) Mix 3	49
Figure 2.25.	Mix-1 complete specimen set after three rotary unrelieved cutting repeats	49
Figure 2.26.	Mix-2 complete specimen set after three rotary unrelieved cutting repeats	50
Figure 2.27.	Mix-3 complete specimen set after three rotary unrelieved cutting repeats	51

Figure 2.28.	Dual data-acquisition setup during cutting tests	52
Figure 2.29.	Exported raw data from SCADA -Test 14 Sample 1, Repeat 3.....	53
Figure 2.30.	Sound-level meter screenshot for Test 14 Sample 1, Repeat 3.....	54
Figure 2.31.	Simple linear regression showing the best-fit line, intercept (β_0), slope (β_1), and random error (ϵ_i) between observed and predicted values.	56
Figure 2.32.	Differences between OLS, Ridge, and Lasso (single predictor).....	59
Figure 2.33.	Random forest classification workflow (GeeksforGeeks, 2025).....	60
Figure 2.34.	The architecture of gradient boosting decision tree, adapted from (Deng et al., 2021)	62
Figure 2.35.	Illustration of PSO method, where each particle updates its position ($x_{it} + 1$) based on individual and global best positions and velocity (v_i) (adapted from(Akbasli, 2024)).	64
Figure 2.36.	The flowchart of PSO (adapted from (Gad, 2022))	65
Figure 3.1.	Results of BTC and UCS of three different mixes	70
Figure 3.2.	Mix 1-repeat 1: vibrations in the X and Z directions.....	73
Figure 3.3.	Mix 1-repeat 2: vibrations in the X and Z directions.....	73
Figure 3.4.	Mix 1-repeat 3: vibrations in the X and Z directions.....	74
Figure 3.5.	Vibrations in X and Z directions over time for Mix-1, repeat 3, Test 18.....	74
Figure 3.6.	Mix 2-repeat 1: vibrations in the X and Z directions.....	75
Figure 3.7.	Mix 2-repeat 2: vibrations in the X and Z directions.....	75
Figure 3.8.	Mix 2-repeat 3: vibrations in the X and Z directions.....	76
Figure 3.9.	Vibrations in X and Z directions over time for Mix 2, repeat 2 Test 18.....	76
Figure 3.10.	Mix 3-repeat 1: vibrations in the X and Z directions.....	77
Figure 3.11.	Mix 3-repeat 2: vibrations in the X and Z directions.....	77
Figure 3.12.	Mix 3-repeat 3: vibrations in the X and Z directions.....	78
Figure 3.13.	Vibrations in X and Z directions over time for Mix 3, repeat 1 Test 6.....	78
Figure 3.14.	Required Python libraries imported in the Google Colab environment, including data handling, preprocessing, modeling, evaluation, inspection, and visualization modules.....	80
Figure 3.15.	Mix-1 SE, model comparison	82
Figure 3.16.	Mix-1 SE, feature importance.....	82
Figure 3.17.	Predicted versus observed SE for Mix-1 using Gradient Boosting (K- fold cross validation).....	83

Figure 3.18.	Mix-2 SE, model comparison	84
Figure 3.19.	Mix-2 SE, feature importance.....	85
Figure 3.20.	Predicted versus observed SE for Mix-2 using Random Forest (K-fold cross validation).....	85
Figure 3.21.	Mix-3 SE, model comparison	87
Figure 3.22.	Mix-3 SE, feature importance.....	87
Figure 3.23.	Predicted versus observed SE for Mix-3 using Gradient Boosting (K-fold cross validation).....	88
Figure 3.24.	Mix-1 Vibration-X, model comparison	89
Figure 3.25.	Mix-1 Vibration-X, feature importance.....	90
Figure 3.26.	Mix-1 Vibration-X, predicted versus observed using Ridge CV.....	90
Figure 3.27.	Mix-2 Vibration-X, model comparison	92
Figure 3.28.	Mix-2 Vibration-X, feature importance.....	92
Figure 3.29.	Mix-2 Vibration-X, predicted versus observed using Random Forest.	93
Figure 3.30.	Mix-3 Vibration-X, model comparison	94
Figure 3.31.	Mix-3 Vibration-X, feature importance.....	95
Figure 3.32.	Mix-3 Vibration-X, predicted versus observed using Ridge CV.....	95
Figure 3.33.	Feature importance by Mix for SE.....	96
Figure 3.34.	Feature importance by Mix for Vibration-X.....	97
Figure 3.35.	Best ML model by mix and target	97
Figure 3.36.	Initialization cell of the Google Colab notebook that sets up the end-to-end optimization pipeline	100
Figure 3.37.	Vib-X distributions with learned thresholds t1 and t2, supporting online class detection and set-point selection.	103
Figure 3.38.	Three-dimensional surface maps of the tungsten-carbide tips of Picks 1–3 before and after excavation.	104
Figure 3.39.	Visual condition of Picks 1–3 after cutting tests.	105

LIST OF TABLES

	Page No
Table1.1. Some important roadheader performance prediction models	15
Table1.2. Machine learning and control methods applied in roadheader automation, with their main uses and supporting references.	22
Table 2.1. Chemical composition of CEM I 42.5 R cement (average Jan–Sep 2021) versus TS EN 197-1 requirements.	27
Table 2.2. Physical properties of CEM I 42.5 R cement (average Jan–Sep 2021)	28
Table 2.3. Mix proportions of cement, water, and sand used for preparing concrete specimens	29
Table 2.4. Upgraded equipment and technical specifications, including component models, operating ranges, sampling rates, accuracy, and mounting locations.	38
Table 2.5. Full factorial design with three variables: cutting depth, cutting head rotational speed, and table speed (27 total tests)	46
Table 2.6. Key technical specifications of the CEM DT-8852 sound-level meter.	53
Table 2.7. Peak metrics within the identified cutting interval for Test 14 (Sample 1, Rep3)	55
Table 3.1. UCS results from cubic concrete specimens	69
Table 3.2. BTS results from all 3 mixes	70
Table 3.3. Modulus of elasticity (10-60%) results from core samples	71
Table 3.4. Vibration-X summary for Mix-1 to Mix-3, with Z included for comparison. The X axis is aligned with the cutting direction.	72
Table 3.5. Mix-1 SE, model performance	81
Table 3.6. Mix-1 SE, permutation feature importance	81
Table 3.7. Mix-2 SE, model performance	84
Table 3.8. Mix-2 SE, permutation feature importance	84
Table 3.9. Mix-3 SE, model performance	86
Table 3.10. Mix-3 SE, permutation feature importance	86
Table 3.11. Mix-1 Vibration-X, model performance	89
Table 3.12. Mix-1 Vibration-X permutation feature importance.	89
Table 3.13. Mix-2 Vibration-X, model performance.	91
Table 3.14. Mix-2 Vibration-X permutation feature importance	91
Table 3.15. Mix-3 Vibration-X, model performance	94

Table 3.16.	Mix-3 Vibration-X permutation feature importance	94
Table 3.17.	The performance comparison in best model by mix and target	98
Table 3.18.	Trade-off sweep results from joint PSO for Mix-1 to Mix-3	102
Table 3.19.	Mass loss of Picks 1–3 after excavation.	104



LIST OF SYMBOLS AND ABBREVIATIONS

Abbreviation	Expansion	Notes
A*	A-star	Graph search algorithm
AI	Artificial Intelligence	
ANN	Artificial Neural Network	
ASTM	American Society for Testing and Materials	e.g., ASTM C109/C109M
BP	Backpropagation	
BPNN	Backpropagation Neural Network	
BTS	Brazilian Tensile Strength	Also called ITS
C-weighting	C-frequency weighting	Acoustic measurement weighting
CAI	Cerchar Abrasivity Index	
CEM I 42.5 R	Portland cement strength class 42.5 R	Per TS EN 197-1
CNN	Convolutional Neural Network	
CPU	Central Processing Unit	
D&B	Drill and Blast	
dBA	A-weighted sound pressure level	Decibels A-weighted
dBC	C-weighted sound pressure level	Decibels C-weighted
DBN	Deep Belief Network	
DE	Differential Evolution	
DL	Deep Learning	
DQN	Deep Q Network	
GA	Genetic Algorithm	
GA-BP	Genetic Algorithm Backpropagation network	
Gini	Gini impurity	Decision tree split metric
GSI	Geological Strength Index	
ICR	Instantaneous Cutting Rate	
IR	Insoluble Residue	

Abbreviation	Expansion	Notes
ISAPSO	Improved Simulated Annealing Particle Swarm Optimization	
ISRM	International Society for Rock Mechanics and Rock Engineering	ISRM Suggested Methods
ITAE	Integral of Time multiplied by Absolute Error	
ITS	Indirect Tensile Strength	Brazilian tensile strength
KTU	Karadeniz Technical University	
LDA	Linear Discriminant Analysis	
LOI	Loss on Ignition	
LSSVM	Least Squares Support Vector Machine	
LVDT	Linear Variable Differential Transformer	Displacement transducer
MAE	Mean Absolute Error	Model evaluation metric
ML	Machine Learning	
MLP	Multi-Layer Perceptron	
MMR	Material Removal Rate	Same physical quantity as ICR
MSE	Mean Squared Error	Model evaluation metric
NX	54.7 mm core size designation	Geotechnical core size
OEM	Original Equipment Manufacturer	
PCA	Principal Component Analysis	
PI	Proportional Integral	
PID	Proportional Integral Derivative	
PLC	Programmable Logic Controller	
PNN	Probabilistic Neural Network	
PSO	Particle Swarm Optimization	
PSO-BP	Particle Swarm Optimization Backpropagation	

Abbreviation	Expansion	Notes
PSO-VMD-LSSVM	Pipeline combining PSO, VMD, and LSSVM	
q95	95th percentile	
QR	QR decomposition	Orthogonal upper triangular factorization
RBF NN	Radial Basis Function Neural Network	
RF	Random Forest	
RH	Roadheader	
RL	Reinforcement Learning	
RMBI	Rock Mass Brittleness Index	
RMCC	Rock Mass Cuttability Classification	
RMCI	Rock Mass Cuttability Index	
RPI	Rock Penetration Index	
RPM	Revolutions per minute	
RQD	Rock Quality Designation	
RS-232	Recommended Standard 232	Serial interface
RS-485	Recommended Standard 485	Differential serial interface
R ²	Coefficient of determination	Model goodness of fit
SCADA	Supervisory Control and Data Acquisition	
SE	Specific Energy	
SOFM	Self-Organizing Feature Map	
SPL	Sound Pressure Level	
SSD	Solid State Drive	
SVM	Support Vector Machine	
SVR	Support Vector Regression	
TBM	Tunnel Boring Machine	
TFT	Thin Film Transistor	Display technology
TS EN 197-1	Turkish and European Standard 197-1	Cement standard

Abbreviation	Expansion	Notes
TÜBİTAK	The Scientific and Technological Research Council of Türkiye	
UCS	Uniaxial Compressive Strength	
UKF	Unscented Kalman Filter	
VMD	Variational Mode Decomposition	
VSAPSO-BP	Variable Step-size Adaptive PSO Backpropagation	



Symbol	Meaning	Units or Notes
$\hat{z}$	Unit vector along rotational axis	dimensionless
c_1, c_2	Acceleration coefficients in PSO	dimensionless
d	Depth of cut	mm
D	Disc or core diameter	mm
d_{real}	Measured (actual) depth of cut	mm
E	Secant modulus of elasticity	GPa
F_C	Cutting (tangential) force at pick	N
F_N	Normal force at pick	N
F_S	Side force at pick	N
g	Standard gravity	$9.80665 \frac{m}{s^2}$
g_best	Global best position in the swarm	
K	Energy transfer ratio	dimensionless
MAE	Mean absolute error	Model evaluation metric
N (unit)	Newton, unit of force	N
N (variable)	Rotational speed (as used in $\omega = 2\pi N$)	$\frac{rev}{s}$
P	Power	W
p_best	Personal best position of a particle	
P_max	Peak load in Brazilian test	N
r_1, r_2	Uniform random numbers in [0,1]	
r_i	Pick i position vector radius	m

Symbol	Meaning	Units or Notes
R_i	Radial distance of pick i	m
Rep	Repetition index	Rep 1, Rep 2, Rep 3
R^2	Coefficient of determination	Model goodness of fit
S	Groove spacing	mm
s/d	Spacing to depth ratio	dimensionless
SE	Specific energy	$\frac{MJ}{m^3}$ or $\frac{kWh}{m^3}$
T	Torque on cutterhead	N m
t	Disc thickness	mm
t_1, t_2	Learned vibration thresholds	$\frac{mm}{s}$
v_i	Velocity of particle i	$\frac{mm}{s}$
v_t	Table speed (traverse speed)	$\frac{cm}{min}$
w	Inertia weight in PSO	
W_{eff}	Effective kerf width	m
x_i	Position of particle i	
α	Rake angle of pick	deg
γ	Attack angle of pick	deg
ε	Axial strain	dimensionless
$\varepsilon_1, \varepsilon_2$	Strain values used for secant modulus	dimensionless
θ	Cone angle of the pick	deg

Symbol	Meaning	Units or Notes
ρ	Density	$\frac{g}{cm^3}$
σ_1, σ_2	Principal Stress values used for secant modulus	MPa
σ_c	Uniaxial compressive strength	MPa
σ_t	Tensile strength	MPa
ω	Angular speed	$\frac{rad}{s}$

1. INTRODUCTION

1.1. Background & Motivation

Boom-type roadheaders remain central to selective mechanical excavation because they deliver continuous cutting, rapid geometric adjustments, and tight profile control across civil and mining projects (Deshmukh et al., 2020). Performance assessment for mechanical excavation is typically framed around specific energy (SE) (the energy input per unit volume removed) following the classic definition in rock cutting and drilling (Teale, 1965). In practice, volumetric production (instantaneous cutting rate, ICR) is governed by operating parameters such as cutterhead rotational speed, traverse speed, and depth of cut, along with pick condition and rock properties. Empirical and semi-empirical models connect ICR to cutting power and an “optimum” SE derived from tests, providing a practical bridge between energy demand and achievable advance (Ebrahimabadi et al., 2011a). Complementary work in metallic ores and mixed conditions further refines prediction using rock indices and test-based parameters (Çomaklı et al., 2014). Despite this foundation, real projects face strong variability in rock mass conditions, tool wear, and operational constraints. These factors make fixed setpoints suboptimal and can push SE upward while depressing production. The interaction between the machine and the rock mass strongly influences performance. Reviews and practice reports show that cutterhead design, pick spacing, and attack angle act together with rock properties such as UCS, abrasivity, texture, and discontinuities to govern cutting response (Deshmukh et al., 2020; Restner & Plinninger, 2015). Meanwhile, the equipment landscape is moving toward sensing-rich platforms and semi-autonomous functions, creating an opportunity to close the loop between data streams, prediction, and parameter selection (Peinsitt et al., 2020).

Contractors seek to optimize SE and tool usage while achieving production objectives and adhering to power, vibration, and profile limitations. Conventional correlations or single-equation predictors are inadequate in real-time adaptation to variations in lithology, jointing, water content, or picking circumstances. This establishes a data-driven framework that (i) models the nonlinear relationships between controllable parameters (rotational speed, traverse speed, depth of cut) and performance metrics (ICR, SE, power, vibration), (ii) quantifies feature significance and uncertainty for interpretability and robustness, and (iii)

conducts multi-objective optimization to propose setpoints that balance production, energy, and tool longevity (Deshmukh et al., 2020; Ebrahimabadi et al., 2011a; Teale, 1965).

Practically, vibration signals captured on the head/boom or machine frame offer a low-intrusion proxy for cutting conditions, rock hardness surrogates, and tool–rock interaction states. Embedding vibration-derived features into predictive models can provide the real-time state estimation needed for adaptive control, suggested in recent automation-oriented tunneling developments but not yet standardized for roadheader cutting optimization (Deshmukh et al., 2020; Peinsitt et al., 2020). A lab-scale, instrumented rotary cutting head enables controlled experiments that generate synchronized datasets of power, depth, and vibration across a variety of operating conditions and materials. Such datasets are well suited for comparing regression/ensemble/kernel methods, establishing reliable SE/ICR predictors, ranking influential variables, and validating optimization policies under constraints. The overarching goal is a reproducible, generalizable machine learning (ML) pipeline that reduces SE at required throughput and can transfer to field-scale operation through careful scaling, feature engineering, and domain-shift assessment.

1.1.1. Tunneling And Mechanical Excavation: Role of Roadheaders.

Roadheaders fill the space between drill-and-blast and full-face tunnel boring machines (TBM). They cut continuously, steer easily, and let crews shape non-circular profiles. That makes them useful for drifts, caverns, cross-passages, and urban work where vibration and overbreak must be kept low (Deshmukh et al., 2020).

Their main strengths are flexibility and selective excavation. Operators can adjust boom position, cutterhead speed, and depth of cut as geology changes. In weak to moderately strong rock like toffs, marls, stratified limestones, field data show stable advance, good profile control, and manageable energy demand (Comakli, 2019). In inclined or mixed zones, performance depends on more than UCS: water ingress, lamination, and tunnel geometry all shift utilization and cutting rate (Bilgin et al., 2004).

Using learning-based prediction (PCA-DBN (principal component analysis-deep belief network)) alongside tuned RPM, feed, and depth of cut, planners can set realistic production ranges and define risk envelopes for hard-rock drives. These methodological developments contribute to the gradual expansion of roadheader use in harder ground conditions, although the overall applicability remains constrained (Guo et al., 2024).

Advances in cutterhead design further extend this operating window. The use of alternative tools and head geometries, including crown picks, mini-disks and lubricated holders, has been shown to mitigate wear and moderate cutting forces in formations with high UCS and abrasivity (Kotwica, 2018).

In short, roadheaders give tunneling teams a controllable, low-vibration option that can follow complex geometry and react to variable ground. Their best domain remains weak to medium-strength rock, but methodical design and data-guided operation continue to widen their use.

1.1.2. Industrial Pain Points: Productivity, Energy Use, Pick Wear, Safety

Productivity in boom-type mechanical excavation depends on how effectively operating parameters and cutterhead kinematics are matched to prevailing rock-mass conditions. Field validation of the rock-mass cuttability classification (RMCC) for roadheaders demonstrates a strong correlation with net cutting rate, indicating that geological and structural characteristics such as jointing, abrasivity and water inflow largely govern achievable advance. In combination with machine and operational factors including available power and torque, applied thrust, rotation speed and operator control, these conditions determine both the upper limit on production and the lower bound on energy demand (Dibavar et al., 2023).

On the energy side, the cutting process itself is the principal sink. Simulator and experimental studies consistently show that cutterhead configuration and the chosen motion strategy strongly influence efficiency, dynamic loads, vibration, and energy consumption (Krauze et al., 2022). In particular, introducing automatic control of cutting-head motion to adapt to workability gradients can reduce average mining energy by roughly 47–63% and lower average mining power by ≈ 30 –35%, while increasing head traverse speed and overall cutting efficiency, evidence that real-time motion planning is an actionable lever on both productivity and energy use (Cheluszka, 2021a).

Tooling remains the chronic bottleneck. Conical picks are consumables whose degradation drives downtime, power draw, and cost. Dynamic measurements on full-scale rigs show that normal forces dominate the pick load spectrum (with representative averages on the order of several KNs) and that load transients concentrate stresses at the tip and shank, precisely where fatigue and chipping initiate (Fan et al., 2023). As wear progresses, both the

mechanics of breakage and the energy signature are associated with cutting change. Volumetric wear quantification through photogrammetry reveals a rapid loss of geometry and illustrates how features in power signals can be utilized to infer the wear states during operation (Pawlik et al., 2022). From an investment standpoint, prolonged service life is not merely a materials problem but a system problem: selection, layout, cooling, and load management must be coordinated to suppress the high-frequency load cycles that accelerate wear (Cheluszka et al., 2024)

Safety intersects with all of the above. First, dust: full-scale laboratory cutting trials show that worn picks produce significantly more respirable and ultrafine particles than sharp picks, elevating operator exposure and complicating environmental control (Slouka et al., 2022). Second, global risk: roadheader operations carry hazards ranging from noise and vibration to explosion and mechanical/electrical risks; systematic assessments emphasize strict procedural discipline, continuous training, and design-stage prevention measures to mitigate these exposures (Škvareková et al., 2021). In this context, digital tools, simulators, monitoring, and advanced control have a dual role: they not only reduce energy and smooth loads but also enable crew withdrawal from the face and support safer, more consistent execution under variable geology (Krauze et al., 2022).

1.1.3. Variability Of Rock Mass Properties and Need for Adaptive Control

Rock masses change from meter to meter. Thin interbeds, hard inclusions, joints, and minor faults cause sudden shifts in stiffness and strength. On a roadheader, these shifts show up as swings in torque, motor power, and vibration. Fixed set-points can't hold the machine near an efficient regime when the ground keeps changing (M. Zhang et al., 2023). The result is simple: a static control strategy is a poor match to a non-stationary face. When density and hardness vary, the cutting motor and key structures see overload–underload cycles. Operators often work “blind” to the true stress state. That is why adaptive, geology-aware control is needed (J. Zhang et al., 2024).

Recent sensing work shows that variability can be read and acted on. Low-cost signals such as motor current, when decomposed and denoised, can identify lithology in real time. A PSO-VMD-LSSVM (Particle Swarm Optimization-variational modal decomposition-least-squares support vector machine) pipeline raised recognition to about 97.5% using only current data from a tunneling rig (Qi et al., 2022). Multi-point vibration fused into time–

frequency images can also track coal–rock states under interbedded conditions, reaching about 98% accuracy in lab-field setups (M. Zhang et al., 2023). But sensing alone is not enough. The controller must close the loop fast and with the right target. A digital-twin stack that links “shape–performance–control” now updates stress predictions about every 0.34 s and adjusts swing speed in ~ 0.3 s to keep stresses within limits. This limits overload when the face composition flips, and it provides an explicit basis for adaptive set-points (J. Zhang et al., 2024).

Geometry also matters. If pick holders are not where the model assumes, sensing and control maps will drift. A stereovision method now measures pick-holder positions during robotic assembly with sub-millimetre spread. That gives reliable geometry for model calibration and closed-loop targets (Cheluszka & Jagieła-Zajac, 2022).

Finally, components age under variable regimes, which can mask geological effects. Bearing faults in the cutting arm are weak and often buried in noise. A variational mode decomposition (VMD)-based, domain-adaptive convolution neural networks (CNN) holds diagnosis accuracy across changing operating conditions. This is key if the same sensors must serve both health monitoring and adaptive control (Qu & Zhang, 2023).

1.1.4. Digitalization in mining:

Over the past decade, underground excavation has taken gradual steps toward tighter integration of sensing and control, often described under the Mining 4.0 umbrella. In practice, adoption is uneven across sites and commodities, and operator-assist functions, tele-remote capability, and telemetry are more common than full autonomy. Digital twins and simulation environments are considered safe, repeatable test beds for tuning control strategies on high-inertia equipment such as roadheaders, helping researchers probe stability limits, energy use, and operator-assist logic before pilot deployment (Cheluszka et al., 2022).

From a mechanic's perspective, specific energy (SE) remains a compact performance metric linking depth of cut, force, and chip formation; recent analyses with conical picks show how SE varies systematically with material toughness and process parameters (X. Wang & Su, 2019).

Major OEMs have steered their product roadmaps in line with this shift, but with a clear emphasis on operator assist, geometric compliance, and connectivity rather than full autonomy. Current Sandvik MT/MR platforms publicly document semi-automated cutting

via Cutronic®, optional profile guidance for line and grade, tele-remote operation, and fleet telemetry; the GEODATA Tauros RH guidance package is integrated at the machine-control layer, underscoring the priority placed on profile accuracy and supervisory control. SANY's STR/EBZ lines highlight remote control, on-board diagnostics, and “no-blind-spot” vision features, again signaling assisted operation as the productized baseline. Across vendors, what is widely available today is a combination of guidance, semi-automated cut routines, and connectivity for monitoring and maintenance; closed-loop, energy-optimal scheduling of rotational speed, traverse speed, and depth of cut is not claimed as a commercial feature.

1.2. Problem Context & Statement

1.2.1. Scientific Gap

Despite a growing literature on performance prediction, there is limited consensus on how process variables (rotational speed, traverse speed, and cutting depth), material properties (e.g., UCS), and in-process signals jointly determine cutting depth per revolution, power demand, vibration response, and specific energy (SE). The situation is complicated by (I) directionality of vibration during cutting, (II) nonlinear interactions among set-points, and (III) material dependence (e.g., across three UCS levels). This research contributes:

1. A procedure to identify the dominant vibration direction during cutting of concrete samples.
2. Data-driven vibration thresholds that separate three UCS classes; and
3. Supervised learning models that approximate the mapping from process variables to performance metrics (including specific energy), enabling parameter optimization.

1.2.2. Limitations of Empirical and Physics-Only Approaches

Classical empirical relations (e.g., SE vs. depth) and physics-only models provide useful trends but often under-represent dynamic effects, tool state (pick wear), and cross-axis vibration coupling. They also struggle with online adaptation when material or boundary conditions change. Consequently, they are difficult to deploy as real-time controllers without additional data-driven corrections. A hybrid framework that integrates vibration-based

inference with learned process-to-performance models aligns more closely with the control objective of minimizing specific energy while satisfying stability and geometric constraints.

1.3. Research Questions & Hypotheses

1.3.1. Research Questions

RQ1. Which sensor axis (x, y, or z in the machine frame) consistently records the largest vibration response during cutting?

RQ2. Can vibration features (especially along the dominant axis) act as reliable proxies for material hardness and SE, enabling threshold-based separation of three UCS levels?

RQ3. How do rotational speed (ω), traverse speed, and cutting depth (d) jointly govern power, vibration, and SE on the laboratory rotary cutting head?

RQ4. Which machine learning (ML) models and feature sets best predict performance (depth per rev, power, vibration, SE) and support parameter optimization, and how robust are they across materials (three UCS levels) and operating ranges?

1.3.2. Hypotheses Aligned with Each Research Question.

H1: Dominant vibration direction

H1a. During cutting, the x sensor axis (cutting direction) in the machine frame exhibits a significantly higher vibration level than the other axes.

H1b. The identified dominant axis remains stable within all three UCS classes.

H2: Vibration features serve as indicators for material hardness and specific energy.

H2a. Directional vibration features (computed along the x-axis) show a stronger association with UCS class and specific energy.

H2b. A vibration-based thresholding scheme can separate three UCS levels with practically useful discrimination.

H3: Process variable effects on performance and SE

H3a. Within the mechanically stable range, increasing cutting depth (d) reduces SE by improving chip formation efficiency, up to a limit beyond which vibration and power rise disproportionately.

H3b. At a fixed depth, higher traverse speed reduces SE per unit volume only while engagement remains stable; excessive traverse induces vibration growth and power spikes.

H4: Predictive models and optimization robustness

H4a. Nonlinear ML models (e.g., gradient-boosted trees, random forests, and kernel SVR) trained on plus directional vibration features outperform linear baselines for predicting depth, power, vibration indices, and SE.

H4b. Dominant-axis selection and feature normalization improve predictive accuracy and numerical stability relative to using all axes indiscriminately.

H4c. Models trained on mixed UCS data maintain acceptable error on held-out operating ranges and materials; when performance degrades, UCS-conditioned or lightly domain-adapted models restore accuracy.

1.4. Mechanical Excavation

Since the mid-twentieth century, underground works have progressively shifted from intermittent drill-and-blast (D&B) cycles to continuous mechanical excavation. Relative to D&B, mechanized systems (I) deliver steadier advance with fewer production interruptions, (II) achieve superior profile control with reduced overbreak and lining backfill, and (III) can minimize vibration, noise, and disruption in urban settings, especially when face support and muck transport are integrated into the excavation train. These advantages are well documented for TBMs in both soft ground and rock and apply more broadly to partial-face machines when the formation is within their economic range (Bilgin et al., 2013). Figure 1.1 summarizes today's mechanical excavation categories, full-face TBMs and partial-face systems (roadheaders, continuous miners, longwall shearers).

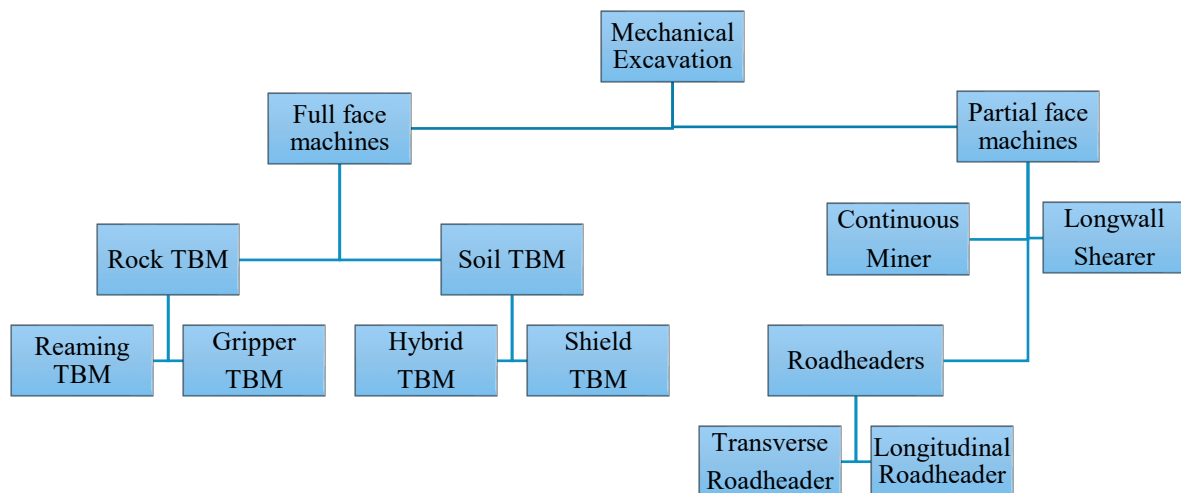


Figure 1.1. Classification of Mechanical excavation machines modified after (Tatiya, 2017)

1.5. Roadheader Types and Applications

Roadheaders are commonly classified by cutterhead orientation:

- Transverse-head machines (most common in tunnelling): two drums on a horizontal shaft; wide cutting envelope and good profile control.
- Axial-head (longitudinal) machines: a single drum on the boom axis; favored in some mining geometries.

Axial-head roadheaders are generally effective in non-abrasive, massive rock with $UCS \approx 60\text{--}80\text{ MPa}$; in jointed ground this may extend to $\sim 80\text{--}100\text{ MPa}$. Transverse roadheaders can cut non- to moderately abrasive massive formations up to $\sim 100\text{--}120\text{ MPa}$ UCS, rising to $\sim 160\text{--}180\text{ MPa}$ where discontinuities assist cutting (Bilgin et al., 2013).

Figure 1.2 shows the Sandvik MT721 transverse roadheader with dual drums on a horizontal shaft and its components and functions; figure 1.3 shows the SANY SCR280 axial (longitudinal) roadheader with a single drum aligned to the boom axis which is applicable in non-coal such as potassium/sodium salt ores.

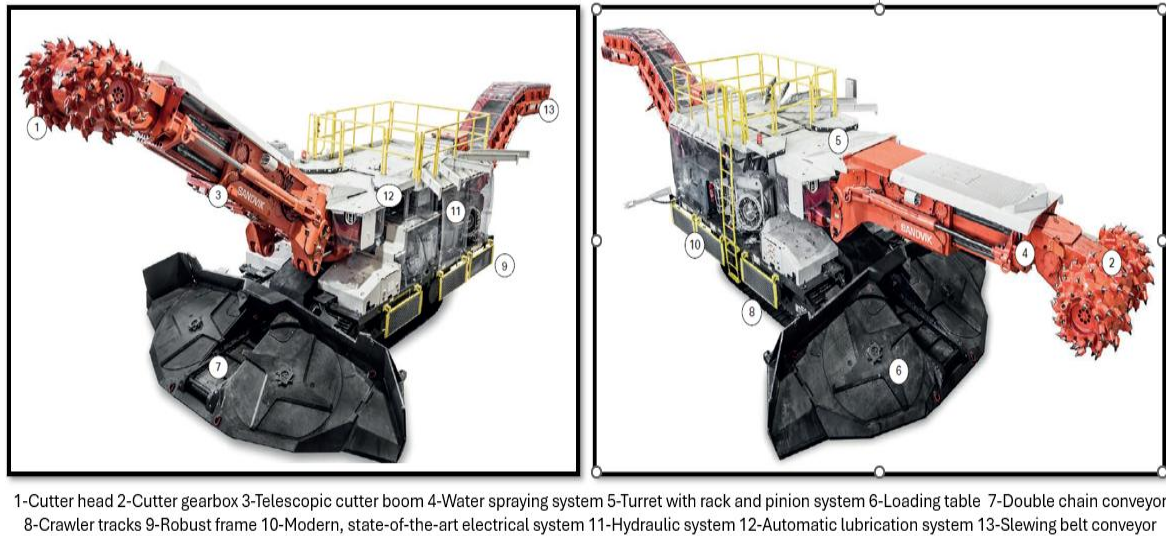


Figure 1.2. Transvers roadheader- Key components and functions of a typical Mt521 Sandvik tunneling roadheader (SANDVIK Group)



Figure 1.3. Axial roadheader- SANY SCR280 High-Gradient axial roadheader (SANY)

Successful applications depend on installed power/torque, cutterhead layout, tool wear, and geologic controls (UCS, Cerchar Abrasivity Index (CAI), structure, water). Field guidance consistently shows that poor machine–formation matching leads to excessive pick consumption and schedule/cost overruns; therefore, feasibility screening (UCS–CAI),

prediction of instantaneous cutting rate (ICR) and specific energy (SE), and, where feasible, linear or pilot rotary cutting tests are recommended prior to commitment (Bilgin et al., 2006).

Drag (point-attack) conical picks mounted in rotational holders are the default tooling on roadheaders; roller (disc) cutters are characteristic of TBMs. Compared with discs, drag tools generally remove a unit volume at lower specific energy (SE) in weak–moderate rocks, but they wear rapidly as abrasivity (CAI) and strength increase, hence the familiar UCS–CAI applicability map for roadheader picks (Bilgin et al., 2013). In hard and abrasive rock, the limited-service life of conical picks is the primary constraint on roadheader applicability (Fig 1.4); performance is further governed by cutterhead geometry and overall machine mass (Fig 1.5) (Sifferlinger et al., 2017).

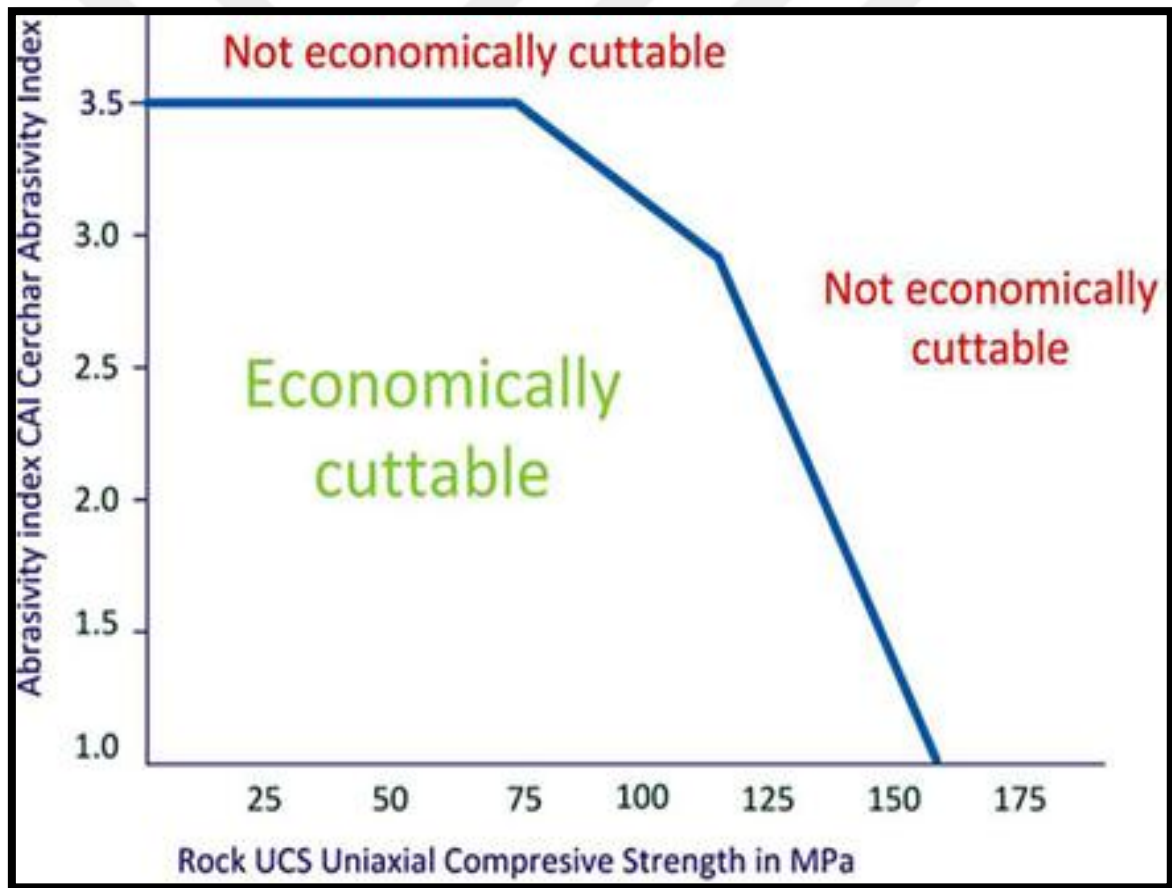


Figure 1.4. The economic cuttability envelope for roadheader pick tools (Sifferlinger et al., 2017)

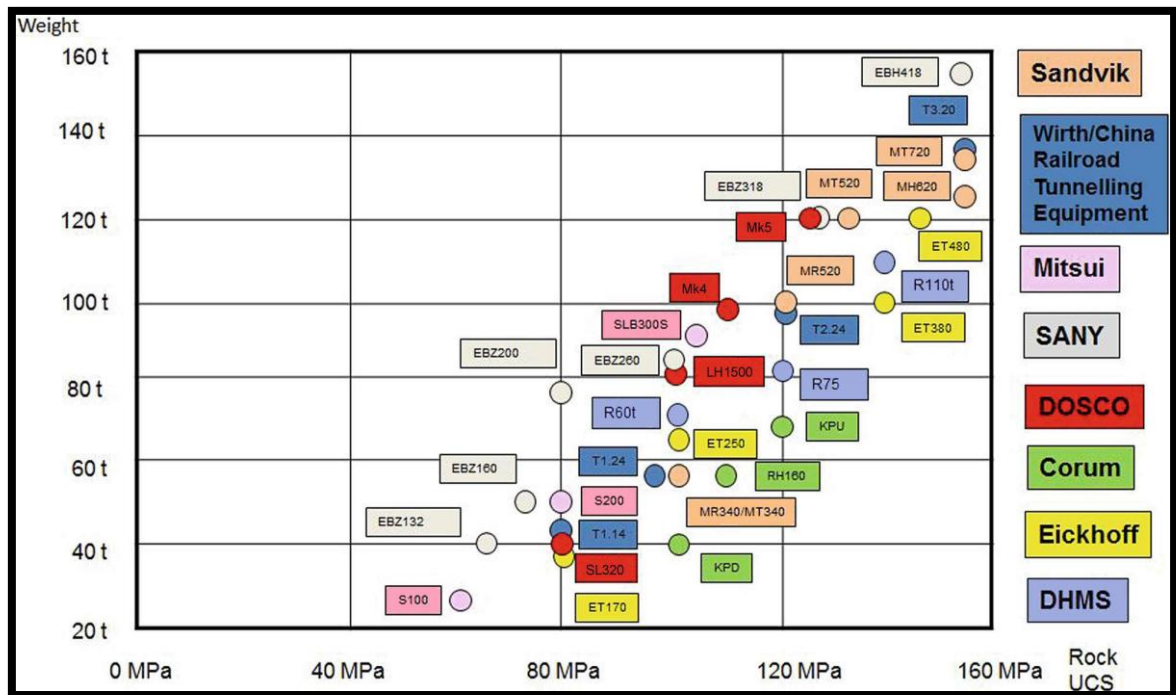


Figure 1.5. Relation of roadheader weight against maximum operation area regarding rock UCS as stated by the manufacturers in 2015 (Sifferlinger et al., 2017)

1.6. Conical Picks and Excavation Principles

1.6.1. Most Important Parameters of Conical Picks

During the cutting process, the tool–rock interaction is commonly resolved into three orthogonal components at the pick tip: drag (cutting) force (F_c) tangential to the tool path, normal force (F_N) acting roughly into the face, and a smaller side force (F_s) perpendicular to the cutting plane (arising from asymmetries and lacing) (Bilgin et al., 2013). Tool orientation (attack, rake and clearance angles) governs the partitioning among and thereby affects chipping, frictional rubbing, and wear.

Some cutting parameters affecting the performance of conical picks are shown in Fig. 1.6. According to the geometrical relationship among the relevant angles, the relationships can be expressed as: $\alpha = \pi/2 - \theta/2 - \gamma$, where α , θ and γ are the rake angle, cone angle of the pick and attack angle, respectively (X. Wang et al., 2018).

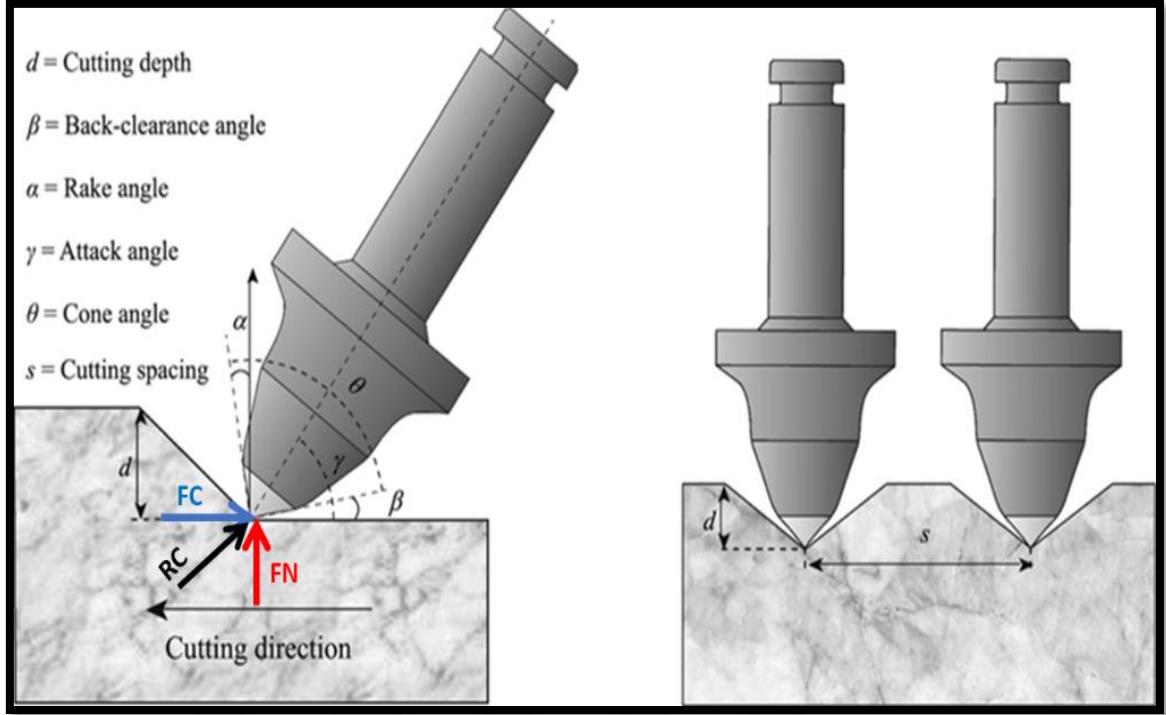


Figure 1.6. Some of the cutting parameters acting on a conical pick cutter configuration modified after (Zhou et al., 2023), where FC is the cutting force, FN is the normal force, and RC is total force.

1.6.2. Torque And Power on a Cutterhead.

On a rotating head, the distributed tangential components generate torque:

$$T = \sum_i (r_i \times F_i) \cdot \hat{z} \approx \sum_i R_i F_{c,i} \quad [N \cdot m] \quad (1)$$

Where R_i is the radial location of Pick i . The instance of machine power is:

$$P = T\omega \quad [W] \quad (2)$$

Where ω is rotational speed in rad/s; $\omega = 2\pi N$ with N rev/s

With electrical measurements corrected for drivetrain efficiency when used as a proxy. For steady operation, specific energy is:

$$SE \approx \frac{\bar{P}}{MMR} \quad [MJm^{-3} \text{ or } kWm^{-3}] \quad (3)$$

Where MMR is volume of material removed per unit time.

1.7. Roadheader Performance Prediction Models

Appropriate selection of roadheaders and their operating criteria remains pivotal to project success. Failure to match machine capability to the geological formation can result in substantial material and financial losses. Compared with other excavation systems, roadheaders must therefore be chosen in accordance with rock mass conditions, and achievable excavation rates should be predicted in advance. During cutterhead design, detailed knowledge of the target formation is essential so that the cutting layout and tool selection can be tailored to the ground (Hekimoglu & Fowell, 1991).

A central determinant of mechanized mining economics is the ability to predict cutting performance across the range of excavation conditions in which the machine will operate. The instantaneous cutting rate (ICR) is widely used for benchmarking and can be adapted for specific applications, whereas net advance rates over time may conflate cutting with auxiliary delays and thus do not isolate cutting performance (Deshmukh et al., 2020). Specific Energy (SE), the energy required to remove a unit volume of rock (typically reported as kWh/m³ or J/m³), provides an additional, mechanistically meaningful predictor of cutting efficiency, with lower SE indicating more efficient cutting. Because direct determination of SE from laboratory or full-scale cutting tests can be demanding and costly, many studies have related SE to rock properties to develop empirical estimation equations. SE is, accordingly, a key covariate in predicting ICR. Table 1.1 summarizes representative models used to predict roadheader performance.

Table1.1. Some important roadheader performance prediction models

Author(s)	Model	Equation number
	$ICR = \frac{719}{\sigma_c^{0.78}}$	(4)
Gehring (1989)	$ICR = \frac{1739}{\sigma_c^{1.13}}$	(5)
	$ICR = 0.28p(0.974)^{RMCI}$	(6)
Bilgin et al. (1991)	$RMCI = \sigma_c \left(\frac{RQD}{100} \right)^{\left(\frac{2}{3} \right)}$	(7)
Rostami et al. (1994)	$ICR = K \frac{P}{SE}$	(8)
Çopur et al. (1998)	$ICR = 27.511 \times e^{0.0023 \times RPI}$	(9)
Thuro & Plinninger (1999)	$ICR = 75.7 - 14.3 \ln \sigma_c$	(10)
	132 KW RH	
	For d=5mm, $ICR = 0.8 \frac{P}{0.37 UCS^{0.86}}$	(12)
Balci et al. (2004)	For d=9 mm, $ICR = 0.8 \frac{P}{0.41 UCS^{0.67}}$	(13)
Keleş (2005)	$ICR = 163.93 \sigma_c^{-0.5737}$	(14)
	$ICR = 5.56 RMBI + 0.6\alpha - 8.17$	(15)
Ebrahimabadi et al. (2011b)	$ICR = -0.18 SE^3 + 28.57SE - 92.82$	(16)
Ebrahimabadi et al. (2012)	$ICR = 35.22 e^{-0.54 \log RMBI }$	(17)
Çomaklı et al. (2014)	$ICR = k \frac{P}{SE}$	(18)
Kahraman and Kahraman (2016)	$ICR = -0.88\sigma_t - 0.54n + 25.01$	(19)
Choudhary et al. (2017)	$ICR = -16573X + 32.121$	(20)

RH: roadheader, ICR: Instantaneous Cutting Rate, SE: Specific Energy, RQD: Rock Quality Designation, RMCI: Rock Mass Cuttability Index, RMBI: Rock Mass Brittleness Index, UCS: Uniaxial Compressive Strength, α : Angle between tunnel axis and the planes of weakness in degrees. σ_c : Compressive Strength, RPI: Rock Penetration Index, P: Installed power in KW, K: Energy Transfer Ratio, σ_t : Tensile Strength, n: porosity (%) X: Bit wear rate (mm/m)

1.8. Automation in Roadheaders

Cutting the face in tunnels and headings is the core function of roadheaders, yet it remains energy intensive, especially in hard rock, and dynamically complex due to interacting mechanical and geological factors. Improving energy efficiency while maintaining cutting stability is therefore central to project economics (Deshmukh et al., 2020; Dolipski et al., 2015). Recent work shows the field moving from single-loop speed regulation toward coordinated sensing, control, and motion planning that target energy use, stability, and geometric accuracy in tandem. OEM practice reflects this shift, with semi-industrial trials on Sandvik's MT720 and laboratory campaigns on FAMUR's R-130 providing reference platforms for evaluation (Cheluszka et al., 2019; Heyduk & Joostberens, 2020).

1.8.1. Cutting Speed and Energy Management

A consistent result across studies is that cutterhead speed should adapt to rock conditions rather than remain fixed. Adjusting angular speed to match hardness reduces power draw and pick wear, and automatic speed control with PID/PI structures has demonstrated energy savings in simulations and tests (Dolipski et al., 2015; Heyduk & Joostberens, 2020). On the R-130 platform, a cascaded architecture with a torque-driven reference generator, a PI (Proportional Integral) speed loop tuned with the ITAE (Integral of Time multiplied by an Absolute value of Error), and fast current protection stabilized speed under variable torque and cut overload time, an arrangement proposed for next-generation mechatronic stacks (Heyduk & Joostberens, 2020). Vision-enabled schemes extend this idea by estimating Geological Strength Index from camera imagery and mapping it to discrete speed bands; in lab trials this yielded an average 5.4% reduction in drive energy, indicating that inverter-equipped machines can move from constant speed to adaptive regimes that track geology in real time (Dong et al., 2022). Building on prior computer investigations, optimization of energy consumption and dynamic behavior is frequently framed around three controllable parameters: cutting-head angular speed, boom swing speed, and cut height (Cheluszka, 2021b).

1.8.2. Control Strategies and Implementation Pathways

Multiple institutions, particularly in Poland and China, have proposed automatic control strategies that program changes in selected parameters during the cut, typically angular speed and boom swing speed, implemented via hydraulic gearing or frequency converters (Cheluszka, 2021b; Jasiulek & Świder, 2013; Joostberens & Heyduk, 2018; H. Wang et al., 2017; P. Wang, Shen, Li, et al., 2021a). Model-based and intelligent control variants are also reported: for example, a swing-speed controller that identifies cutting load using an improved particle-swarm-optimized backpropagation network and applies a fuzzy controller tuned with simulated-annealing particle swarm optimization achieved higher accuracy and faster response in simulation (P. Wang, Shen, Ji, et al., 2021a). Laboratory testing shows that proposed speed-control systems meet design aims but benefit from more robust control structures, better dynamic sensors, and attention to practical constraints such as inverter input rate limiters; coordinated closed-loop control of horizontal and vertical angular displacement is recommended because these directly affect depth of cut and load-torque variation (Cheluszka & Jagieła-Zajac, 2022). Most commercial machines still ship with constant-speed heads, but the literature argues for speed scheduling linked to rock hardness to reduce power use and tool wear (Heyduk & Joostberens, 2020).

1.8.3. Sensing, Measurement, and Environmental Limits

Progress depends on reliable sensing. High-speed optical methods and photogrammetry can capture fast phenomena and deliver time histories of displacement, velocity, and acceleration in 3D, but underground performance is constrained by air transparency, temperature, humidity, and lighting (Cheluszka & Mann, 2019). These practical limits motivate hybrid sensing stacks that combine vision with drive-side measurements and vibration signals to keep controllers informed under variable conditions (Cheluszka & Mann, 2019; Dong et al., 2022)

1.8.4. Precision Cutting, Dynamics, and Pose Compensation

Holding line and grade requires more than speed control; loads, floor irregularities, and body pose drift must be modeled and compensated. Accurate vibration modeling, including rigid–flexible coupling of substructures, is foundational for hard-rock designs and

for subsequent automation (Z. Zhang et al., 2022a). Reduced-order active disturbance rejection control at the boom has been shown to cancel pitch and yaw deviations, yielding centimeter-level cross-section tolerances compatible with roadway quality requirements (Deyi et al., 2024).

1.8.5. Path Planning and Autonomous Motion

Automation also encompasses how the machine plans and sequences cuts. For large-section headings, planners combine roadway stability analysis with kinematics-aware cutting sequences; an industrial trial at Wangjialing’s 12307 belt roadway validated a reverse-loop secondary-cutting plan, reporting stable behavior and good section forming (Wu et al., 2023). At the chassis level, safe relocation in confined headings can be posed as collision-aware path planning. An improved A* with collision prediction and Bézier smoothing shorten paths while preserving clearance by embedding the roadheader’s non-convex geometry into the heuristic, well suited to moving between staged cutting points in narrow tunnels (C. Zhang et al., 2025).

1.9. Artificial Intelligence and Machine Learning

Artificial intelligence (AI) is the broad goal of building systems that perceive their environment, reason about what to do, act, and then improve from feedback. In practice, most of today’s progress comes from machine learning (ML), algorithms that learn patterns from data rather than being explicitly programmed. ML now powers everything from recommendation engines and speech interfaces to autonomous equipment and industrial robotics, largely because it scales as data and computing grow (Jordan & Mitchell, 2015).

Within ML, it helps to separate learning strategies by the kind of feedback available. In supervised learning, models learn a mapping from inputs to predefined targets (for example, predicting rock strength from sensor features). Unsupervised learning uncovers structure in unlabeled data, such as clustering or dimensionality reduction. Reinforcement learning (RL) takes a different route: an agent interacts with an environment, receiving rewards and penalties, and learns policies that maximize long-term return. These families are complementary and almost combined in real systems (Arulkumaran et al., 2017; Sarker, 2021).

Deep learning (DL) is a subset of ML and denotes multilayer neural networks that learn hierarchical features from raw, normally unstructured data such as images, audio, and time series. The term “deep” simply refers to the number of stacked layers. As networks become deeper and datasets larger, DL has achieved state-of-the-art results in perception and language, which is why it underpins many modern AI applications (LeCun et al., 2015). Figure 1.7 presents the relationship in a simple hierarchy: AI contains machine learning, and machine learning involves deep learning.

Two recent trends seem to tighten the AI–ML link. Firstly, deep reinforcement learning couples DL’s representation power with RL’s decision making and may enable agents to learn control policies directly from high-dimensional inputs; reports suggest this approach has helped robotics and autonomous systems (Arulkumaran et al., 2017; Tang et al., 2025). Secondly, self-supervised learning can reduce the need for labeled data by creating predictive pretext tasks; in several studies these methods have matured quickly and, in some cases, approach the performance of fully supervised pretraining across modalities (Ericsson et al., 2022; Huang et al., 2023).

Put simply, AI is the ambition, ML is the toolkit, and DL is today’s most influential tool inside that kit. Where labels are scarce or decisions must be optimized by trial and error, self-supervised and reinforcement learning expand ML’s reach, bringing more of AI’s promises into deployable reality, whether in consumer products or in industrial domains like underground automation. Figure 1.8 is a mind map that organizes the major machine-learning paradigms: supervised, unsupervised, semi-supervised, reinforcement learning, deep learning, ensemble methods, and other AI algorithms, and shows representative techniques branching from each family (e.g., logistic regression, SVM, K-means, PCA, DQN, CNNs, and XGBoost).

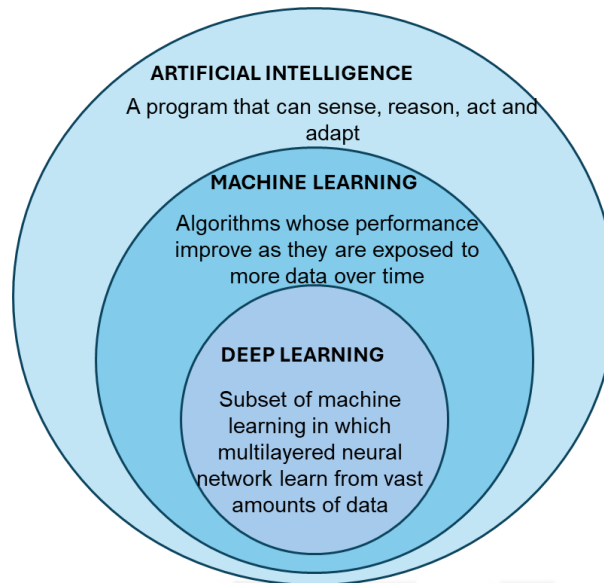


Figure 1.7. Relationship between artificial intelligence, machine learning and deep learning (Arooj et al., 2022)

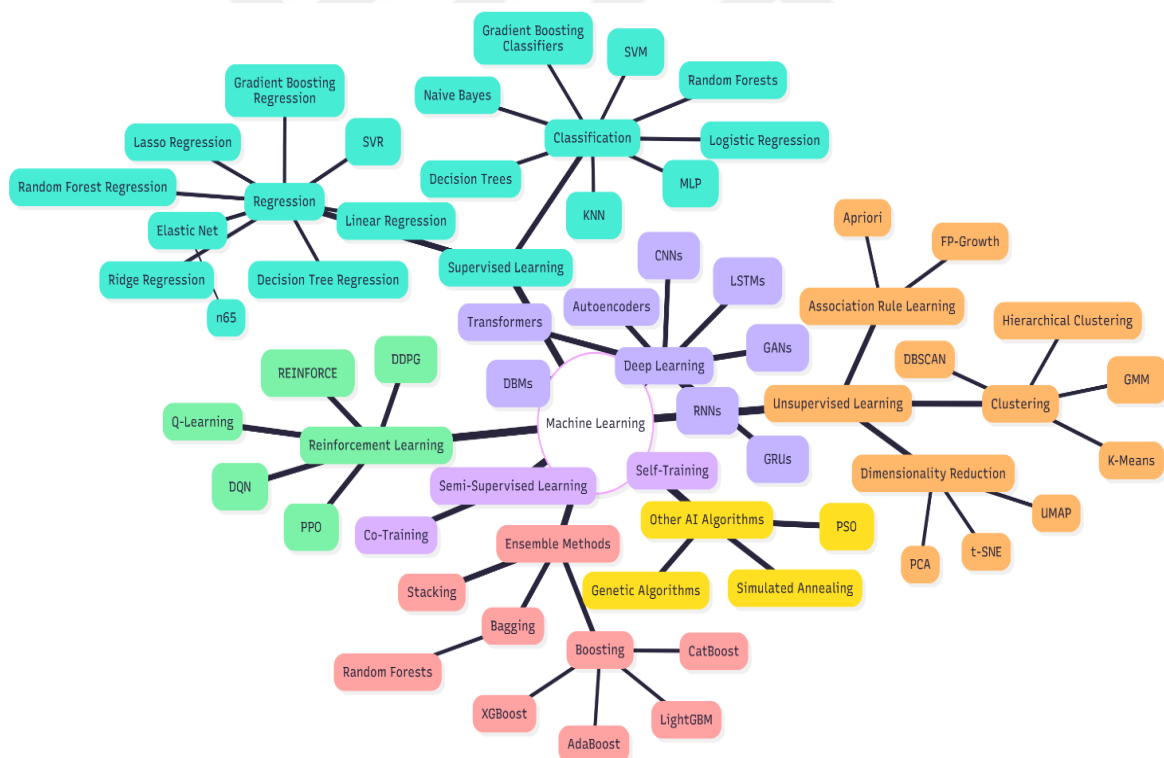


Figure 1.8. Machine-learning taxonomy mind map illustrating core paradigms and example algorithms; the diagram is illustrative rather than exhaustive.

1.10. Machine Learning Application in Roadheader Automation

Various machine learning techniques have been tested by researchers to improve roadheader automation. To keep these efforts in view, Table 1.2 pulls together examples of the main models, what they have been used for, and where they appear in literature. Some of these focus on prediction, like neural networks trained to link geological strength with cutting performance. Others lean on optimization, combining algorithms such as backpropagation with particle swarm or genetic methods to fine-tune speed and load control. A different line of work looks at fault detection, anomaly recognition, and path planning, often borrowing tools from computer science and adapting them to the underground setting.

1.10.1. Perception Of Ground Conditions and Property Recognition

Learning robust mappings from sensor streams to rock descriptors enables adaptive cutting. One line of work fits Artificial Neural Network and Backpropagation models to relate Geological Strength Index to fracture characteristics; a GA-BP variant improves accuracy and convergence, and a history-based update module broadens applicability (Dong et al., 2022). In related experiments, wavelet-packet analysis produced band-limited time–frequency representations; features derived via SVD were fused with LDA and then passed to a DE– and QR-optimized PNN to classify rock hardness during cutting (L. Zhang et al., 2020).

1.10.2. Cutting-Load Identification and Adaptive Speed Control

Stable operation depends on identifying the cutting load and adjusting swing or cutterhead speed in near real time. A BP controller, optimized with an improved PSO, was used for cutting-load identification and coal–gangue discrimination, with special-condition logic implemented in Automation Studio (P. Wang, Yang, et al., 2021). A multiparameter strategy that combined an optimized BP network with a fuzzy PID controller was reported to improve control accuracy and response time for unmanned cutting (P. Wang, Shen, Ji, et al., 2021b). A speed-estimation controller based on a BP network with improved PSO optimization leveraged multisensor information to raise efficiency and extend service life (P. Wang, Shen, Li, et al., 2021b). Further work proposed adaptive swing-speed control driven by multisensor inputs using an RBF neural network with GA optimization to handle

changing loads (D. Wang et al., 2022). A self-adaptive scheme combined a BP network with a sliding-window mechanism to tune PID gains and stabilize power under abrupt hardness changes (Feng & Changqing, 2021). To maintain near-constant power, a judging method combined with GA-BP load recognition was used to modulate swing speed in response to varying conditions (Zong et al., 2021).

Table1.2. Machine learning and control methods applied in roadheader automation, with their main uses and supporting references.

ML type	Model / Algorithm	Application in roadheader automation	Reference
Supervised learning	ANN, BP, GA-BP	Mapping GSI to fracture features; support for speed scheduling	Dong et al. (2022)
Supervised learning	Optimized PNN (DE + QR)	Rock-hardness recognition from fused time–frequency features	L. Zhang et al. (2020)
Feature engineering (for supervised models)	Wavelet packet, SVD, LDA	Time–frequency feature extraction and fusion	L. Zhang et al. (2020)
Supervised learning + optimization	BP + improved PSO; Automation Studio logic	Cutting-load identification; coal–gangue discrimination; special-condition handling	P. Wang, Yang, et al. (2021)
Hybrid control (supervised + fuzzy)	BP + fuzzy PID with ISAPSO/PSO	Swing-speed control; higher accuracy and response	P. Wang, Shen, Li, et al. (2021b)
Supervised learning + meta-heuristic	RBF neural network + GA	Adaptive swing-speed control under varying loads	D. Wang et al. (2022)
Hybrid control	Neural network-aided PID	Auto-rectification accuracy and efficiency	M. Zhang et al. (2018)
Hybrid control	Neural network + GA	Automatic cantilever adjustment; reduced power loss	Zhao et al. (2014)
One-class / anomaly detection	VSAPSO-BP	Anomaly detection for rotary tables	Yang et al. (2019)
Supervised learning + online adaptation	BP + sliding window	Constant-power motor control; power-oscillation mitigation	Feng & Changqing (2021)
Supervised learning + meta-heuristic	GA + BP	Constant-power cutting via swing-speed adjustment	Zong et al. (2021)

Table 1.2. (continued)

ML type	Model / Algorithm	Application in roadheader automation	Reference
Supervised learning + optimization	PSO-BP; fuzzy logic	Swing-speed regulation and power stabilization in inclined roadways	M. Zhang et al. (2021)
Supervised / ensemble	MLP; Random Forest; PCA/bagging ensembles	Instantaneous cutting-rate and performance prediction	Seker & Ocak (2019)
Supervised vs. statistical baseline	ANN vs. multiple-variable regression; sensitivity analysis	Performance prediction and factor importance	Salsani et al. (2014)
Supervised + unsupervised	MLP + Kohonen SOFM	Cutting-performance prediction; behavior visualization	Ebrahimabadi et al. (2015)
Planning / meta-heuristic	Ant-colony optimization	Cutting-trajectory planning; collision avoidance	S. Wang & Wu (2019)
Supervised + state estimation	BP + state estimation (e.g., SVD-UKF)	Path tracking under pose-error and skid uncertainty	Y. Qu et al. (2022)
Supervised learning	Naive Bayes; SVM (dynamic cuckoo optimization)	Cutting-arm fault diagnosis	Q. Liu et al. (2021)

ANN = Artificial Neural Network; BP = Backpropagation; GA = Genetic Algorithm; GA-BP = Genetic Algorithm–Backpropagation; PNN = Probabilistic Neural Network; DE = Differential Evolution; QR = QR (orthogonal–upper-triangular) decomposition; SVD = Singular Value Decomposition; LDA = Linear Discriminant Analysis; PSO = Particle Swarm Optimization; PSO-BP = Particle Swarm Optimization–Backpropagation; ISAPSO = Improved Simulated Annealing Particle Swarm Optimization; RBF NN = Radial Basis Function Neural Network; PID = Proportional–Integral–Derivative; MLP = Multi-Layer Perceptron; RF = Random Forest; PCA = Principal Component Analysis; SOFM = Self-Organizing Feature Map; UKF = Unscented Kalman Filter; SVD-UKF = SVD-preconditioned UKF; VSAPSO-BP = Variable Step-size Adaptive Particle Swarm Optimization–Backpropagation.

1.10.3. Motion Control, Rectification, and Posture Regulation

Maintaining line, grade, and chassis stability requires coordinated motion and posture control. Using an auto-rectification kinematic model, a neural-network-assisted PID scheme was reported to improve rectification accuracy and efficiency relative to a PID-only baseline (M. Zhang et al., 2018). For inclined headings, a dynamic rectification model with neural-network-based PID parameter design mitigated slippage and improved running performance (M. Zhang et al., 2021). In coal-mine settings, swing-speed control was tuned with an ML-

assisted optimization approach; the study reports quicker response and steadier tracking when coal hardness changed (F. Li et al., 2022). Posture accuracy during excavation was improved using a fuzzy-neural PID controller derived from kinematic and differential-geometry analysis that links posture error to roadway-section quality (C. Li et al., 2022).

1.10.4. Trajectory Planning and Path Tracking

Learning-aided planning and estimation support safe, efficient face shaping and relocation. Autonomous cutting in complex compositions was planned to use multi-sensor analysis, grid modelling, and an ant-colony trajectory planner, with simulations indicating effective collision avoidance and operational safety (S. Wang & Wu, 2019). Path-tracking robustness under pose-detection errors and track skidding was increased by integrating a BP model with state estimation in a path-correction scheduling strategy (Y. Qu et al., 2022).

1.10.5. Performance Prediction

Reliable estimates of cutting performance support parameter selection and project planning. For instantaneous cutting rate (ICR), Multi-Layer Perceptron and Random Forest performed good among the algorithms evaluated, and ensembles that combined PCA with bagging delivered additional, smaller gains useful for scheduling analyses (Seker & Ocak, 2019). In separate comparison, ANN models outperformed multiple-variable regression and enabled sensitivity analysis to identify influential inputs (Salsani et al., 2014). In coal-measure rocks, an MLP predicted cutting performance and a Kohonen self-organizing map offered insight into system behavior (Ebrahimabadi et al., 2015).

1.10.6. Fault Diagnosis and Anomaly Detection

Operational reliability depends on early detection of faults and anomalies. Tests on cutting-arm data compared BPNN with GA optimization, Naive Bayes with GA optimization, and two SVM variants; in that setting, the SVM using dynamic cuckoo optimization achieved the highest diagnostic accuracy, and the analysis linked likely fault causes to design choices relevant for digital-twin workflows (Q. Liu et al., 2021). For one-class anomaly detection in rotary-table operations, a VSAPSO-BP model distinguished

healthy from faulty behavior with clear separation on the evaluated signals (Yang et al., 2019).

1.11. Organization of the Thesis

This thesis is organized into four chapters. Chapter 1 explains why boom-type roadheaders matter and where current practice falls short. It reviews the study background and motivation, the role of roadheaders in tunneling and mechanical excavation, key industrial pain points, variability in rock-mass properties, and the state of digitalization in mining. It then defines the problem context and scientific gap, notes the limits of empirical and physics-only approaches, and states the research questions and hypotheses. Chapter 2 details the methods. It covers materials and sample preparation; cutting head build and modernization with sensor integration (11 kW drive, about 22.5 cm radius head with 80° conical picks at 50° attack angle); mechanical tests (UCS, E, BTS); the excavation design (full factorial 3×3×3 over depth, rotational speed, and table speed) and unrelieved cutting; data capture and; most-important-feature detection with linear and ensemble models; and a particle swarm optimization routine that returns implementable set points. Chapter 3 presents and interprets results, including mechanical property statistics, confirmation that the machine-frame X axis is the dominant vibration direction, model performance for specific energy and vibration (Gradient Boosting and Random Forest versus Ridge and Lasso), and optimization of cutting variables with a clear trade-off between energy use and vibration. Chapter 4 concludes by answering each research question, stating contributions and limits, and showing how the lab pipeline, especially vibration thresholds and PSO-derived set points, can transfer to field roadheader operation to support adaptive, vibration-aware control.

2. MATERIAL AND METHODS

2.1. Materials

2.1.1. Materials Used for Samples Preparation

For the preparation of concrete samples, only sand and cement were used; no coarse aggregate was added. The Güsey Concrete Plant, located in the Yomra region of Trabzon Province (Figure 2.1), supplied the sand. A Portland cement type CEM I 42.5 R, produced by the Akçakale manufacturing group in Trabzon, was selected as the binder material (Figure 2.2). Portland cement is the predominant type of cement utilized in building due to its strength, durability, and versatility. Several significant properties influence its behavior. Compressive strength indicates the material's capacity to withstand substantial loads, whereas setting times denote the duration required for the material to transition from a fresh mixture to a solidified condition. The heat of hydration, resulting from chemical reactions to water, determines the likelihood of cracking during the initial phases and the progression of strength (Mindess et al., 2003). Tables 2.1 and 2.2 illustrate the chemical and physical parameters of Akçakale cement from the manufacturer's technical datasheet. The chemical composition of a substance dictates its reactivity with water and its ability to form binding compounds. The physical qualities, such as fineness, density, and setting behavior, affect its efficacy in both fresh and hardened conditions.



Figure 2.1. Sand supplied from the Güsey Concrete Plant in Yomra, Trabzon, used in sample preparation



Figure 2.2. Portland cement type CEM I 42.5 R manufactured by the Akçakale Manufacturing Group, Trabzon, used as the binder in sample preparation

Table 2.1. Chemical composition of CEM I 42.5 R cement (average Jan–Sep 2021) versus TS EN 197-1 requirements.

Property	Value (%)	TS EN 197-1 Requirement
SiO ₂	17.75	—
Al ₂ O ₃	4.60	—
Fe ₂ O ₃	2.97	—
CaO	58.96	—
MgO	1.55	—
SO ₃	3.05	≤ 4.0
Cl ⁻	0.0110	≤ 0.1
Equivalent alkali (Na ₂ O)	1.03	—
Loss on ignition	3.91	≤ 5.0
Insoluble residue	0.87	≤ 5.0

Table 2.2. Physical properties of CEM I 42.5 R cement (average Jan–Sep 2021)

Property	Value	TS EN 197-1 Requirement	Unit
Density	3.10	—	g/cm ³
Initial setting time	170	≥ 60	minutes
Final setting time	225	—	minutes
Volume expansion	1.0	≤ 10	mm
Fines (<32 μm)	9.4	—	%
Blaine fineness	3356	—	cm ² /g
Compressive strength (2 d)	29.3	≥ 20	MPa
Compressive strength (28 d)	53.0	42.5–62.5	MPa

2.1.2. Design of Casting Molds

Concrete specimens were cast in a wooden mold that was custom designed with internal dimensions of $30 \times 30 \times 10$ cm and constructed from timber with a thickness of 4cm. This thickness is the standard size of timber commonly available in the market. To meet the experimental program, a total of 50 identical molds were constructed. The schematic drawing of the mold and its dimensions is shown in Figure 2.3a, while the photograph of the fabricated mold is presented in Figure 2.3b.

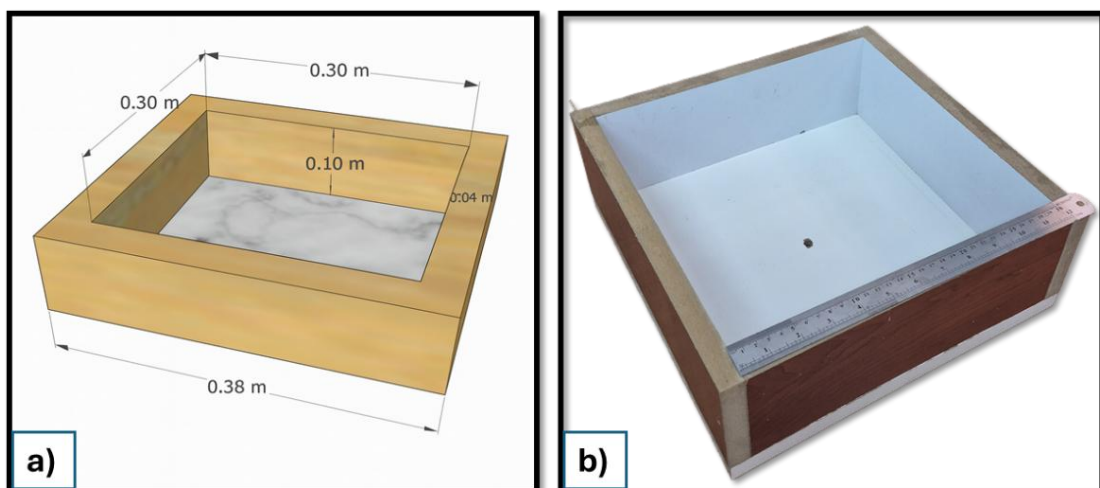


Figure 2.3. Wooden mold designed for specimen preparation: (a) schematic drawing with dimensions in cm; (b) fabricated mold used in the laboratory

2.1.3. Samples Casting

In this study, three different kinds of concrete samples are cast. The proportions of sand, cement, and water were prepared to produce specimens with targeted uniaxial compressive strength (UCS) values, as illustrated in Table 2.3. The casting process was carried out step by step to maintain uniformity and reliability across all samples. Based on the sample number, the required quantities of sand, cement, and water were first weighed and mixed in an ALTRAD mechanical mixer for 2-3 minutes until a consistent blend was obtained. While the concrete was being mixed, the wooden molds were brushed with a thin layer of oil. This stopped the mix from sticking to the surfaces and made it easier to remove the specimens later without damage. Compaction began with hand-tamping using steel rods, which pushed out trapped air and helped the material settle, especially around the corners. The molds were then set on the present vibration table (Yuksal Kaya Makina) in the laboratory to finish the compaction and remove any remaining voids. The surface of each specimen was leveled and wiped clean, and an identification code was added before the curing stage. The sequence of specimen casting is illustrated in Figure 2.3, while Figure 2.4 presents the equipment used: (a) the Yuksal Kaya Makina vibration table, (b) the compaction rods, and (c) the ALTRAD mechanical mixer.

Table 2.3. Mix proportions of cement, water, and sand used for preparing concrete specimens

Sample ID	Cement(g)	Water(g)	Sand(g)
350-3(Mix 1)	3850	2500	21587
250-2 (Mix 2)	4125	3000	29436
175 (Mix 3)	2888	3000	29436

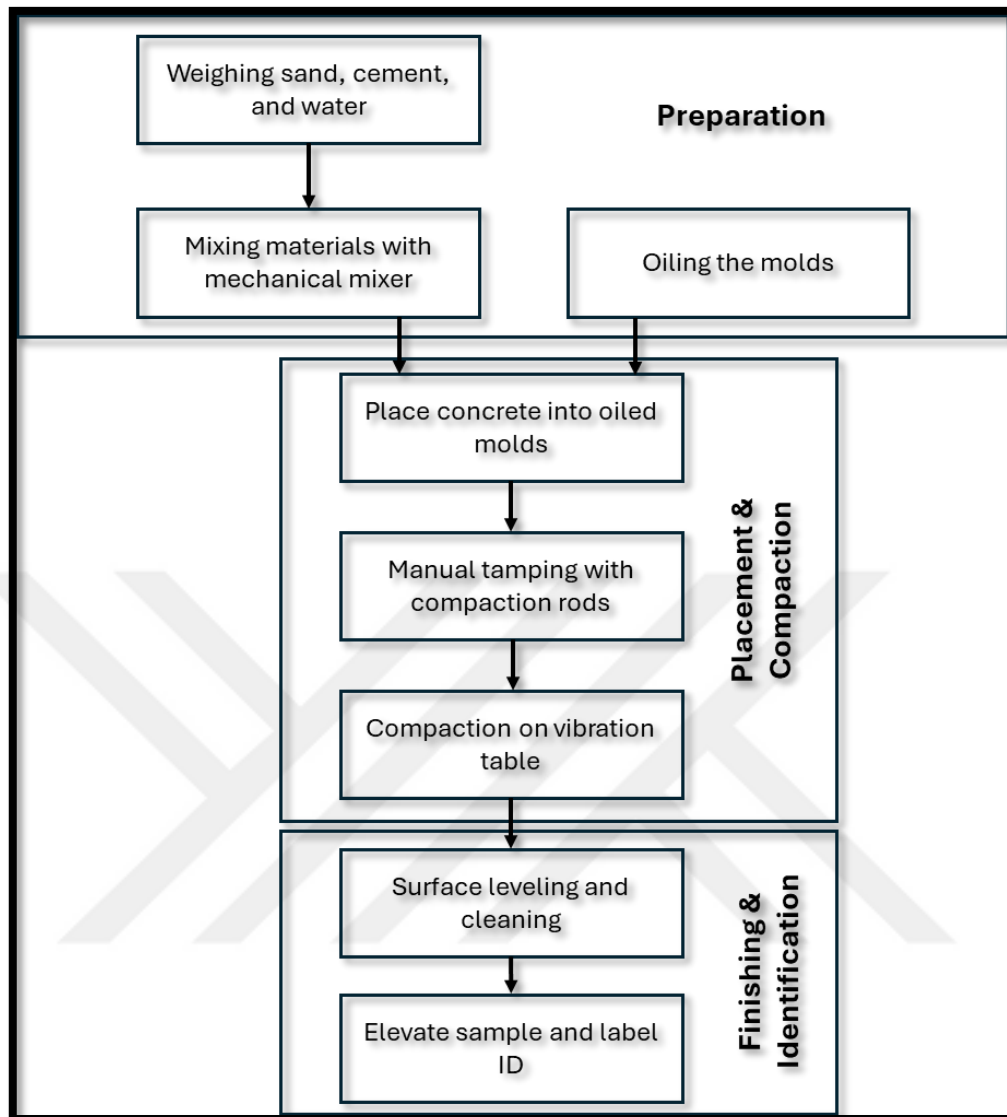


Figure 2.4. Sequence of specimen casting steps

2.1.4. Samples Curing

After casting, the specimens were left in their molds for about 24 hours at room temperature to allow the concrete to set (Figure 2.6). Keeping the samples in the molds at this stage prevented moisture loss and helped avoid surface cracks. Once the initial setting period had passed, the molds were removed, and the specimens were placed in a water tank for curing (Figure 2.7).

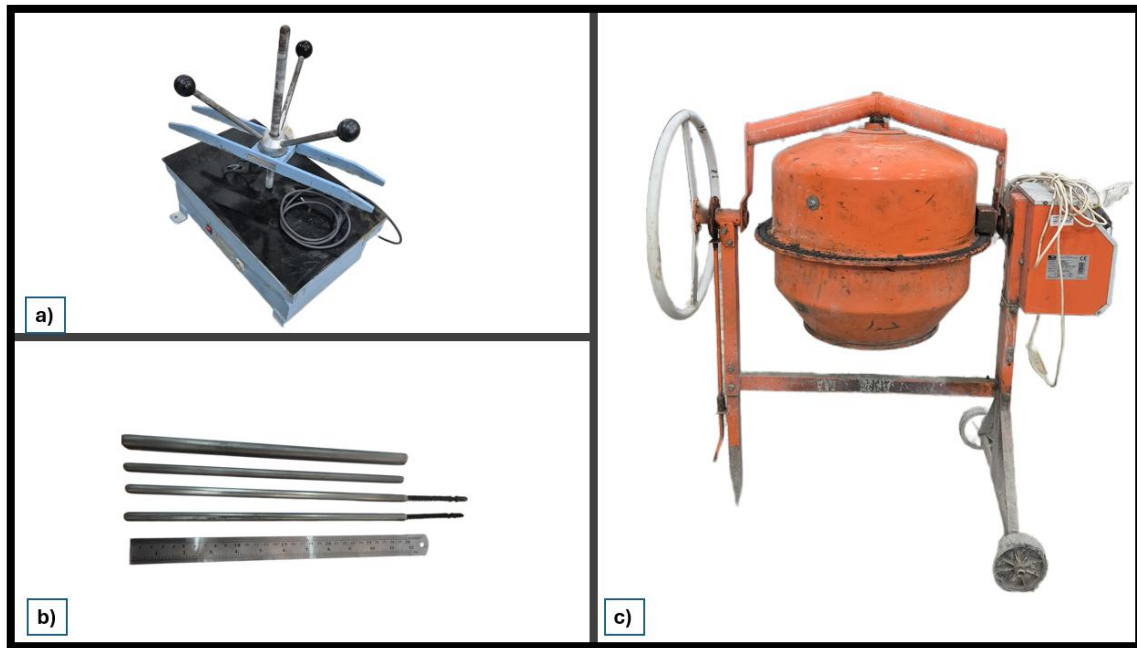


Figure 2.5. Equipment used for casting: (a) Vibration table, (b) Steel compaction rods, and (c) Mechanical mixer



Figure 2.6. Freshly cast specimens kept in their molds for 24 hours prior to demolding

The curing water was kept at roughly 20 ± 2 °C so that the concrete stayed fully hydrated and gained strength in a controlled way. Immersion curing was chosen because it ensures continuous moisture, which is essential for proper hydration of the cement paste. The specimens were kept submerged in water until their scheduled testing age, with an initial curing period of 7 days in the tank.



Figure 2.7. Concrete specimens submerged in a water tank during the curing stage

2.2. Manufacturing of Cutting Head

The laboratory-scale rotary cutting head was developed at Karadeniz Technical University with support from an internally funded project (Project No. FBA-2020-8537). The purpose of this work was to replicate the excavation principles and rock-cutting mechanisms of roadheaders at a reduced scale. To operate the cutting head, a rotary motion system was required. For this purpose, the marble cutting experimental setup in the Mining Engineering Department of KTU was modified and used (Figure 2.8). The system consisted of a cutting disc, motors, a load table, an automation unit, and a manual control panel. Cutting was carried out by moving the wagon carrying the specimen beneath the rotating disc. Instead of moving the disc across the sample, the specimen itself was shifted relative to the fixed cutter. Wagon movement was controlled through a chain–sprocket assembly installed beneath the wagon, ensuring stable and smooth motion (Aydin, 2012).

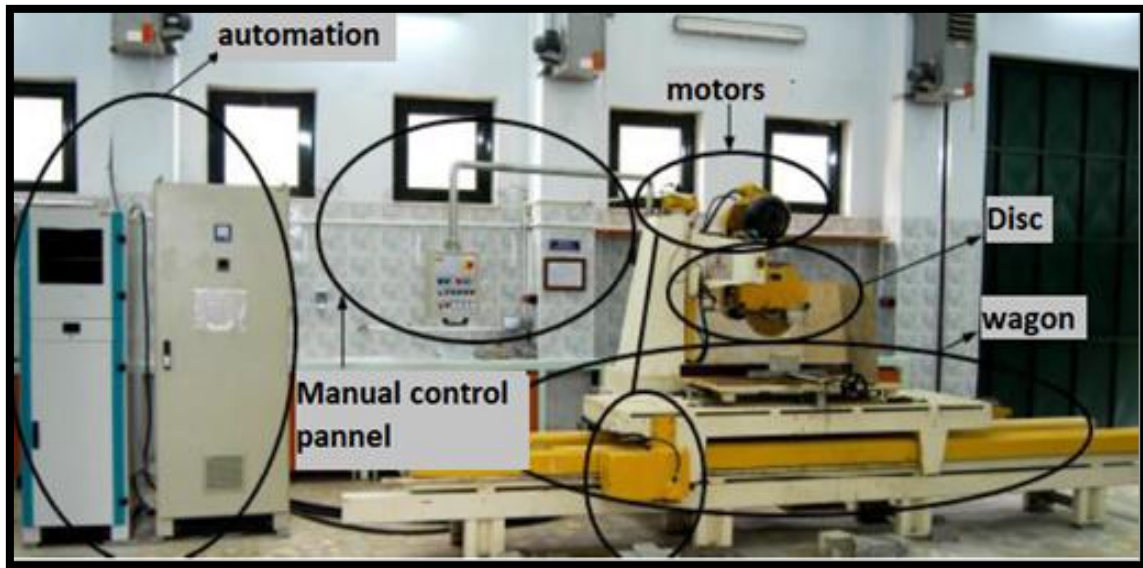


Figure 2.8. Marble cutting experimental setup at the Mining Engineering Department of Karadeniz Technical University (adapted from Aydın, 2012)

Pick forces generated during the cutting process were measured using 4 force gauges mounted on the wagon. Vertical and horizontal positioning of the disc was controlled by two 0.75 kW induction motors. Disc rotation was driven by an 11-kW asynchronous motor, while wagon movement was provided by a 0.75-kW asynchronous motor connected to a chain–sprocket assembly (Aydın, 2012).

The original disc cutter of the marble cutting machine (Figure 2.9a) was removed, and the newly manufactured rotary cutting head was installed in its place. The cutting head was driven by an 11-kW motor. It was fabricated from stainless steel, and bit holders designed with different attack angles were mounted on its surface (Figure 2.9b). The bit holders were positioned at attack angles of 45° , 50° , and 55° . Conical picks, each measuring 8.5 cm in total length from tip to shank and featuring a cone angle of 80° , were used (Figure 2.10).

The cutter head radius (measured as the distance between the pick tip and the drum center) is approximately 22.5 cm (Figure 2.10). By comparison, cutter heads of full-scale roadheaders typically have a radius of about 50 cm and are powered by motors of around 200 kW. Given the scaled-down radius, the available 11 kW motor was considered sufficient for laboratory operation.

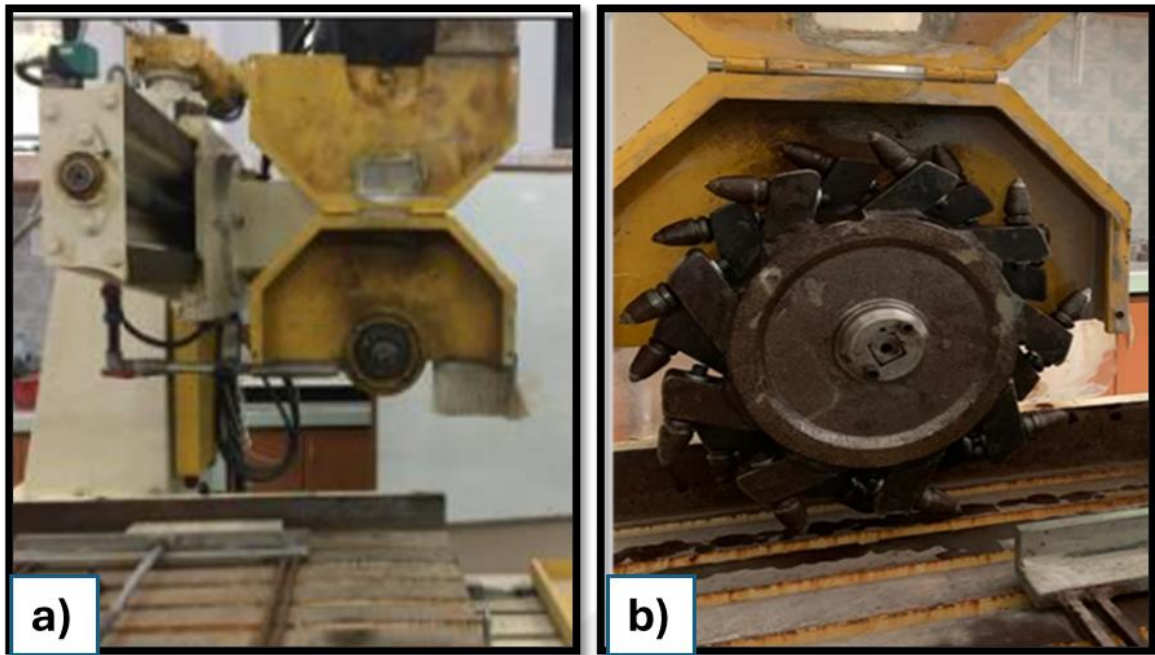


Figure 2.9. Modification of the marble cutting machine: (a) disassembled disc cutter housing, and (b) mounted laboratory-scale rotary cutting head with conical picks

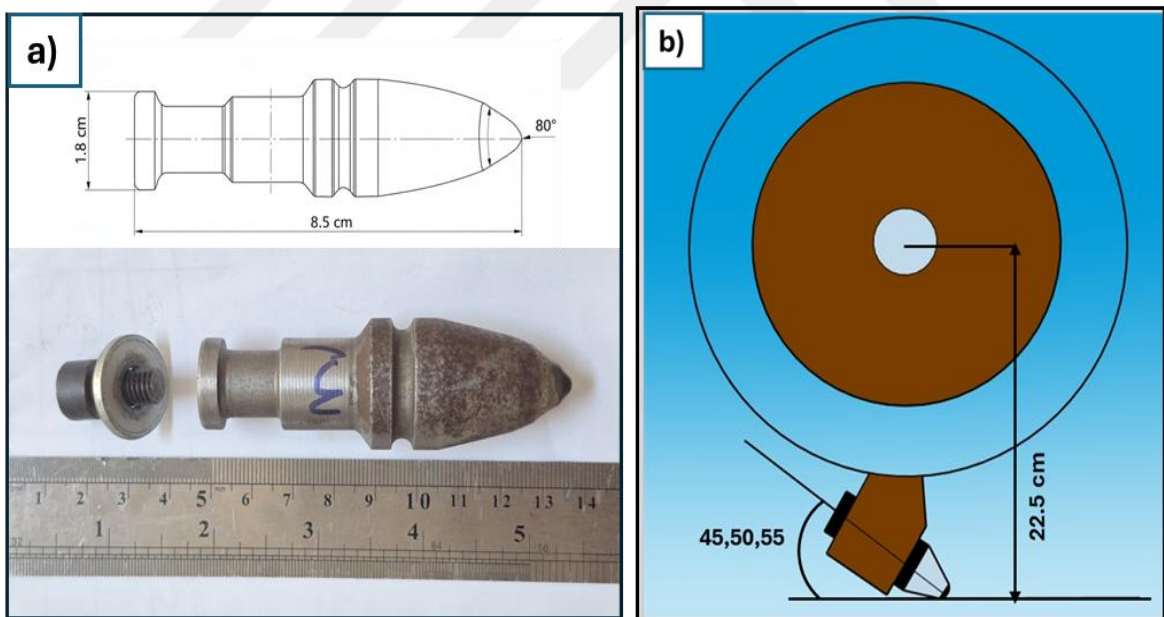


Figure 2.10. (a) Used conical pick (Schematic and real one) (b) Rotary cutting head schematic

2.3. Cutting Head Modernization and Sensor Integration

Modernization of the laboratory cutting head was funded by TÜBİTAK under Project 123M879, “Development of an AI-Supported Autonomous Excavation Optimization System for a Boom-Type Roadheader”. Two vibration sensors were installed on the cutting head. The X-direction sensor is an MTN/1185C model (Vibra Tek, Türkiye), while the Z-direction sensor is a HS-1501005008 model (RI-SE, Türkiye). To measure cutting depth and spacing in excavation experiments, two linear magneto strictive transducers were mounted on the Y- and Z-axes of the machine.

Figures 2.11a–c present schematic views of the rotary cutting head machine from the front, back, and side, showing the positions of sensors, motors, and structural elements. Figure 2.12 gives a view of the full setup in the laboratory, while Figures 2.13 and 2.14 illustrate the actual installation of the vibration sensors on the cutting head and the placement of the linear transducers, respectively.

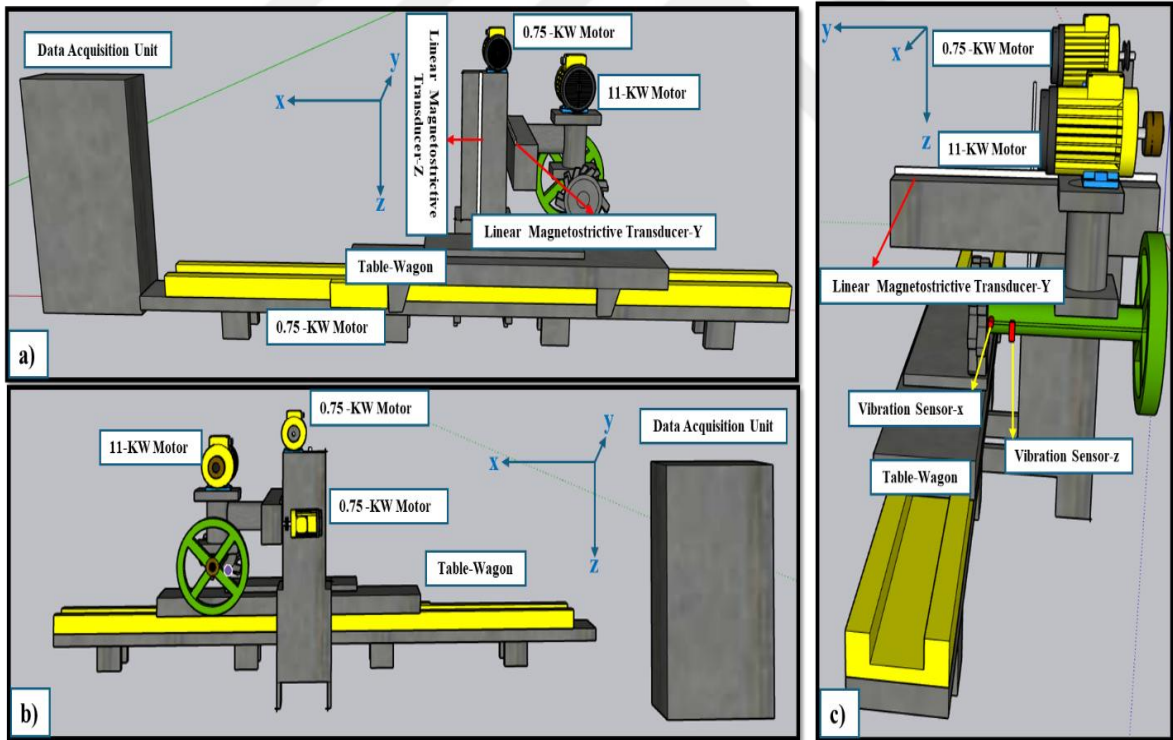


Figure 2.11. Schematic views of the machine set up (a) front, (b) back, and (c) side, indicating the locations of sensors, motors, and key structural members

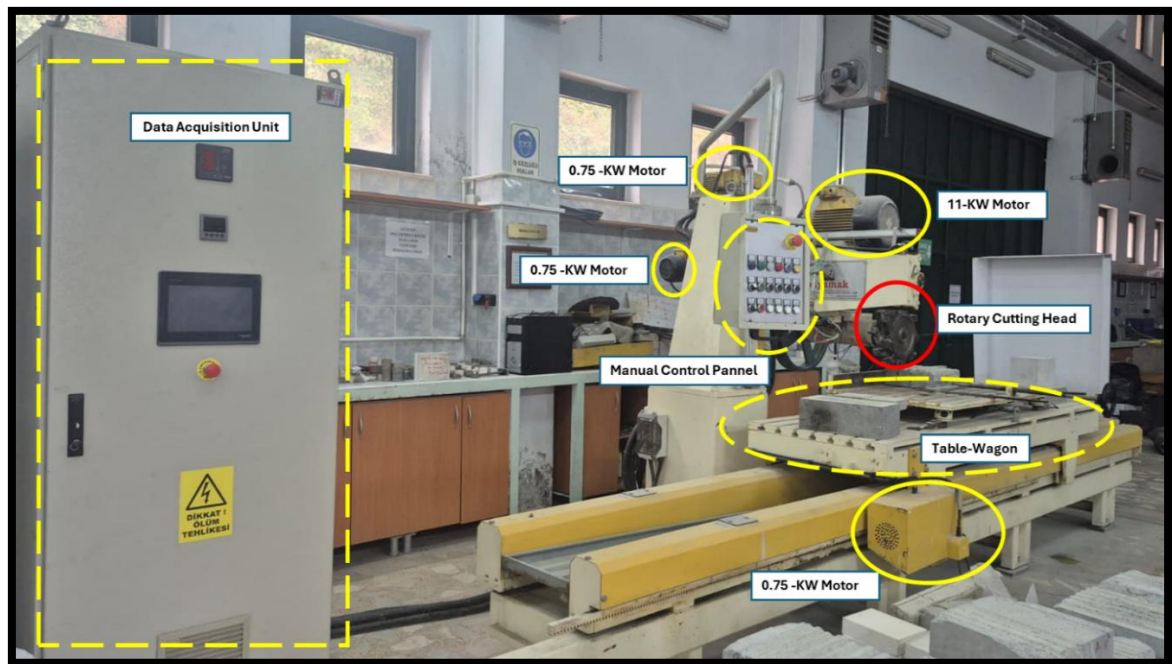


Figure 2.12. View of the full setup of rotary cutting head machine

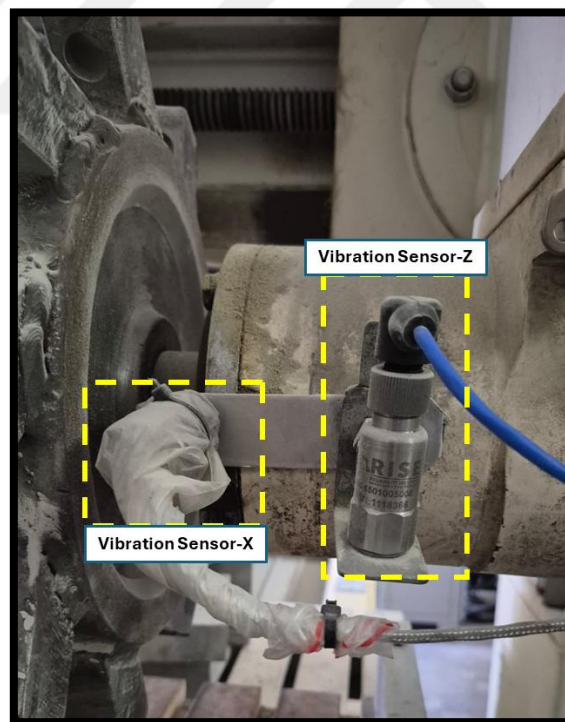


Figure 2.13. Actual installation of the vibration sensors on the cutting head

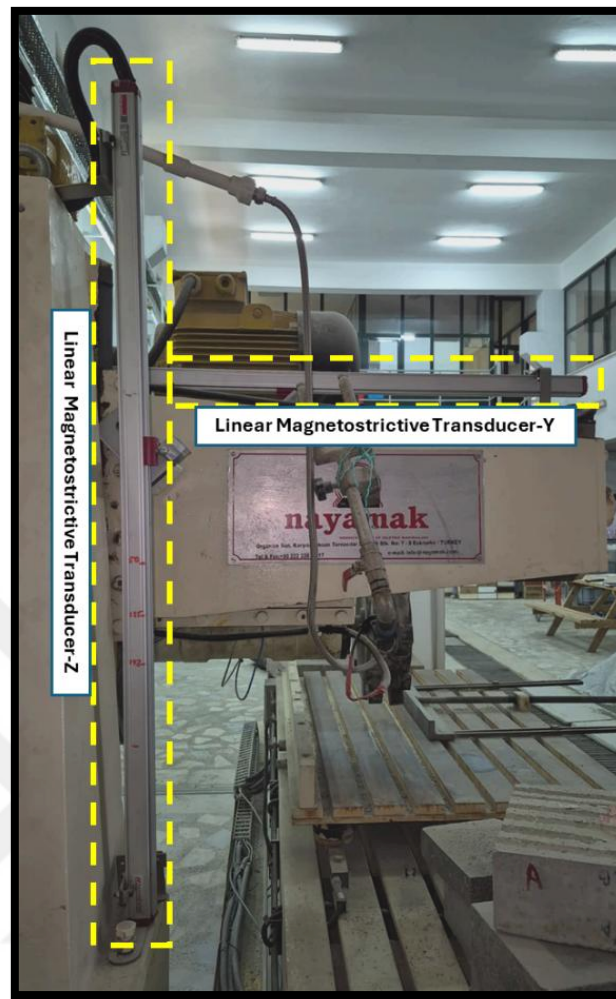


Figure 2.14. Placement of the linear displacement transducers on the machine

Additional upgrades included a 500 kN inductor load cell for penetration force measurement, a PLC-based data logging unit with modular input–output capability, and a dedicated automation computer supported by a 12.1-inch sunlight-readable TFT display. For data acquisition, control, and post-processing, a Dell Vostro 3520 laptop was also integrated into the system. The laptop is equipped with a 12th Gen Intel® Core™ i7-1255U processor (1.70 GHz), 8 GB RAM, Intel® UHD graphics, a 512 GB SSD, and runs on Windows 11 Pro. A detailed overview of the upgraded equipment and their technical specifications is provided in Table 2.4.

Table 2.4. Upgraded equipment and technical specifications, including component models, operating ranges, sampling rates, accuracy, and mounting locations.

Equipment	application	Technical details
Accelerometer (Vibration Sensor)	Measurement of cutter head vibrations	Brand: Hansford Sensors (distributed under RI-SE in Turkey) Model: HS-1501005008 Output: 100 mV/g (AC acceleration) Connector: 2-pin MS, top entry Series: HS-150
Data Logger 4- 20mA- RS485-RS- 232 Output + Connection Units	Collect load cell and sensor data. Providing fast and precise connection between sensor datalogger and datalogger control system	The modular PLC system has a minimum of 16 analog inputs and 8 analog outputs to support analog modules, 32 input and 32 output modules, and 4 fast counter modules
Inductor Loadcell	Measuring cutter head penetration power	500 KN capacity Suitable for instantaneous measurement
Computer	Installation and operation of the automation system	AMD R7 processor 16 GB RAM 512 GB SSD (changeable)
12.1-inch Sunlight Readable TFT Display	Interface for operation of the automation system	16M Colors, 512 MB Flash, Web connection REMOVABLE
Linear Magnetostrictive Transducer	Measurement of Excavation depth	Brand: Opkon Elektronik, Turkey Product: Magnetostrictive Ruler
2-channel frequency counter	Data reading, transfer	32 Bit 6 in-6 out per channel
Shielded cable set	Automation displays connections	40-way terminal -multi-ended at one end- for M340 I/O- 3 m
2-channel counter module	Data reading, transfer	

2.4. Mechanical Properties of Samples

2.4.1. Uniaxial Compressive Strength (UCS)

Cubic specimens measuring $100 \times 100 \times 100$ mm were produced for the UCS tests. Three concrete mix proportions were investigated, identified as Mix IDs 350-3, 250-2, and 175. In total, 11 specimens were prepared: three from Mix 350-3, four from Mix 250-2, and four from Mix 175. After casting, the specimens were cured in a water pool maintained at 18 ± 2 °C for seven days (Figure 2.15 a-c). Following curing, the cubes were removed from the pool and allowed to dry at room temperature until the time of testing.



Figure 2.15. Casting, demolding, and curing of $100 \times 100 \times 100$ mm concrete cubes: (a) placement and compaction; (b) demolding after initial set; (c) water curing at 18 ± 2 °C for 7 days.

The UCS tests were conducted with a BESMAK 200-ton servo hydraulic press at the excavation and mechanization laboratory of the Mining Engineering Department at Karadeniz Technical University. The testing procedure followed ASTM C109/C109M (ASTM, 2021). It should be considered that the standard requires a controlled loading rate

to avoid shock loading and rate-dependent strength loss. For 50-mm cubes, the specified rate is 0.9 ± 0.2 kN/s, which is equivalent to a stress rate of 0.25 ± 0.05 MPa/s. Since the specimens in this study were 100 mm cubes for UCS tests, the loading rate was recalculated based on the loaded surface area. Each cube had a bearing face of 0.01 m^2 ($100 \times 100 \text{ mm}$). The equivalent force rate was determined by multiplying this area with the target stress rate, yielding a load application rate of approximately 2.5 kN/s, within an acceptable range of 2.0–3.0 kN/s. For a target stress rate of 0.25 MPa/s, the force application rate was:

$$F = \sigma \times A = \left(0.25 \times 10^6 \left(\frac{\text{N}}{\text{m}^2} \right) \right) \times 0.01 \text{ m}^2 = 2500 \frac{\text{N}}{\text{s}} = 2.5 \text{ kN/s}$$

With the permitted range of 0.20–0.30 MPa/s, the equivalent force rate for 100 mm cubes was 2.0–3.0 kN/s, with 2.5 kN/s taken as the target. The testing machine was therefore operated within this range. All mixes met ASTM C109/C109M validity criteria. The 350-3 cube failed by diagonal shear with minor spalling (Fig. 2.16a); 250-2 showed a similar shear band with slight platen crushing (Fig. 2.16b); 175 failed by diagonal shear with localized spalling (Fig. 2.16c).

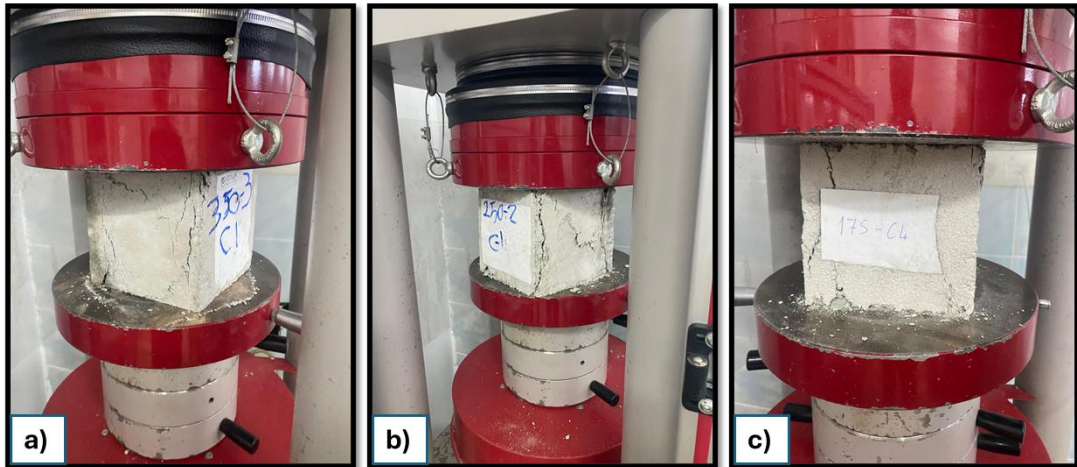


Figure 2.16. UCS failure modes of concrete cubes for three mixes: (a) 350-3, (b) 250-2, (c) 175

2.4.2. Modulus of Elasticity (E)

Concrete blocks were cast in $30 \times 30 \times 10$ cm wooden molds. After demolding, they were water cured for 7 days at 18 ± 2 °C and then kept at room temperature. Cores were

drilled from these blocks and trimmed for testing in line with the ISRM Suggested Methods (Ulusay & Hudson, 2007). The diameter of the cored samples was 54.7 mm (NX). Each core met the geometry tolerances: $L/D = 2.0\text{--}2.5$, end faces flat and parallel, and perpendicular to the axis. Figure 2.17 shows the coring procedure; Figure 2.18 shows the trimmed cores prepared for the static modulus of elasticity (E) test alongside the companion cores for the Brazilian Tensile Strength (BTS) test.

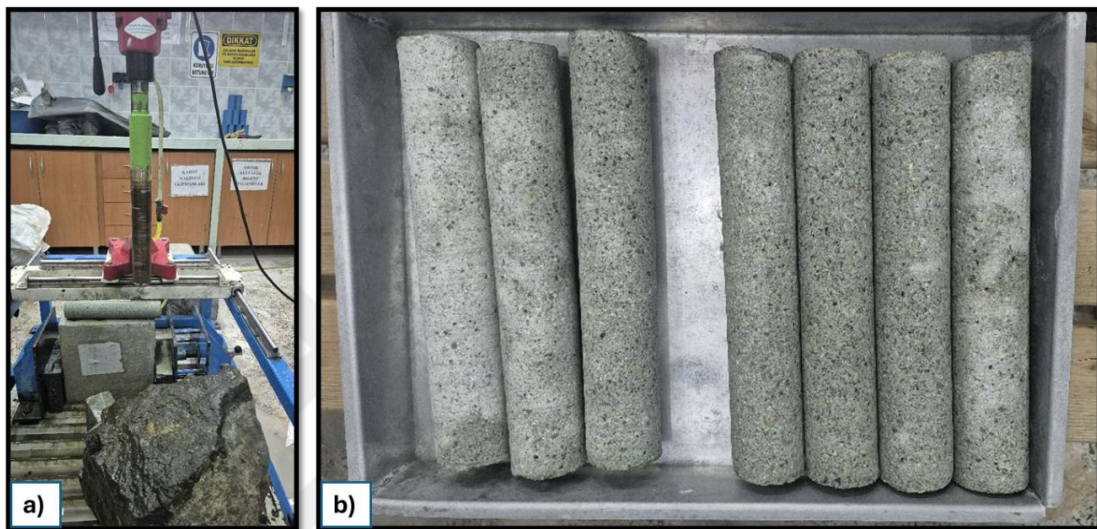


Figure 2.17. Coring procedure and recovered specimens: (a) laboratory coring rig used to extract cylindrical; (b) obtained cores after coring, prior to trimming

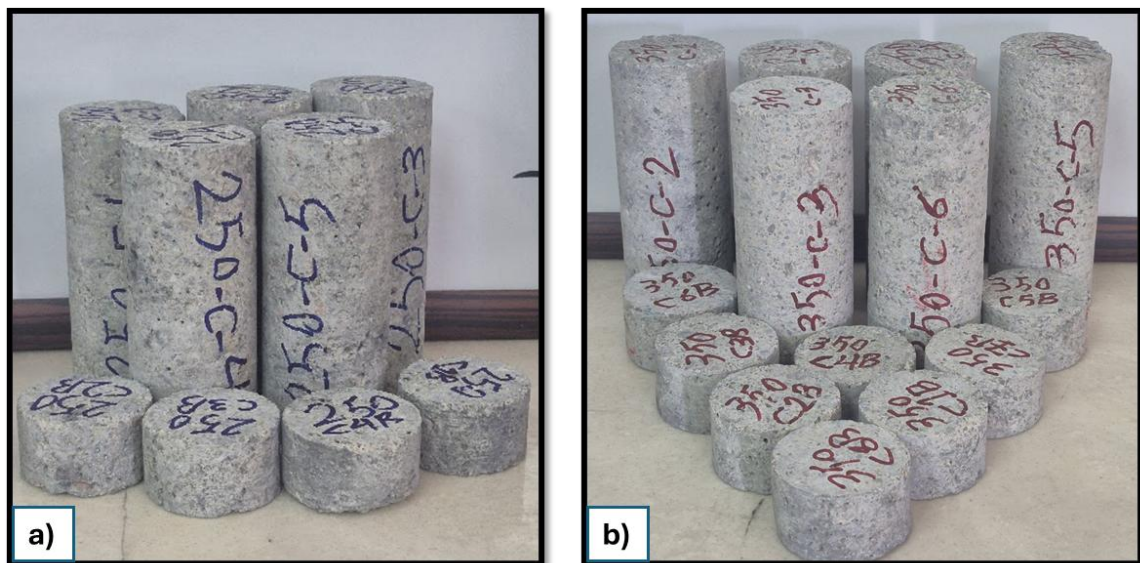


Figure 2.18. Trimmed cores prepared for modulus testing (companion cores also used for BTS)

Tests were performed in the previously mentioned BESMAK 200-ton servo-hydraulic press. Axial load was recorded by the machine load cell. Axial strain was measured with external LVDTs mounted directly on the specimen over a defined gauge length, as recommended by ISRM to avoid frame compliance. LVDT placement is shown in Figure 2.19. Two LVDTs were mounted axially on opposite sides, and their readings were averaged to reduce bending influence, while a third LVDT was mounted horizontally (diametrically) to monitor lateral strain.



Figure 2.19. LVDT layout for modulus test; two vertical and one horizontal on the specimen, mounted over the gauge length per ISRM

The static modulus was taken from the linear portion of the stress–strain curve as a secant modulus:

$$E = \frac{\sigma_1 - \sigma_2}{\varepsilon_1 - \varepsilon_2} \quad (21)$$

Where σ_1 and σ_2 are stresses and ε_1 and ε_2 are the corresponding axial strains within the quasi-linear range (typically between ~30% and 60% of peak stress), following ISRM guidance (Figure 2.20).

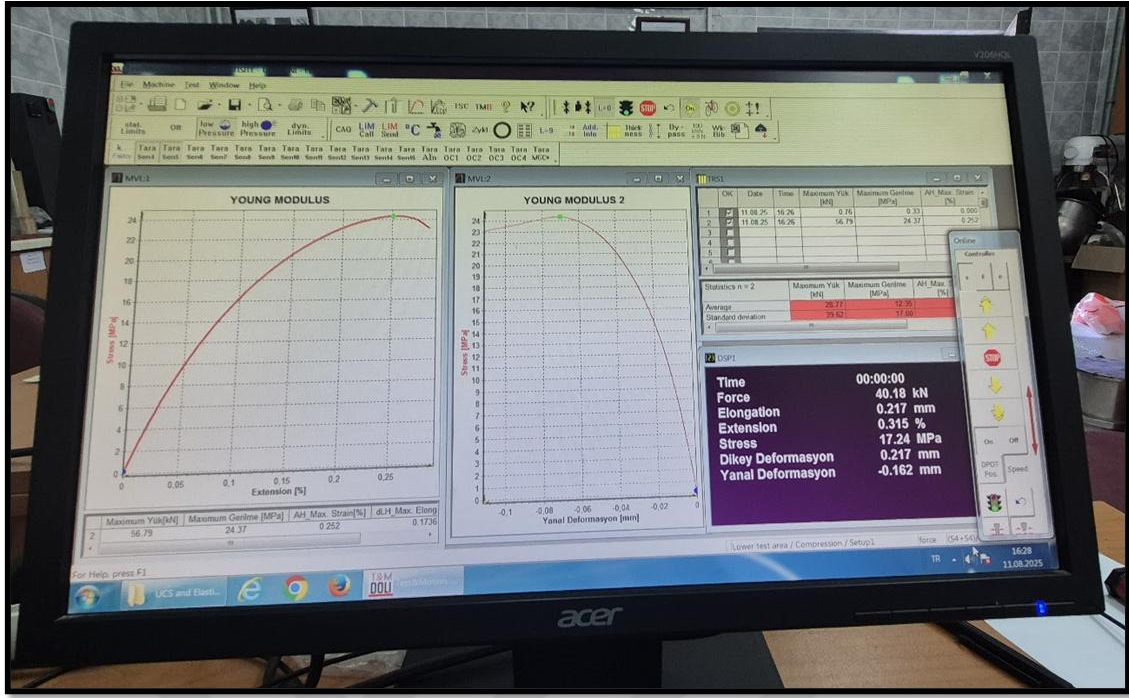


Figure 2.20. Screenshot of the elasticity test interface (BESMAK/T&M): real-time plots of axial stress–axial strain (“Young Modulus1”) and lateral stress–lateral strain (“Young Modulus 2”)

2.4.3. Indirect Tensile Strength (ITS) or BTS

The indirect tensile strength (ITS), commonly called the Brazilian tensile strength (BTS), was determined on core-disc specimens in accordance with the ISRM Suggested Methods for rock materials (Ulusay & Hudson, 2007). Discs were prepared from $30 \times 30 \times 10$ cm blocks; end faces were flat/parallel and geometric tolerances satisfied $0.5D \leq t \leq 0.75D$, giving a target thickness range $t \approx 27 - 41$ mm for the measured diameter $D=54.6$ mm.

Tensile strength was computed from the peak load $P_{\max}$ as:

$$\sigma_t = \frac{2P_{\max}}{\pi Dt} = 0.636 \frac{P_{\max}}{Dt} \quad (22)$$

Loading was applied at 2 N/s under displacement control. A total of 8 valid discs were tested for the 350-3 mix and 4 for the 250-2 mix. Specimens exhibiting off-axis or shear-controlled failure were rejected in accordance with ISRM validity criteria; representative diametral splitting is shown in Figure 2.21.

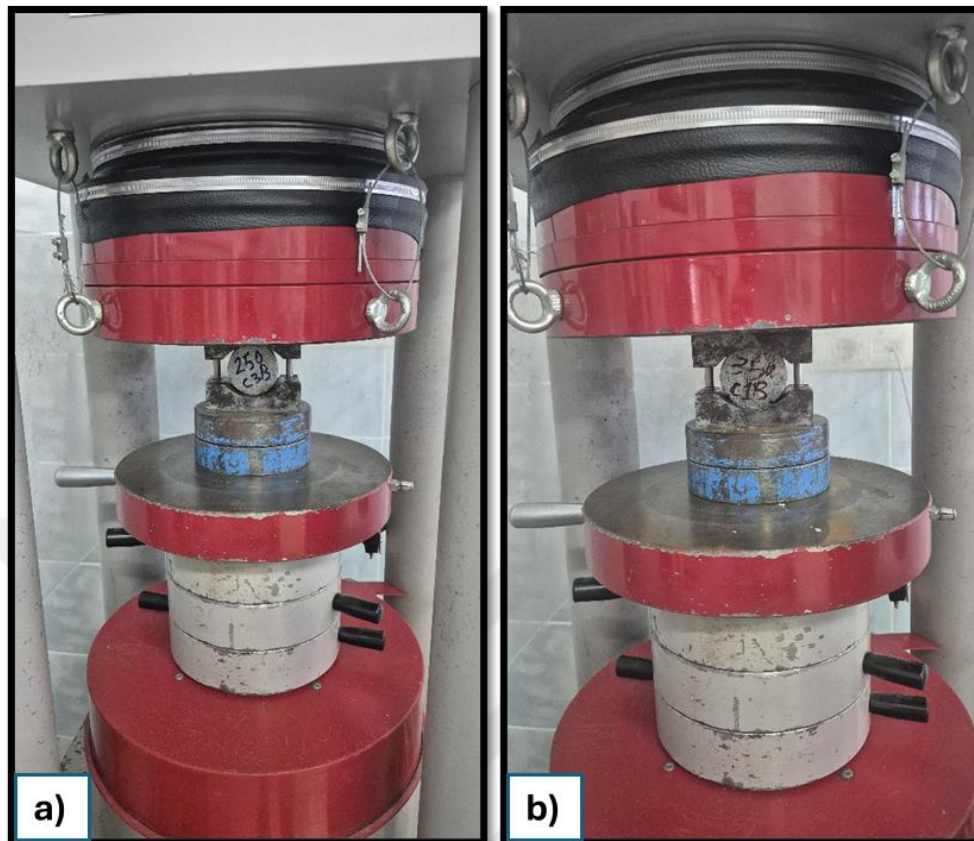


Figure 2.21. Brazilian (splitting) tensile test setup and representative failures for the 250-2 (left) and 350-3 (right) mixes.

2.5. Experimental Design of Excavation Tests

Experimental design provides a structured methodology for investigating the influence of process variables on system performance. In engineering and materials science, rigorous experimental planning is fundamental to ensuring that data collection is both efficient and statistically meaningful. Rather than varying parameters individually, experimental design methods allow multiple factors to be studied simultaneously, improving the ability to capture the complexity of real-world processes (Montgomery, 2017). Factorial designs are widely recognized for their effectiveness in identifying both the independent (main) and interactive effects of process parameters, a feature that is essential when variables are interdependent (Box et al., 2005).

2.5.1. Full Factorial Test Design

A full factorial design considers every possible combination of the selected factors at their indicated levels. This approach differs from one factor at a time method, which only tests single variables and often misses main interactions. By covering the full experimental space, factorial designs make it possible to understand not only the effect of each variable but also how they work together (Myers et al., 2009). This is particularly relevant in mechanical excavation, where performance is rarely controlled by one factor alone. Cutting depth and traverse speed, for example, both affect energy use, but it is often their combined influence that explains whether the process remains stable, produces high vibration, or accelerates tool wear. Full factorial designs require substantial effort because the number of experimental runs grows quickly with the number of factors and levels. Despite this, when the number of factors is limited, they remain the preferred choice. Their strength lies in producing complete statistical information, which supports the development of predictive models and provides a solid basis for optimization (Montgomery, 2017).

2.5.2. Experimental Design of Excavation Tests

In this study, a full factorial experimental design was applied to investigate the cutting performance of concrete samples under controlled laboratory conditions. Three process variables were selected because of their direct influence on mechanical excavation performance:

1. Cutting depth
2. Rotational speed of the cutting head
3. Table speed

Each factor was studied at three levels. This produced a 3^3 design, resulting in 27 experimental runs. This design generated a dataset that supports a detailed evaluation of how operating variables affect key performance indicators, including cutting force, power consumption, vibration, and specific energy. The factor levels and their combinations are summarized in Table 2.4.

Table 2.5. Full factorial design with three variables: cutting depth, cutting head rotational speed, and table speed (27 total tests)

Cutting Depth(mm)	Rotational Speed(m/s)	Table Speed(cm/min)	Test Number
4	3.4	162.80	Test1
4	3.4	187.37	Test2
4	3.4	211.94	Test3
6	3.4	162.80	Test4
6	3.4	187.37	Test5
6	3.4	211.94	Test6
8	3.4	162.80	Test7
8	3.4	187.37	Test8
8	3.4	211.94	Test9
4	3.78	162.80	Test10
4	3.78	187.37	Test11
4	3.78	211.94	Test12
6	3.78	162.80	Test13
6	3.78	187.37	Test14
6	3.78	211.94	Test15
8	3.78	162.80	Test16
8	3.78	187.37	Test17
8	3.78	211.94	Test18
4	4.17	162.80	Test19
4	4.17	187.37	Test20
4	4.17	211.94	Test21
6	4.17	162.80	Test22
6	4.17	187.37	Test23
6	4.17	211.94	Test24
8	4.17	162.80	Test25
8	4.17	187.37	Test26
8	4.17	211.94	Test27

2.6. Rock Cutting Tests

In mechanical excavation, two cutting modes are distinguished based on whether adjacent grooves interact:

1. Unrelieved cutting (also called isolated or single groove cutting) occurs when a single pass is made without a neighboring groove at an interaction distance. The rock ahead and to the sides of the pick remains laterally confined; chip formation is limited, forces and specific energy (SE) are higher, and fine debris is more prevalent, especially at small penetrations. In full-scale linear cutting machines (LCMs), the first pass on a fresh surface is conventionally treated as the unrelieved case.(Bilgin et al., 2006; Jeong et al., 2020).
2. Relieved cutting occurs when a following pass is executed at a spacing that facilitates the convergence of fractures from adjacent grooves, hence creating a free surface and enhancing chip production. Under relieved conditions, cutting forces and SE generally decrease, and coarser chips are produced. The transition between the two modes is governed principally by the spacing-to-depth ratio (s/d), with a broad optimum reported in the literature (typically $s/d \approx 2-5$ for conical picks), and by rock brittleness and confinement state (Averin et al., 2021; Bilgin et al., 2006; Jeong et al., 2020; X. Wang et al., 2022).

Conical-pick cutting is characterized by a crushed zone forming beneath the tip, tensile cracks initiating near the crushed zone boundary, and lateral crack propagation that is highly sensitive to the presence of a neighboring groove. When spacing is large (unrelieved), lateral cracks arrest before linking to a free face; the tool must grind and plough confined material, elevating normal and cutting forces as well as SE. When spacing is near-optimal (relieved), damage fields from adjacent grooves overlap and chips detach between grooves, yielding lower mean forces and SE and a coarser chip-size distribution (Jeong et al., 2020; X. Wang et al., 2022). As illustrated in Figure 2.22, chip-efficient relieved cutting occurs near the optimum spacing, whereas unrelieved cutting at large suppresses groove interaction, produces ridges, and generally increases specific energy.

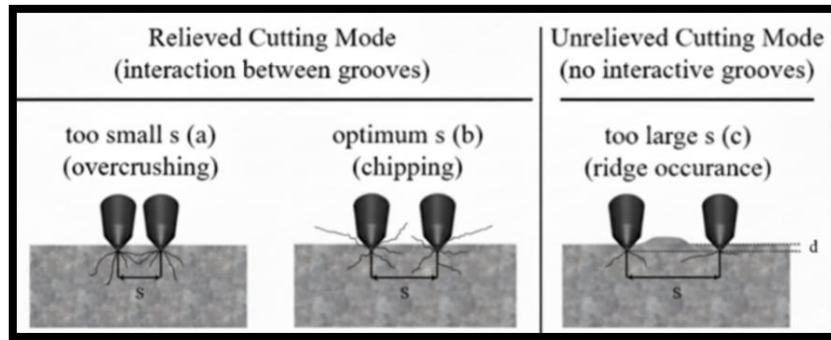


Figure 2.22. Relieved and unrelieved cutting modes with conical picks as a function of groove spacing s relative to depth of cut d . Modified after (Jeong et al., 2020)

In this research, all cutting tests were conducted in the unrelieved mode using a point-attack conical pick (cone angle = 80° , attack angle = 50° , skew angle = 0° , and tilt angle = 0°). For each mix, a 27-point test matrix was executed in triplicate (81 cuts per mix). A new pick was installed at the start of each mix to eliminate wear effects. Following each pass, the real depth of cut (d_{real}) was measured on the groove with a digital vernier caliper (Figure 2.23). Representative post-cut surfaces for the three mixes are shown in Figure 2.24. Figures 2.25–2.27 present the complete specimen sets for Mix-1, Mix-2, and Mix-3 after completion of the three cutting rounds (triplicate runs), shown column-wise from Rep-1 to Rep-3.



Figure 2.23. Post-cut verification of the real depth of cut using a digital vernier caliper on grooves



Figure 2.24. Representative specimens after rotary unrelieved cutting: (a) Mix 1, (b) Mix 2, (c) Mix 3



Figure 2.25. Mix-1 complete specimen set after three rotary unrelieved cutting repeats



Figure 2.26. Mix-2 complete specimen set after three rotary unrelieved cutting repeats



Figure 2.27. Mix-3 complete specimen set after three rotary unrelieved cutting repeats

2.7. Data Acquisition and Cleaning

For data acquisition two laptops were used during the cutting tests. One laptop interfaced with the data-acquisition (SCADA) system to log machine signals (rotational speed, table speed, power, and vibration). Another independent laptop was connected to the sound-level meter to record the acoustic signal and store the sound-pressure-level (SPL) time series. (Figure 2.28).

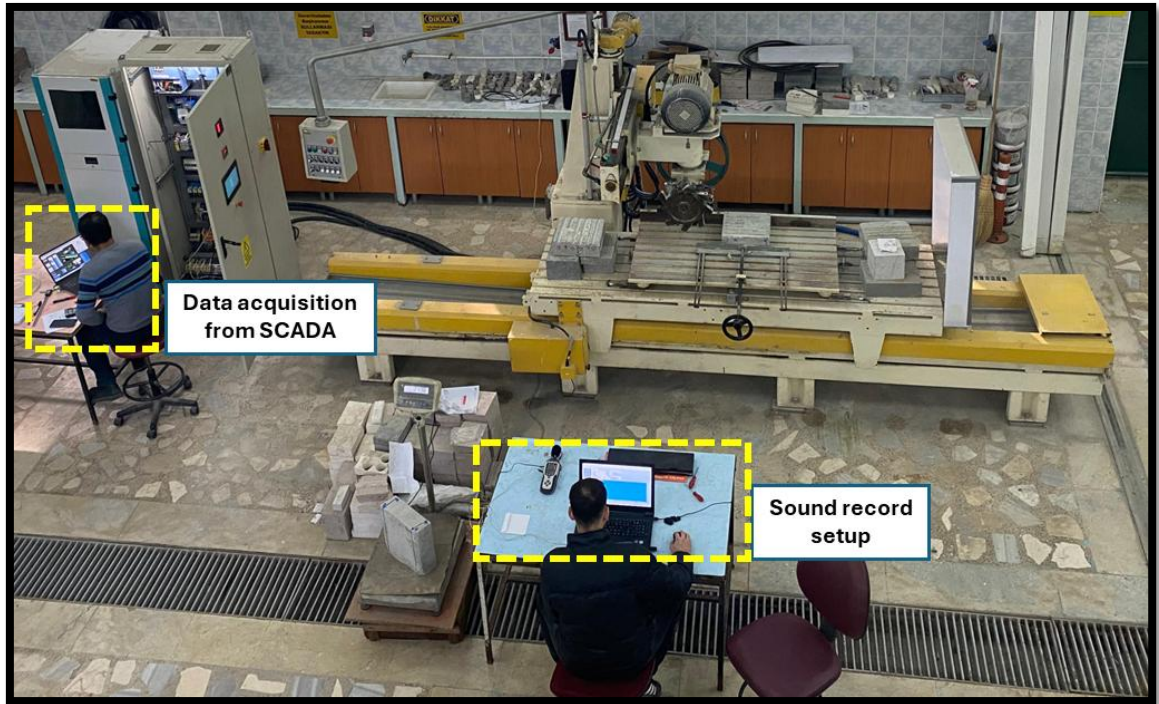


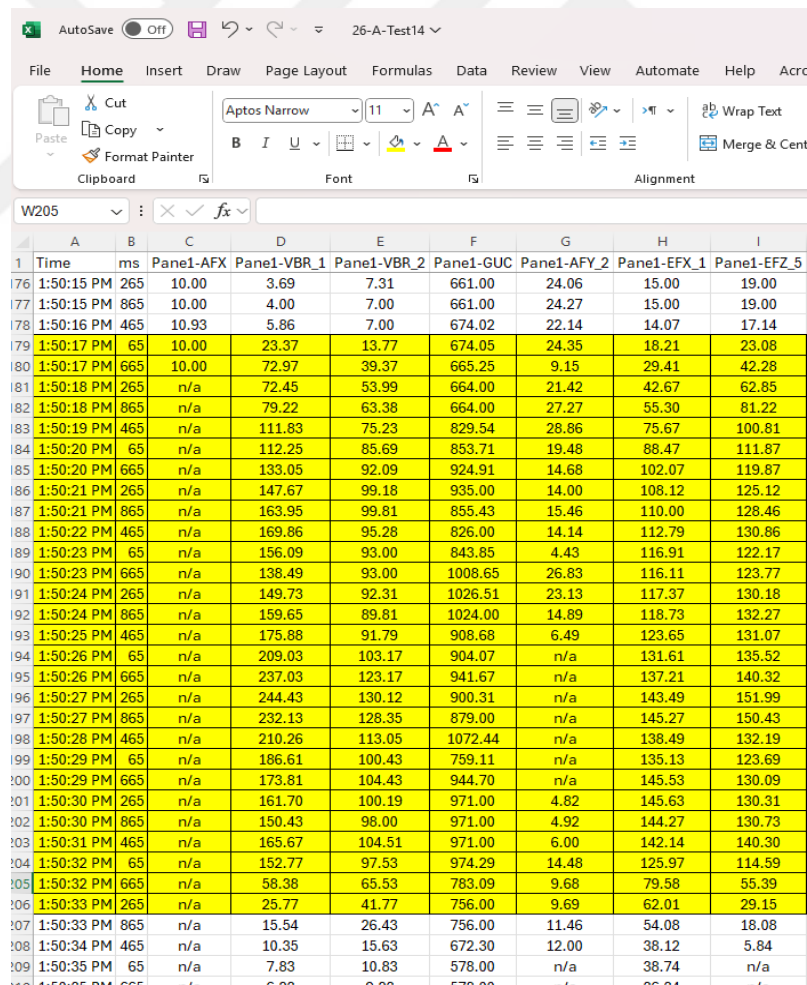
Figure 2.28. Dual data-acquisition setup during cutting tests

In this study, equivalent noise levels were measured with an external sound-level meter (CEM DT-8852) to quantify acoustic emissions as a function of operating parameters (Table 2.6). During testing, the instrument was fixed at a location 1.5 m from the cutting head. Measurements were acquired with C-weighting, a 30–130 dBC measurement range, and Fast time weighting (125 ms detector response). C-weighted sound-pressure levels (dBC) were logged at 0.2 s intervals using the manufacturer's software. Calibration readings were recorded for every session to confirm measurement, accuracy and repeatability. Figure 2.29 provides a representative raw SCADA dataset for Test 14 (Sample 1, Repeat 3), and figure 2.30 presents the corresponding sound-level-meter record for the same test. Following data

acquisition, the interval during which cutting occurred was identified from the signals and is highlighted in yellow. In both figures, the sections highlighted in yellow indicate the cutting time.

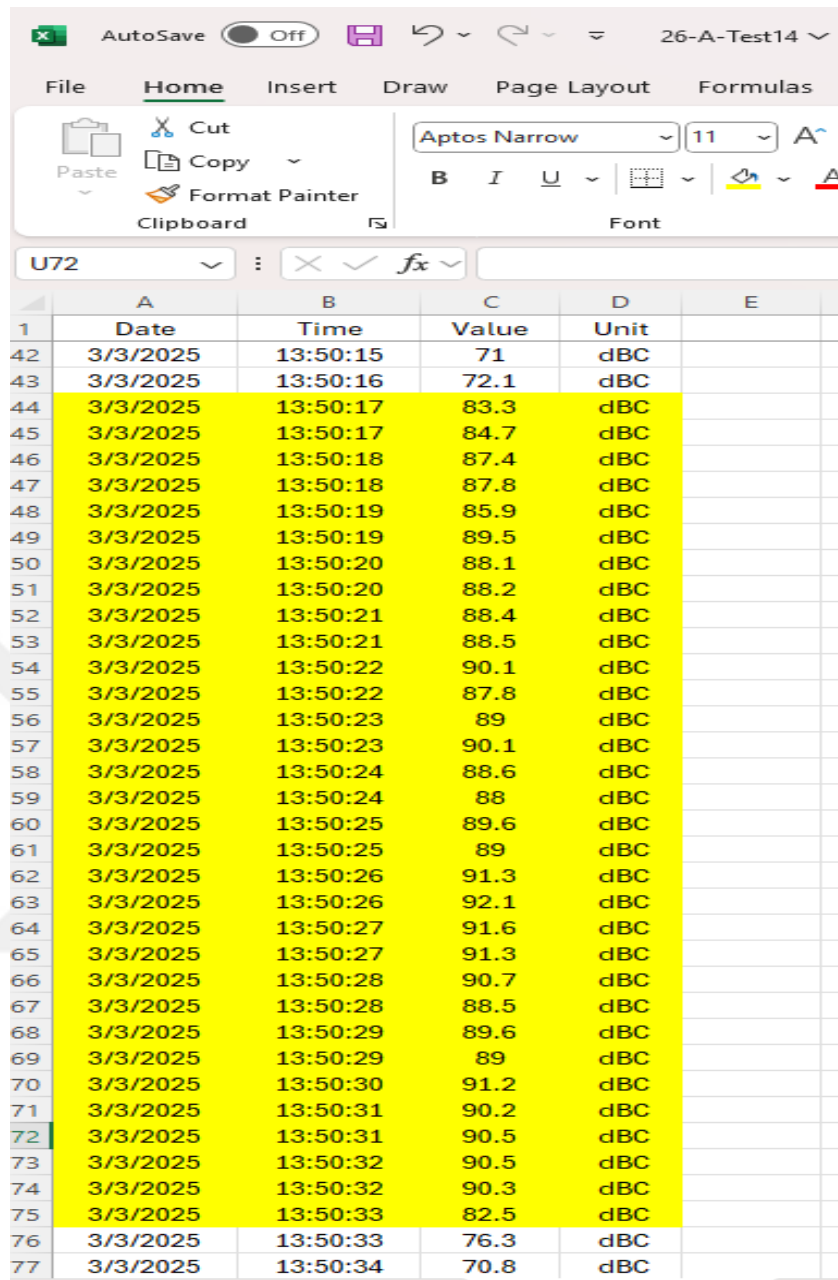
Table 2.6. Key technical specifications of the CEM DT-8852 sound-level meter.

Parameter	Specification
Frequency range	31.5 Hz – 8 kHz
Accuracy	±1.4 dB
Frequency weighting	A / C
Measurement range	30 – 130 dB (selectable)
Time weighting (detector response)	Fast: 125 ms; Slow: 1 s



	A	B	C	D	E	F	G	H	I
1	Time	ms	Pane1-AFX	Pane1-VBR_1	Pane1-VBR_2	Pane1-GUC	Pane1-AFY_2	Pane1-EFX_1	Pane1-EFZ_5
76	1:50:15 PM	265	10.00	3.69	7.31	661.00	24.06	15.00	19.00
77	1:50:15 PM	865	10.00	4.00	7.00	661.00	24.27	15.00	19.00
78	1:50:16 PM	465	10.93	5.86	7.00	674.02	22.14	14.07	17.14
79	1:50:17 PM	65	10.00	23.37	13.77	674.05	24.35	18.21	23.08
80	1:50:17 PM	665	10.00	72.97	39.37	665.25	9.15	29.41	42.28
81	1:50:18 PM	265	n/a	72.45	53.99	664.00	21.42	42.67	62.85
82	1:50:18 PM	865	n/a	79.22	63.38	664.00	27.27	55.30	81.22
83	1:50:19 PM	465	n/a	111.83	75.23	829.54	28.86	75.67	100.81
84	1:50:20 PM	65	n/a	112.25	85.69	853.71	19.48	88.47	111.87
85	1:50:20 PM	665	n/a	133.05	92.09	924.91	14.68	102.07	119.87
86	1:50:21 PM	265	n/a	147.67	99.18	935.00	14.00	108.12	125.12
87	1:50:21 PM	865	n/a	163.95	99.81	855.43	15.46	110.00	128.46
88	1:50:22 PM	465	n/a	169.86	95.28	826.00	14.14	112.79	130.86
89	1:50:23 PM	65	n/a	156.09	93.00	843.85	4.43	116.91	122.17
90	1:50:23 PM	665	n/a	138.49	93.00	1008.65	26.83	116.11	123.77
91	1:50:24 PM	265	n/a	149.73	92.31	1026.51	23.13	117.37	130.18
92	1:50:24 PM	865	n/a	159.65	89.81	1024.00	14.89	118.73	132.27
93	1:50:25 PM	465	n/a	175.88	91.79	908.68	6.49	123.65	131.07
94	1:50:26 PM	65	n/a	209.03	103.17	904.07	n/a	131.61	135.52
95	1:50:26 PM	665	n/a	237.03	123.17	941.67	n/a	137.21	140.32
96	1:50:27 PM	265	n/a	244.43	130.12	900.31	n/a	143.49	151.99
97	1:50:27 PM	865	n/a	232.13	128.35	879.00	n/a	145.27	150.43
98	1:50:28 PM	465	n/a	210.26	113.05	1072.44	n/a	138.49	132.19
99	1:50:29 PM	65	n/a	186.61	100.43	759.11	n/a	135.13	123.69
100	1:50:29 PM	665	n/a	173.81	104.43	944.70	n/a	145.53	130.09
101	1:50:30 PM	265	n/a	161.70	100.19	971.00	4.82	145.63	130.31
102	1:50:30 PM	865	n/a	150.43	98.00	971.00	4.92	144.27	130.73
103	1:50:31 PM	465	n/a	165.67	104.51	971.00	6.00	142.14	140.30
104	1:50:32 PM	65	n/a	152.77	97.53	974.29	14.48	125.97	114.59
105	1:50:32 PM	665	n/a	58.38	65.53	783.09	9.68	79.58	55.39
106	1:50:33 PM	265	n/a	25.77	41.77	756.00	9.69	62.01	29.15
107	1:50:33 PM	865	n/a	15.54	26.43	756.00	11.46	54.08	18.08
108	1:50:34 PM	465	n/a	10.35	15.63	672.30	12.00	38.12	5.84
109	1:50:35 PM	65	n/a	7.83	10.83	578.00	n/a	38.74	n/a
110	1:50:35 PM	665	n/a	6.22	9.22	578.00	n/a	36.24	n/a

Figure 2.29. Exported raw data from SCADA -Test 14 Sample 1, Repeat 3



	A	B	C	D	E
1	Date	Time	Value	Unit	
42	3/3/2025	13:50:15	71	dBC	
43	3/3/2025	13:50:16	72.1	dBC	
44	3/3/2025	13:50:17	83.3	dBC	
45	3/3/2025	13:50:17	84.7	dBC	
46	3/3/2025	13:50:18	87.4	dBC	
47	3/3/2025	13:50:18	87.8	dBC	
48	3/3/2025	13:50:19	85.9	dBC	
49	3/3/2025	13:50:19	89.5	dBC	
50	3/3/2025	13:50:20	88.1	dBC	
51	3/3/2025	13:50:20	88.2	dBC	
52	3/3/2025	13:50:21	88.4	dBC	
53	3/3/2025	13:50:21	88.5	dBC	
54	3/3/2025	13:50:22	90.1	dBC	
55	3/3/2025	13:50:22	87.8	dBC	
56	3/3/2025	13:50:23	89	dBC	
57	3/3/2025	13:50:23	90.1	dBC	
58	3/3/2025	13:50:24	88.6	dBC	
59	3/3/2025	13:50:24	88	dBC	
60	3/3/2025	13:50:25	89.6	dBC	
61	3/3/2025	13:50:25	89	dBC	
62	3/3/2025	13:50:26	91.3	dBC	
63	3/3/2025	13:50:26	92.1	dBC	
64	3/3/2025	13:50:27	91.6	dBC	
65	3/3/2025	13:50:27	91.3	dBC	
66	3/3/2025	13:50:28	90.7	dBC	
67	3/3/2025	13:50:28	88.5	dBC	
68	3/3/2025	13:50:29	89.6	dBC	
69	3/3/2025	13:50:29	89	dBC	
70	3/3/2025	13:50:30	91.2	dBC	
71	3/3/2025	13:50:31	90.2	dBC	
72	3/3/2025	13:50:31	90.5	dBC	
73	3/3/2025	13:50:32	90.5	dBC	
74	3/3/2025	13:50:32	90.3	dBC	
75	3/3/2025	13:50:33	82.5	dBC	
76	3/3/2025	13:50:33	76.3	dBC	
77	3/3/2025	13:50:34	70.8	dBC	

Figure 2.30. Sound-level meter screenshot for Test 14 Sample 1, Repeat 3

After identifying the cutting interval for each test, the maximum vibration along the X- and Z-axes, maximum power, and maximum C-weighted sound level (dBC) were computed within that window. Table 2.7 presents the results for Test 14, Sample 1 (Repeat 3).

Table 2.7. Peak metrics within the identified cutting interval for Test 14 (Sample 1, Rep3)

Test 14 Sample 1 Rep 3	Panel- AFX (kg)	Panel- VBR_1 (mm/s)	Panel- VBR_2 (mm/s)	Panel- GUC (w)	Panel- AFY_2 (kg)	Panel- EFX_1 (kg)	Panel- EFZ_5 (kg)	Sound (dBC)
MAX	10.00	244.43	130.12	1072.44	28.86	145.63	151.99	92.10
AV	10.00	145.52	88.86	876.12	14.96	107.57	111.81	88.47

2.8. Most Important Feature Detection

Understanding which cutting parameters (cut depth, rotational speed, and traverse speed) strongly influence vibration and specific energy is essential for optimizing excavation efficiency. To achieve this, several machine learning (ML) models were implemented and compared. The analysis combined interpretable linear approaches with nonlinear ensemble methods, supported by cross-validation, diagnostic plots, and feature importance measures. Additionally, symbolic regression was applied to obtain compact closed-form equations that can support mechanistic understanding of rock cutting performance. In our work, three linear models and two tree-based models are used. Linear, ridge, and lasso (linear models) require RobustScaler preprocessing to normalize predictor ranges and reduce sensitivity to outliers. In contrast, Random Forest and Gradient Boosting (tree-based models) are scale-invariant, handling raw numeric inputs directly without normalization.

2.8.1. Linear Regression

Linear regression is among the earliest and most basic predictive modeling techniques in statistics and machine learning. It creates a relationship between a dependent variable, y , and one or more independent variables, x_i , by fitting a linear function. The standard formulation is

$$y = \beta_0 + \sum_{i=1}^p \beta_i x_i + \epsilon \quad (23)$$

Here β_0 is the interception, β_i are the regression coefficients, and ϵ is the error term. The coefficients are found by minimizing the residual sum of squares. Under the Gauss–Markov assumptions, this gives the best linear unbiased estimators (Montgomery et al., 2012). In machine learning, linear regression is a supervised method because it trains on

labeled data (observations with known inputs and outputs) to learn predictive relationships (GeeksforGeeks, 2023). Owing to its simplicity, it is frequently adopted as a baseline approach in regression studies, providing a balance of interpretability and computational efficiency and serving as a reference point against which more complex algorithms are evaluated (Hastie et al., 2009). Figure 2.28 illustrates a simple linear regression model. The regression line $y = \beta_0 + \beta_1 x$ represents the best-fit relationship between the predictor X and the response Y . The intercept, β_0 , is the value of Y when $X=0$, while the slope β_1 indicates the rate of change in Y per unit change in X . The random error term (ϵ_i) represents the discrepancy between the observed and predicted values.

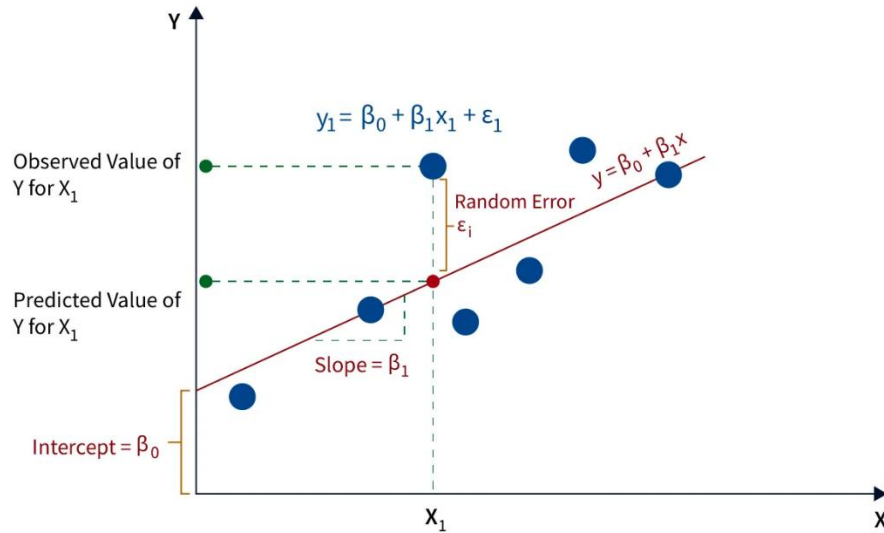


Figure 2.31. Simple linear regression showing the best-fit line, intercept (β_0), slope (β_1), and random error (ϵ_i) between observed and predicted values.

2.8.2. Least Absolute Shrinkage and Selection (Lasso) Regression

Least Absolute Shrinkage and Selection Operator (Lasso) regression is a linear modeling method that adds an L1 penalty to the loss function. The penalty shrinks many coefficients toward zero and often sets some exactly to zero. This gives you a model that is both regularized and sparse. In practice, it helps when the number of features is large or when you expect only a few features to matter (Emmert-Streib & Dehmer, 2019; Muthukrishnan & Rohini, 2016). Lasso Regression is derived from Linear Regression,

sharing the same hypothetical function for prediction. The cost function of Linear Regression is represented by:

$$J = \sum_{i=1}^m (y^{(i)} - h(x^{(i)}))^2 \quad (24)$$

Here

- m is the total number of training examples in the dataset.
- $h(x^{(i)})$ represents the hypothetical function for prediction.
- $y^{(i)}$ represents the value of target variable for $i^{(th)}$ training example.

The cost function of Lasso regression is modified as follows by adding the L1 penalty term:

$$J = \sum_{i=1}^m (y^{(i)} - h(x^{(i)}))^2 + \lambda \sum_{j=1}^n |w_j| \quad (25)$$

Where:

- w_j represents the weight for the $j^{(th)}$ feature.
- n is the number of features in the dataset.
- λ is the regularization strength.

Lasso is a standard tool in supervised learning pipelines. It acts as both a predictor and a built-in feature selector, which simplifies models and speeds downstream tasks. Typical uses include high-dimensional biomedical inference (selecting genomic markers), text and n-gram models (sequence of n words), and engineering systems where many measured controls compete to explain a response (Bühlmann & van de Geer, 2011; Hastie et al., 2009).

2.8.3. Ridge Regression (RidgeCV)

Hoerl and Kennard (1970a, 1970b) introduced ridge regression to address the instability of ordinary least squares in the presence of highly correlated predictors. The method modifies the regression objective by adding an L2 regularization term, which constrains the size of the coefficients. This adjustment lowers variance and enhances the reliability of predictions when multicollinearity is a concern.

$$\beta^{\lambda} = (X^T X + \lambda I)^{-1} X^T y \quad (26)$$

Here X is the design matrix, y is the response vector, and $\lambda \geq 0$ is the regularization parameter, controlling the amount of shrinkage. When $\lambda=0$, the estimator reduces to ordinary least squares; as λ increases, coefficients are pulled closer to zero, lowering variance at the cost of introducing bias (Hoerl, 2020). The cost function of Ridge regression is modified as follows by adding L2 penalty term:

$$J = \sum_{i=1}^m (y^{(i)} - h(x^{(i)}))^2 + \lambda \sum_{j=1}^n w_j^2 \quad (27)$$

In contemporary machine learning applications, ridge regression is often employed in settings with several predictors or where strong correlations exist among variables. The regularization term shrinks coefficient values without excluding predictors, which helps balance their relative effects when inputs are correlated (Hastie et al., 2009). This feature is particularly relevant in engineering studies, where parameters such as cutting depth, rotational speed, and table speed are typically interdependent. By stabilizing coefficient estimates, ridge regression provides more consistent predictions for outcomes like vibration and specific energy than ordinary least squares.

OLS, Lasso, and Ridge regressions are all linear models with the same prediction form. They differ only in how they penalize coefficients during estimation. Ordinary Least Squares (OLS) minimizes squared errors without a penalty, so it uses every predictor. It can indicate low bias but high variance when predictors are correlated or the sample is small. Ridge regression adds an L2 penalty. It shrinks coefficients toward zero, lowers variance, and handles multicollinearity, but it does not set coefficients exactly to zero. Lasso adds an L1 penalty. It also shrinks coefficients and can set some exactly to zero, which performs embedded feature selection. When predictors are strongly correlated, Lasso may pick one and drop the rest; elastic net is a practical compromise if you need both stability and sparsity. Effective practice is simple. Standardize predictors before using Ridge or Lasso. Choose the penalty strength (λ) by cross-validation. Inspect residuals and outliers because all three optimize squared error and remain sensitive to extreme points. With a single predictor, ridge and lasso lines often look almost identical to OLS unless λ is made large; this outcome is expected. Figure 2.29 shows these differences clearly: OLS follows the data without penalty; Ridge is shallower from L2 shrinkage; Lasso is also shrunk and would flatten at sufficiently large λ in the one-dimensional case.

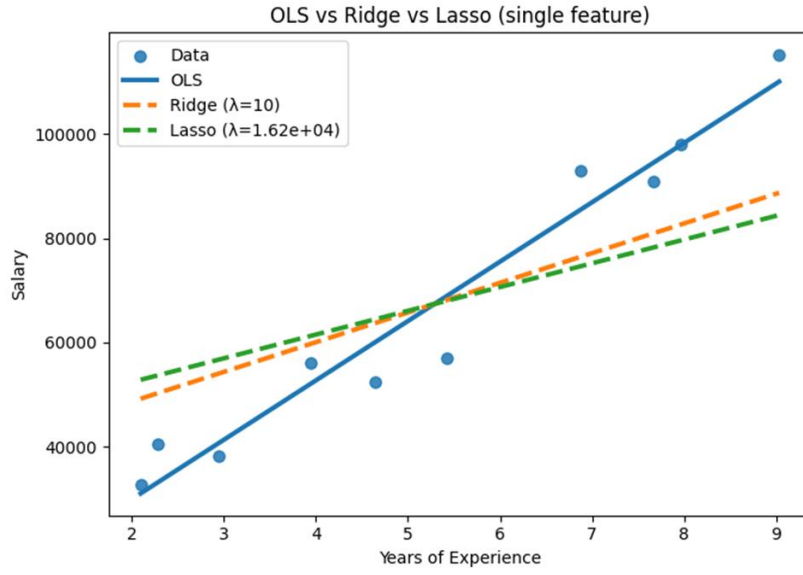


Figure 2.32. Differences between OLS, Ridge, and Lasso (single predictor)

2.8.4. Random Forest Regression

Random forest regression is an ensemble of decision trees trained on bootstrap samples, with each split chosen from a random subset of predictors. Predictions are the average of the trees. This simple design reduces variance and captures non-linear effects and interactions without heavy modeling choices (Breiman, 1996, 2001). In practice it works well “out of the box” on mixed, noisy, or moderately high-dimensional data and needs little preprocessing standardization is not required (Hastie et al., 2009). At its core, it is an ensemble of decision-tree models trained on resampled data; this “forest” structure is what defines it as an ensemble method (Y. Liu et al., 2012). Each tree is a standard decision tree, and the forest combines their outputs, by majority vote for classification and by averaging for regression, to produce a final prediction (Manzali et al., 2023). Because random forests rely on bootstrap sampling of rows and random subsampling of features at each split, the procedure is nondeterministic. (Jurczuk et al., 2023).

For regression, the prediction is the average of T trees (Eq. 28). Nodes are split by minimizing mean-squared error (Eq. 29). For classification, common impurities are Gini or entropy (Eqs. 30–31). Trees are grown on bootstrap samples, and each split considers a random subset of features; these two sources of randomness reduce correlation among trees. The overall training–prediction pipeline is shown in figure 2.33.

$$\hat{f}_{RF}(x) = \frac{1}{T} \sum_{t=1}^T \hat{f}^{(t)}(x) \quad (28)$$

$$MSE = \frac{1}{N} \sum_{i=1}^N (f_i - y_i)^2 \quad (29)$$

$$Gini = 1 - \sum_{c=1}^C p_c^2 \quad (30)$$

$$Entropy = - \sum_{c=1}^C p_c \log_2 p_c \quad (31)$$

Here:

- N - the number of data points you're averaging over.
- f_i - the model's prediction for the i -th sample.
- y_i - the observed (true) target value for the i -th sample.
- C - the number of classes (distinct labels) used when computing Gini or entropy in classification
- p_c - the proportion of class c among the samples in the set you're evaluating

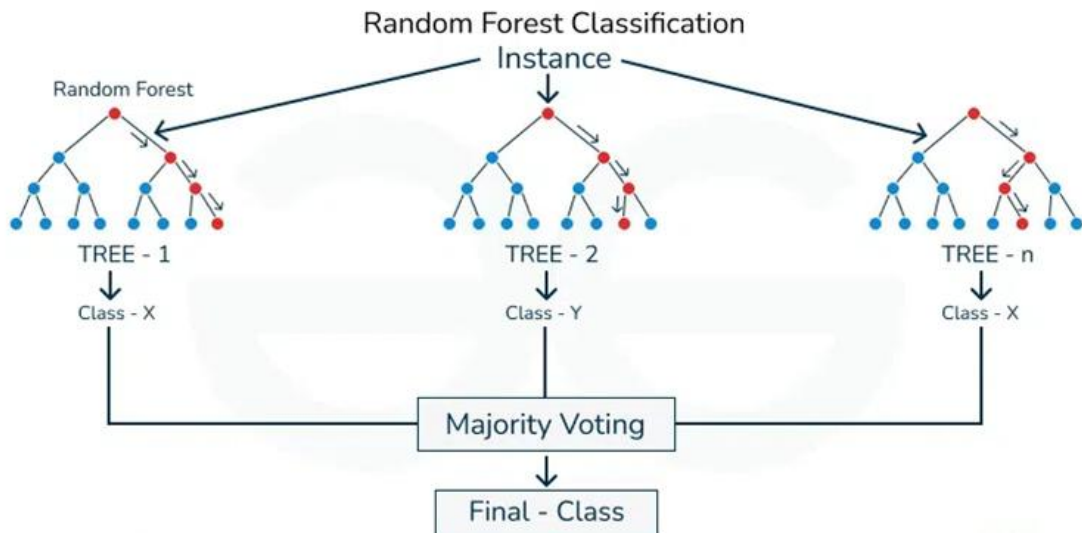


Figure 2.33. Random forest classification workflow (GeeksforGeeks, 2025)

RFs that are built as ensembles of randomized decision trees perform reliably on both classification and regression tasks (Akhiat et al., 2021). They tend to reach high accuracy,

tolerate noisy inputs, and scale to high-dimensional predictors (Chourib et al., 2022). The primary cost is computational. Building and storing a large ensemble needs large memory and time, and the resulting forest can contain redundant trees that add little to accuracy (H. Wang, 2023). They can also be used in text and image classification and in climate prediction, where they handle large, complex datasets and generalize to previously unseen conditions (Ding et al., 2025; Manzali & Elfar, 2023).

2.8.5. Gradient Boosting

Gradient boosting is a stage-wise ensemble supervised method that builds a strong predictor by adding many weak learners, most often shallow regression trees. At each step, the algorithm moves in the direction that reduces a chosen loss. This view links boosting to gradient descent in function space and explains why it works well across many tasks. The approach traces back to work that formalized boosting as additive modeling and gradient descent, and it is now a standard tool in supervised learning (Bühlmann & Hothorn, 2007; Friedman, 2001; Hastie et al., 2009). In the functional-gradient view, gradient boosting performs gradient descent in function space by fitting each new weak learner to the negative gradient (pseudo-residuals) of the loss with respect to the current model's predictions (Friedman, 2001).

Formally, let the differentiable empirical loss be:

$$L(f) = \sum_{i=1}^n l(y_i, f(x_i)). \quad (32)$$

Initialize

$$f_0(x) = \underset{c}{\operatorname{argmin}} \sum_{i=1}^n l(y_i, c). \quad (33)$$

For $m=1, \dots, M$:

1. Pseudo-residuals

$$r_{im} = - \left. \frac{\partial l(y_i, f(x_i))}{\partial f(x_i)} \right|_{f=f_{m-1}} \quad (34)$$

2. Fit base learner

$$h_m \leftarrow \text{fit to } \{x_i, r_{im}\}_{i=1}^n \quad (35)$$

3. Line search

$$\rho_m = \underset{c}{\operatorname{argmin}} \sum_{i=1}^n l(y_i, f_{m-1}(x_i) + \rho h_m(x_i)) \quad (36)$$

4. Additive update with shrinkage

$$f_m = f_{m-1}(x) + \nu \rho_m h_m(x), \quad 0 < \nu \leq 1 \quad (37)$$

The stage-wise procedure is illustrated in Figure 2.34: at each iteration, a shallow tree is fit to the current pseudo-residuals and adds (with shrinkage) to the ensemble.

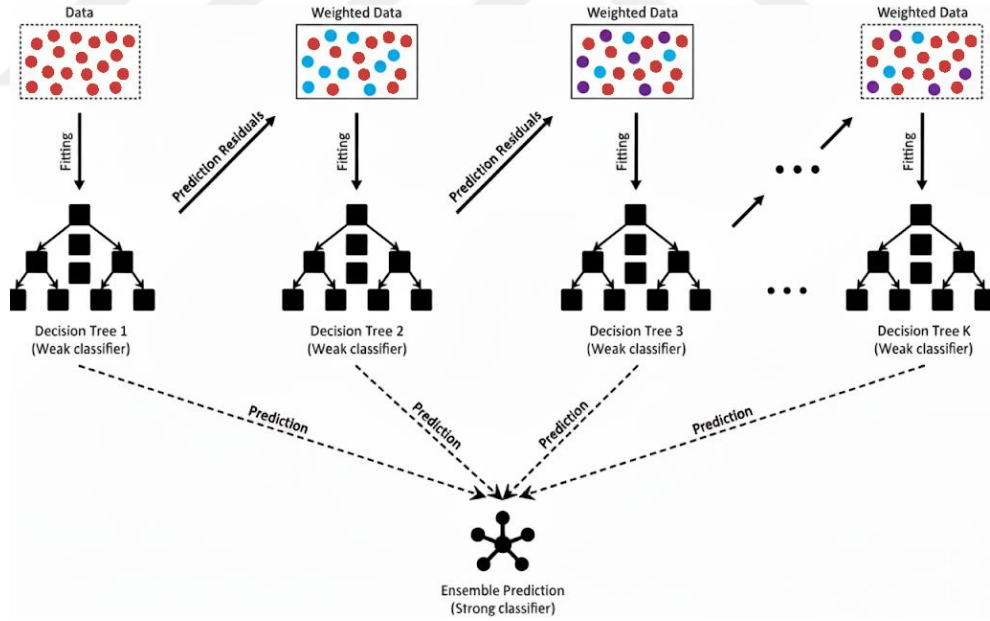


Figure 2.34. The architecture of gradient boosting decision tree, adapted from (Deng et al., 2021)

Regularization is important. Small learning rates (any but shallow trees) and early stopping help control overfitting. Subsampling the training data at each iteration (without replacement) yields “stochastic gradient boosting,” which lowers variance and often

improves test error while speeding training. These choices make the method robust on noisy, high-dimensional, and tabular data (Friedman, 2002; Hastie et al., 2009). Modern libraries implement these ideas on a scale. XGBoost adds second-order optimization, sparsity-aware splits, and a weighted-quantile sketch for rapid tree building. LightGBM introduces histogram-based splitting and leaf-wise growth with depth limits for better results. CatBoost uses “ordered boosting” and target-aware encodings to handle categorical features while reducing prediction shift (Chen & Guestrin, 2016; Ke et al., 2017; Prokhorenkova et al., 2019). In practice, gradient boosting has a better performance when relationships are nonlinear and include interactions. It needs little feature scaling, handles mixed data types, and provides feature importance to aid interpretation. The trade-offs are longer training times and more hyperparameters, but careful tuning of the learning rate, number of trees, tree depth, and subsampling usually pays off with strong, reliable predictions (Friedman, 2001; Hastie et al., 2009).

2.9. Particle Swarm Optimization

Particle Swarm Optimization (PSO) was introduced in 1995 as a simple population-based algorithm for numerical optimization (Eberhart & Kennedy, 1995). Its design draws on sociobiology, where groups such as fish schools or bird flocks gain by sharing what each member learns. When one bird finds food, the flock can use that discovery and benefit. PSO turns this idea into computation. Each “particle” remembers its own best position and learns from the swarm’s best, so information spreads and the search improve over (Eberhart & Kennedy, 1995; Poli et al., 2007).

PSO also connects to evolutionary computation and artificial life. Millonas (1994) outlined five traits for cooperative swarm systems which are, proximity, quality, diverse response, stability, and adaptability. The fourth and fifth principles are opposing tendencies, yet together with the other three they define the core of artificial life and guiding swarm systems. In PSO, particles update position and velocity in response to the environment, satisfying proximity and quality. They range widely through the search space while preserving enough stability to adapt as conditions change, thereby fulfilling all five principles (Millonas, 1993; Poli et al., 2007).

Within the PSO methodology, the positional attribute represented by x_i , of each particle serves as a solution to the designated problem, with the optimal solution denoting

the source of this solution. Each particle maintains an individual memory to store its previously attained best position and the globally optimal position within the entire population to which it belongs. Through the information archived in its memory and its velocity, denoted as v_i , each particle can iteratively update its position, x_i^{t+1} , as illustrated in Fig 2.35. To get the optimum solution, each particle moves towards its prior personal best position (p_{best}) and the global best position (p_{best}) in the swarm (X. Zhang et al., 2018).

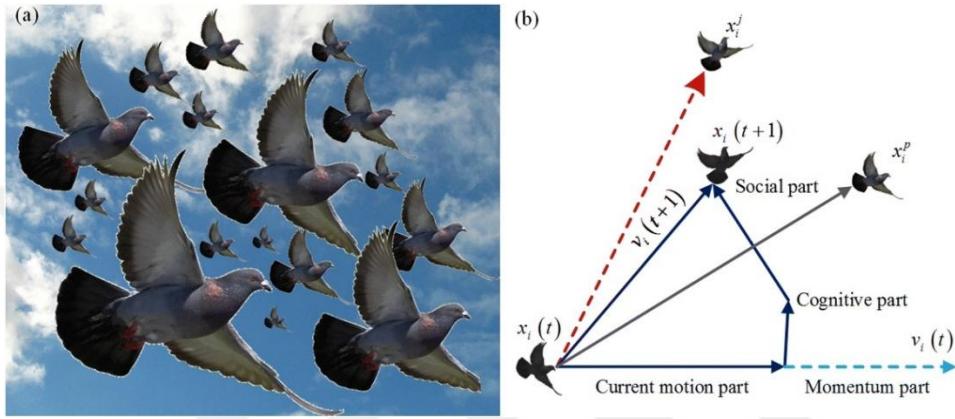


Figure 2.35. Illustration of PSO method, where each particle updates its position (x_i^{t+1}) based on individual and global best positions and velocity (v_i) (adapted from(Akbasli, 2024)).

Assuming a minimization problem, one have:

$$p_{best_i}^t = x_i^* \mid f(x_i^*) = \min_{k=1,2,3,\dots,t}(\{f(x_i^k)\}), \quad (38)$$

Where $i \in \{1,2, \dots, N\}$, and

$$g_{best}^t = x_*^t \mid f(x_*^t) = \min_{\substack{i=1,2,3,\dots,N \\ k=1,2,\dots,t}}(\{f(x_i^k)\}) \quad (39)$$

The following equations update, at each current iteration $t + 1$, the velocity v and position x of each particle i as:

$$v_i^{t+1} = wv_i^t + c_1r_1(p_{best_i}^t - x_i^t) + c_2r_2(g_{best}^t - x_i^t), \quad (40)$$

$$x_i^{t+1} = x_i^t + V_i^{t+1}, \quad (41)$$

Here V represents the velocity vector, w is the inertia weight utilized to balance the local exploitation and global exploration, r_1 and r_2 are random vectors uniformly distributed within the range $[0,1]^D$ (D being the search space dimensionality or the size of the problem at hand), and c_1 and c_2 called acceleration coefficients, are positive constants. A compact procedural view is initializing particles and velocities; loop over evaluate $\rightarrow$ update personal bests $\rightarrow$ update global best $\rightarrow$ update velocities and positions; then return the best solution (Elsheikh & Abd Elaziz, 2019). The flowchart to demonstrate the flux of PSO is depicted in Fig 2.36

In short, PSO offers a transparent balance of exploration and exploitation with few hyperparameters and broad applicability. It has become a dependable tool in engineering design, control, and machine learning tasks such as feature selection and hyperparameter tuning, where black-box objectives and noisy evaluations are common (Kennedy, 2011; Poli et al., 2007).

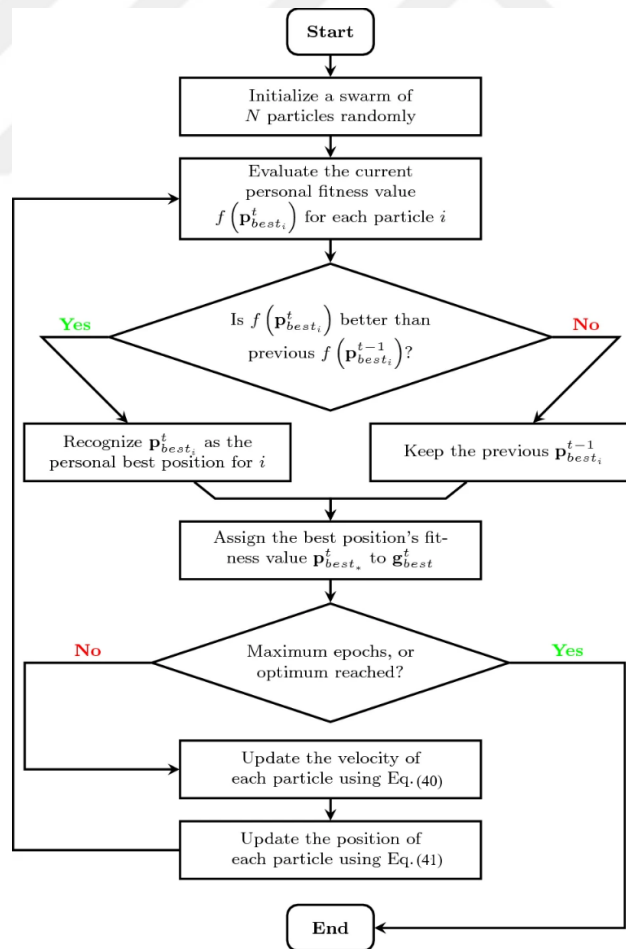


Figure 2.36. The flowchart of PSO (adapted from (Gad, 2022))

2.10. Model Performance Evaluation

The performance of the machine-learning models was assessed using four evaluation metrics: the coefficient of determination (R^2), mean absolute error (MAE), root mean squared error (RMSE) and mean absolute percentage error (MAPE). These metrics collectively quantify the agreement between predicted and measured values and capture both absolute and relative error behavior.

2.10.1. Coefficient of Determination (R^2)

The Coefficient of Determination, R^2 , measures the proportion of the variance in the dependent variable that is predictable from the independent variables. It ranges from $-\infty$ to 1, where a value closer to 1 indicates a better fit, meaning the model accurately captures the underlying trends of the data. A value of 1 signifies a perfect prediction. In high-stakes fields (e.g., aerospace, critical infrastructure), an R^2 above 0.9 may be necessary. In fields with high inherent data variability (e.g., social or ecological systems), a value between 0.5 and 0.8 might be considered successful and highly informative.

The formulation for R^2 is defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (42)$$

Where y_i is the actual value, $\hat{y}_i$ is the predicted value, $\bar{y}$ is the actual value, and n is the number of data points.

2.10.2. Mean Absolute Error (MAE)

The Mean Absolute Error (MAE) represents the average magnitude of the errors in a set of predictions, without considering their direction. It is the average of the absolute differences between the actual and predicted values. Because the errors are averaged in absolute terms, MAE gives equal weight to all errors and is a robust metric that is less sensitive to outliers compared to RMSE. The units of MAE are the same as the target variable. A lower MAE indicates better model performance.

The formulation for MAE is given by:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (43)$$

2.10.3. Root Mean Squared Error (RMSE)

The Root Mean Squared Error (RMSE) is the square root of the average of squared differences between the predicted values and the actual values. Squaring the errors before averaging means that large errors (outliers) are penalized much more heavily than smaller errors. This characteristic makes RMSE a useful metric when large errors are particularly undesirable or critical. Like MAE, its units are the same as the target variable. A lower RMSE indicates better model performance.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (44)$$

The acceptable threshold for MAE and RMSE must be evaluated relative to the scale and mean of the predicted variable. For a model to be deemed acceptable, the error magnitude (MAE or RMSE) should typically represent a small percentage (e.g., 5-15%) of the actual values' range or mean. A significantly higher RMSE compared to MAE suggests that the model is making a few severe errors (outliers).

2.10.4. Mean Absolute Percentage Error (MAPE)

The Mean Absolute Percentage Error (MAPE) expresses the prediction accuracy as a percentage. It is calculated as the average of the absolute percentage errors. MAPE is especially useful for understanding performance in a relative sense, making it easily interpretable across different datasets or models, regardless of the scale of the target variable. However, it is undefined if any actual value (y_i) is zero. A lower MAPE percentage indicates higher accuracy. The formulation for MAPE is:

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100\% \quad (45)$$

MAPE is often the most intuitive metric for relative performance. An acceptable MAPE often falls below 20%, with values under 10% generally considered highly accurate for most forecasting and prediction tasks. However, in applications demanding extreme precision, such as quality control or financial modeling, the acceptable limit may be strictly set below 5%.



3. RESULTS AND DISCUSSION

3.1. Mechanical Properties

Figure 3.1 and Tables 3.1–3.3 summarize compressive strength (UCS), Brazilian tensile strength (BTS), and the 10–60% secant modulus of elasticity (E) for mixes 350-3, 250-2, and 175. Differences between uniaxial compressive strength (UCS) measured on molded cubes and on drilled cores arise because the specimens are not mechanically equivalent in geometry, boundary conditions, or history. Cubes are squat and experience greater platen-induced lateral restraint, whereas core strength depends on the length-to-diameter ratio; without correction to $L/D \approx 2$, cores can be biased. Moisture state and test age further shift UCS (drier/older higher; saturated/younger lower), and cores sample in-situ heterogeneity and casting anisotropy that cubes may not capture. In our data, core UCS surpassed cube UCS for mixes 350-3 and 250-2 but was lower than 175, consistent with geometry and conditioning effects outweighing coring damage in the stronger mixes and the opposite in the weakest.

Table 3.1. UCS results from cubic concrete specimens

Mix ID	Specimen No	UCS (MPa)	Average UCS (MPa)	Standard Deviation
350-3	C-1	21.65	21.21	1.06
	C-2	21.99		
	C-3	20.00		
250-2	C-1	12.64	13.69	0.72
	C-2	14.04		
	C-3	14.25		
	C-4	13.82		
175	C-1	7.17	6.78	1.09
	C-2	7.96		
	C-3	5.37		
	C-4	6.61		

Table 3.2. BTS results from all 3 mixes

Mix ID	Specimen No	BTS (MPa)	Average BTS (MPa)	Standard Deviation
350-3 (Mix 1)	C2B	2.71	3.09	0.37
	C3B	2.74		
	C4B	3.00		
	C5B	3.06		
	C6B	3.20		
	C7B	3.81		
	C8B	3.13		
250 (Mix 2)	C1B	1.91	2.15	0.34
	C2B	2.13		
	C3B	1.93		
	C4B	2.63		
175 (Mix 3)	C1B	1.90	1.86	0.06
	C2B	1.82		

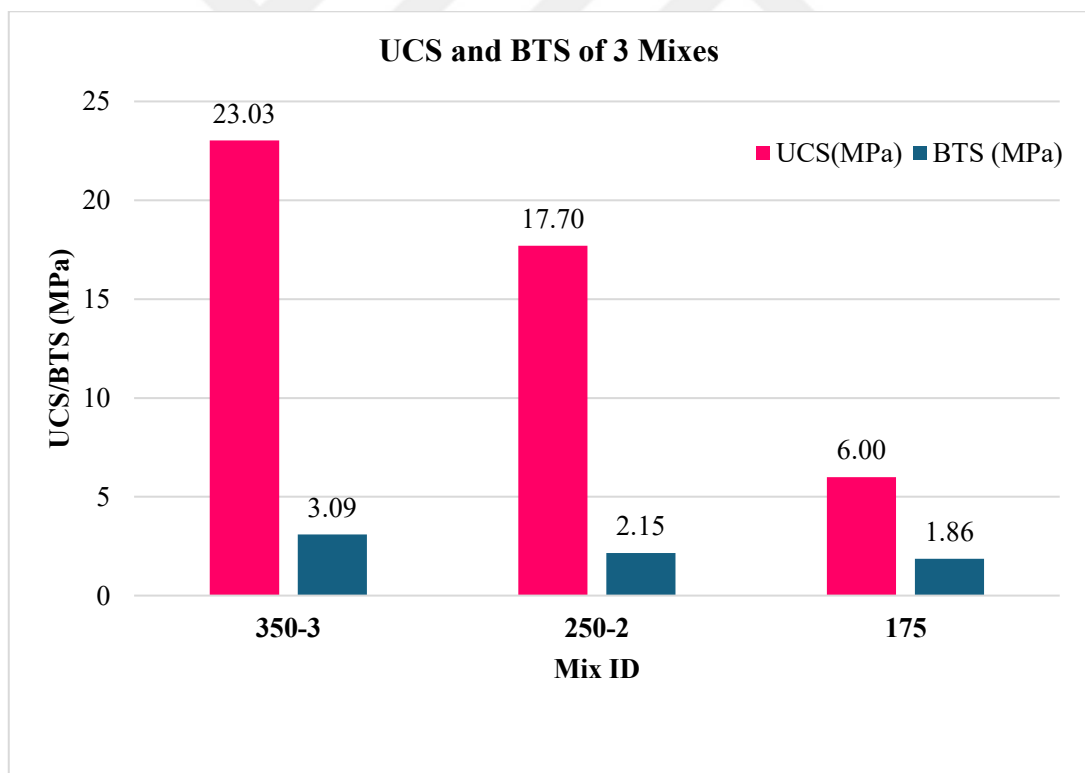


Figure 3.1. Results of BTC and UCS of three different mixes

Table 3.3. Modulus of elasticity (10-60%) results from core samples

Mix ID	Specimen No	Density (g/cm ³)	UCS(MPa)	E (GPa) (10-60%)	UCS Average (MPa)	E Average (GPa)
350-3 (Mix 1)	C1	2.27	26.53	11.98	23.03	14.31
	C2	2.32	16.605	13.025		
	C3	2.26	24.369	14.536		
	C4	2.27	24.489	15.295		
	C5	2.26	23.155	16.734		
250-2 (Mix 2)	C1	2.28	18.529	10.947	17.70	11.92
	C2	2.26	16.23	12.341		
	C3	2.26	17.507	11.272		
	C4	2.27	18.535	13.138		
175 (Mix 3)	C1	2.11	6	6.7	6.00	6.70

3.2. Dominant Machine-Frame Vibration Axis During Cutting

Appendix 1, Tables 1–9 report the cleansed dataset from the rock-cutting experiments for three mixes (Mix1 to Mix3) and three repetitions (Rep 1 to Rep 3). In these data set, the effective kerf width produced by a conical pick is computed as:

$$W_{eff} = 2d \tan\left(\frac{\theta}{2}\right) \quad (46)$$

Here, d is the real cutting depth and $\frac{\theta}{2}$ is pick half-cone angle. In this study which is $\frac{\theta}{2} = 40$. Specific energy is defined as power per unit material removal rate:

$$SE\left(\frac{Mj}{m^3}\right) = \frac{P(w)}{d(m) \times W_{eff}(m) \times v_t(m/s)} \quad (47)$$

Where p is power during cutting, v_t is table speed, also called traverse speed, d is the cutting depth, and W_{eff} is given above.

Across all three repetitions, the X-axis (cutting direction) shows the largest vibration response in the data. As summarized in Table 3.4, the mean X-axis vibration in Mix-1 is 178.59 mm/s, compared with 95.22 mm/s in Z, a difference of roughly 83 mm/s. A similar pattern appears in Mix-2, where the mean X-axis level is 157.72 mm/s versus 91.87 mm/s in Z ($\Delta \approx 66$ mm/s). Mix-3 shows the same trend, though at lower overall magnitudes: mean X-axis vibration is 49.10 mm/s compared with 26.28 mm/s in Z ($\Delta \approx 23$ mm/s). Higher-percentile and peak values reinforce this observation. In Mix-1, the 95th-percentile vibration on the X-axis (265.22 mm/s) exceeds that on Z (144.23 mm/s) by about 121 mm/s; for Mix-2 the difference is approximately 131 mm/s (252.75 vs 121.73 mm/s), and for Mix-3 about 31 mm/s (81.61 vs 50.23 mm/s). Maximum recorded vibrations follow the same pattern: X exceeds Z by roughly 113 mm/s in Mix-1 (266.80 vs 153.64 mm/s), 123 mm/s in Mix-2 (266.85 vs 144.01 mm/s), and 49 mm/s in Mix-3 (109.23 vs 60.63 mm/s). Overall, the consistently higher X-axis values across mean, high-percentile, and peak metrics indicate that vibration is greatest in the cutting direction for all mixes.

Table 3.4. Vibration-X summary for Mix-1 to Mix-3, with Z included for comparison. The X axis is aligned with the cutting direction.

Mix	Axis	Mean	STD	Median	q95	MAX-Vibration
Mix-1	X	178.59	55.91	177.75	265.22	266.80
Mix-2	X	157.72	50.31	141.83	252.75	266.85
Mix-3	X	49.10	17.57	44.88	81.61	109.23
Mix-1	Z	95.22	27.42	91.36	144.23	153.64
Mix-2	Z	91.87	18.90	90.63	121.73	144.01
Mix-3	Z	26.28	10.90	23.56	50.23	60.63

Figures 3.2–13 corroborate Table 3.4. For every mix and repetition, X exceeds Z across Tests 1 to 27, with larger peaks and plateaus during active cutting. The intra-test traces show synchronous variation of both axes with a persistent amplitude offset favoring X, indicating directional excitation aligned with the cutting direction. Absolute levels are highest in Mix 2 and lowest in Mix 3, while the X over Z ordering is preserved. For each mix, one representative test illustrates the cutting window and X–Z vibration response. Mix 1 (Rep 3–Test 18) shows rapid onset and high X-direction peaks (>250 mm/s), with Z remaining lower but following similar trends. Mix 2 (Rep 2–Test 18) exhibits the same pattern with

slightly higher magnitudes, reflecting more demanding cutting. Mix 3 (Rep 1–Test 6) shows substantially lower amplitudes in both axes while retaining X-dominance.

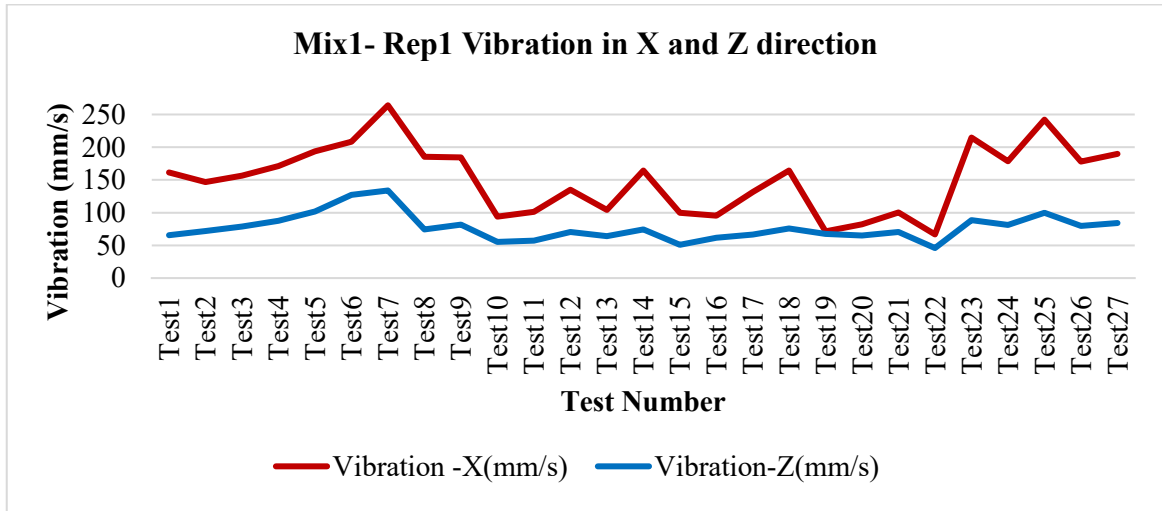


Figure 3.2. Mix 1-repeat 1: vibrations in the X and Z directions.

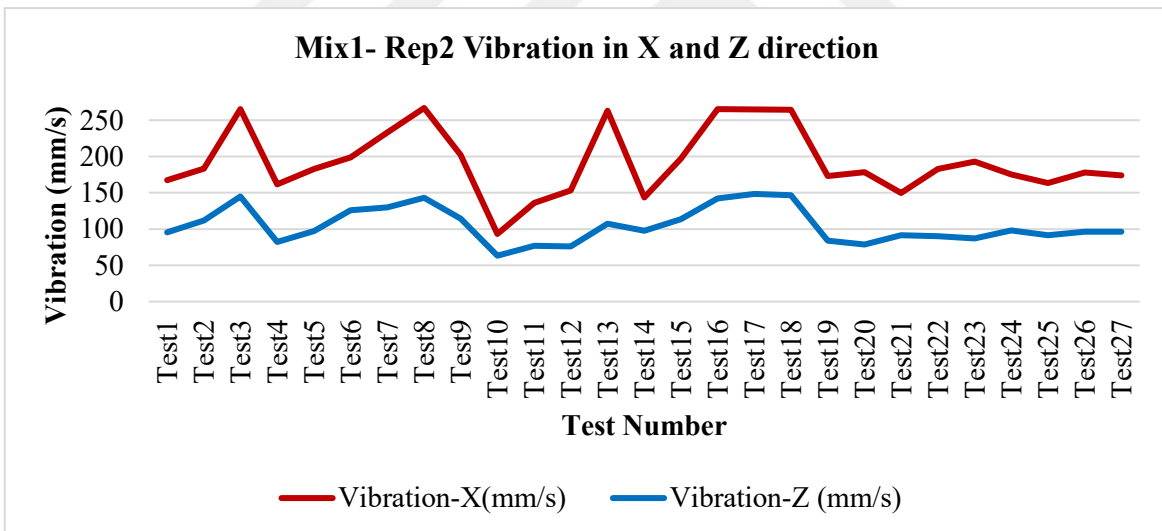


Figure 3.3. Mix 1-repeat 2: vibrations in the X and Z directions.

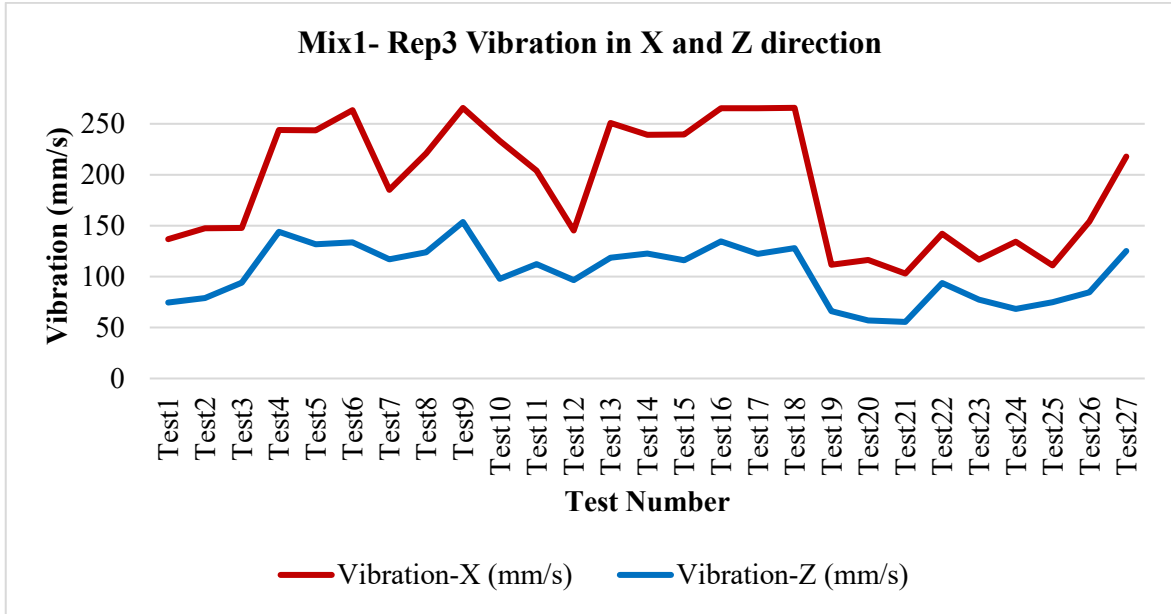


Figure 3.4. Mix 1-repeat 3: vibrations in the X and Z directions.

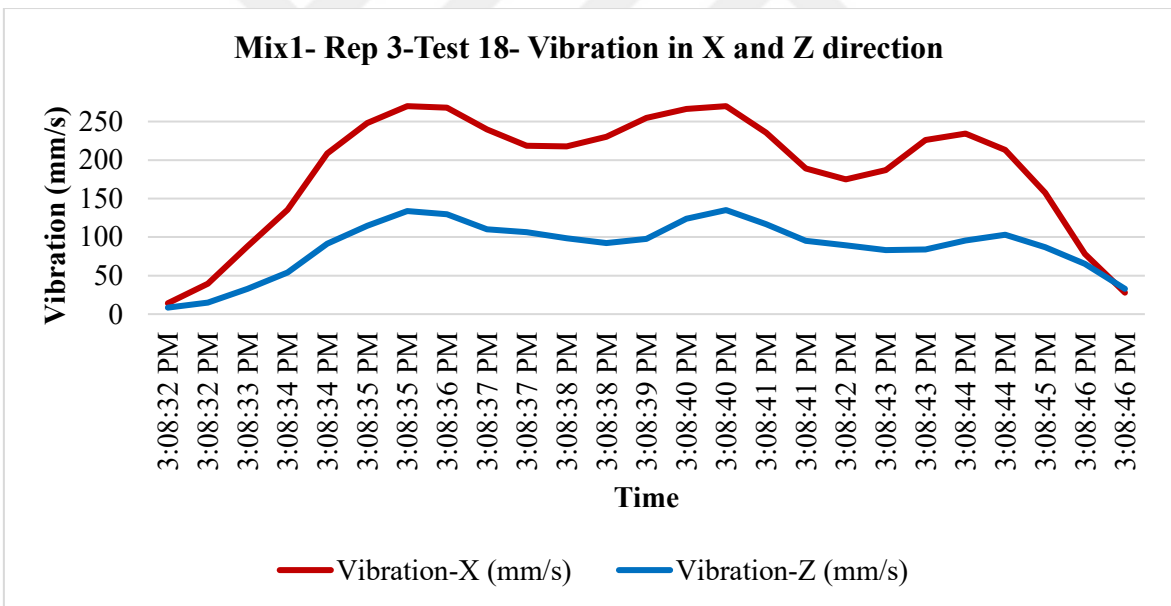


Figure 3.5. Vibrations in X and Z directions over time for Mix-1, repeat 3, Test 18.

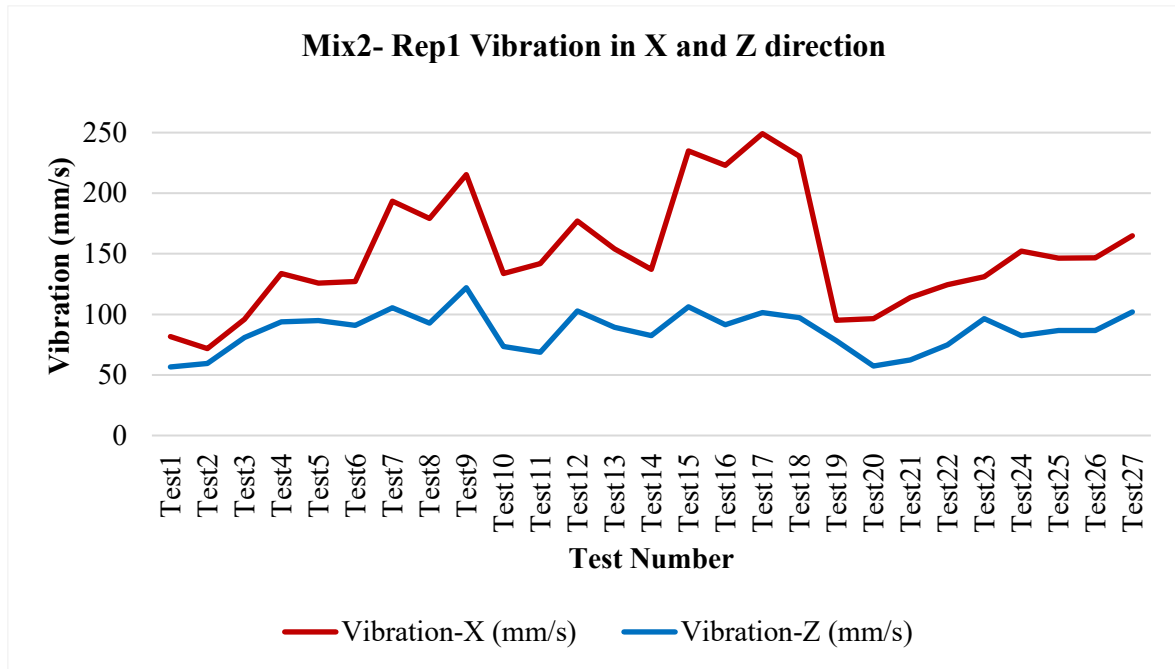


Figure 3.6. Mix 2-repeat 1: vibrations in the X and Z directions.

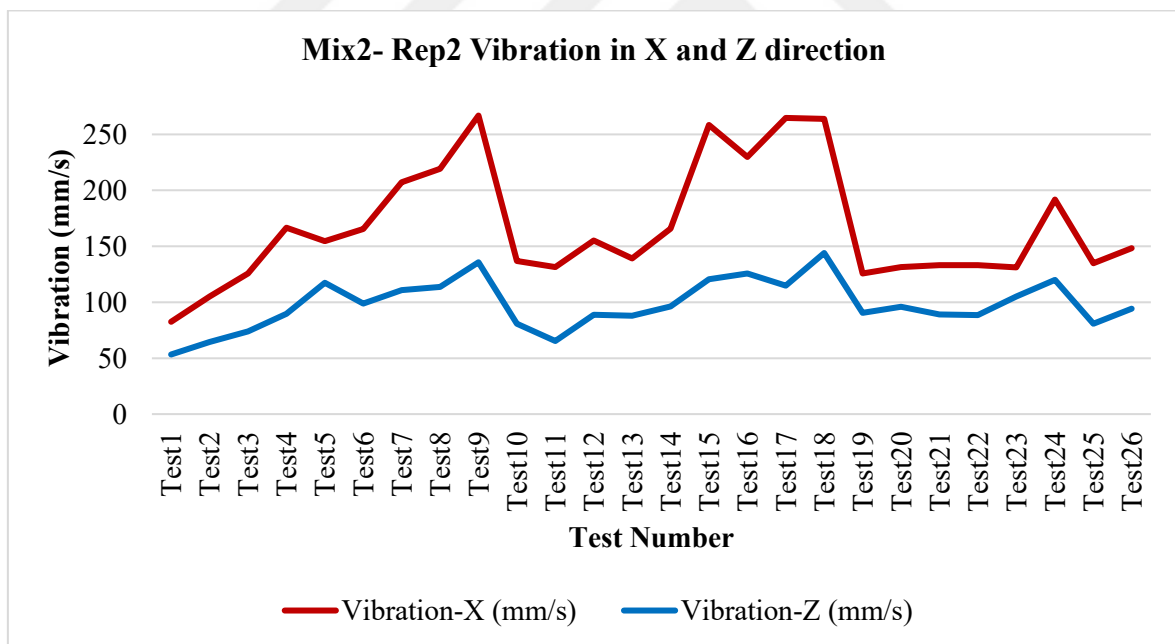


Figure 3.7. Mix 2-repeat 2: vibrations in the X and Z directions.

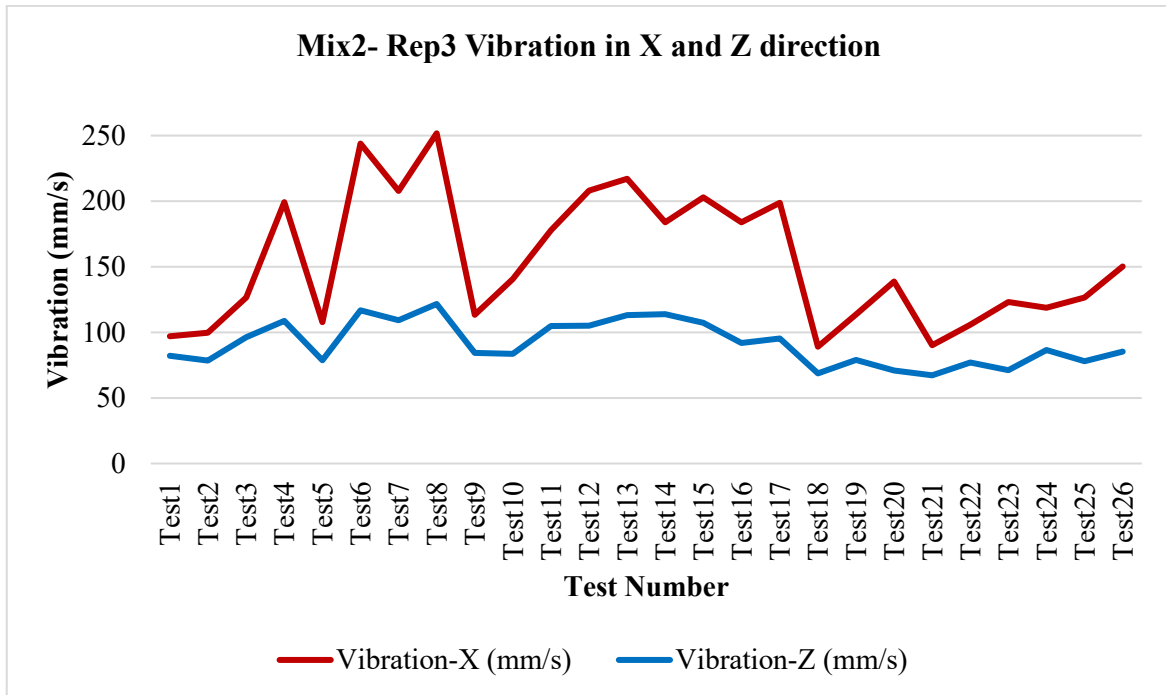


Figure 3.8. Mix 2-repeat 3: vibrations in the X and Z directions.

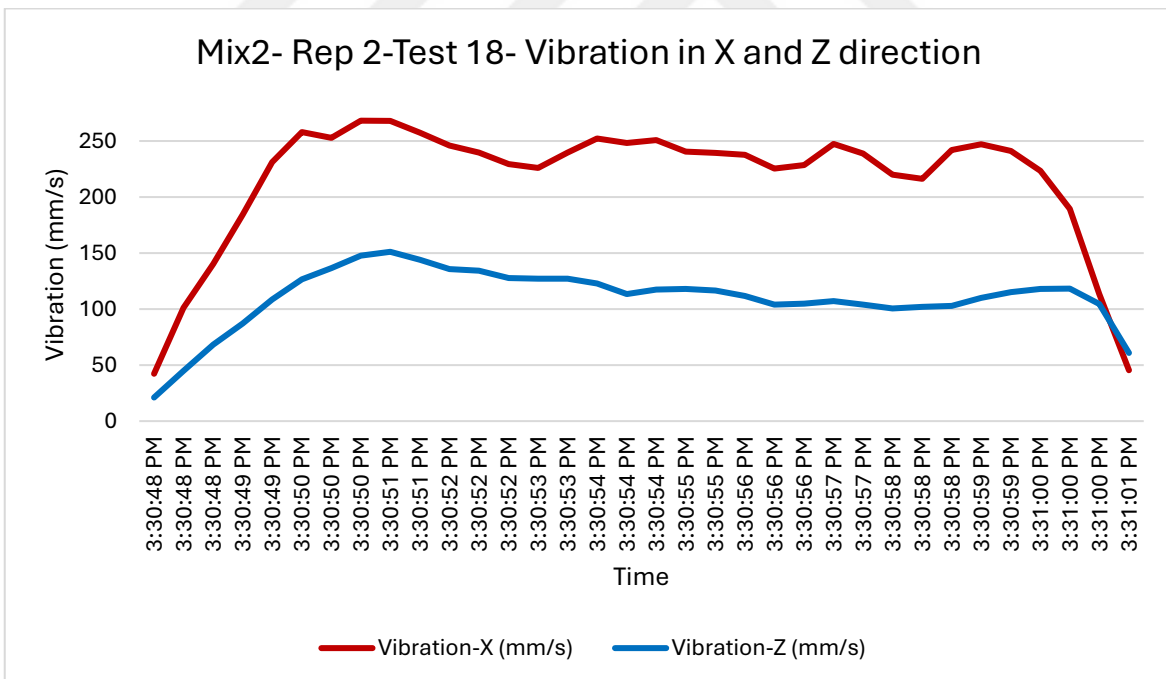


Figure 3.9. Vibrations in X and Z directions over time for Mix 2, repeat 2 Test 18.

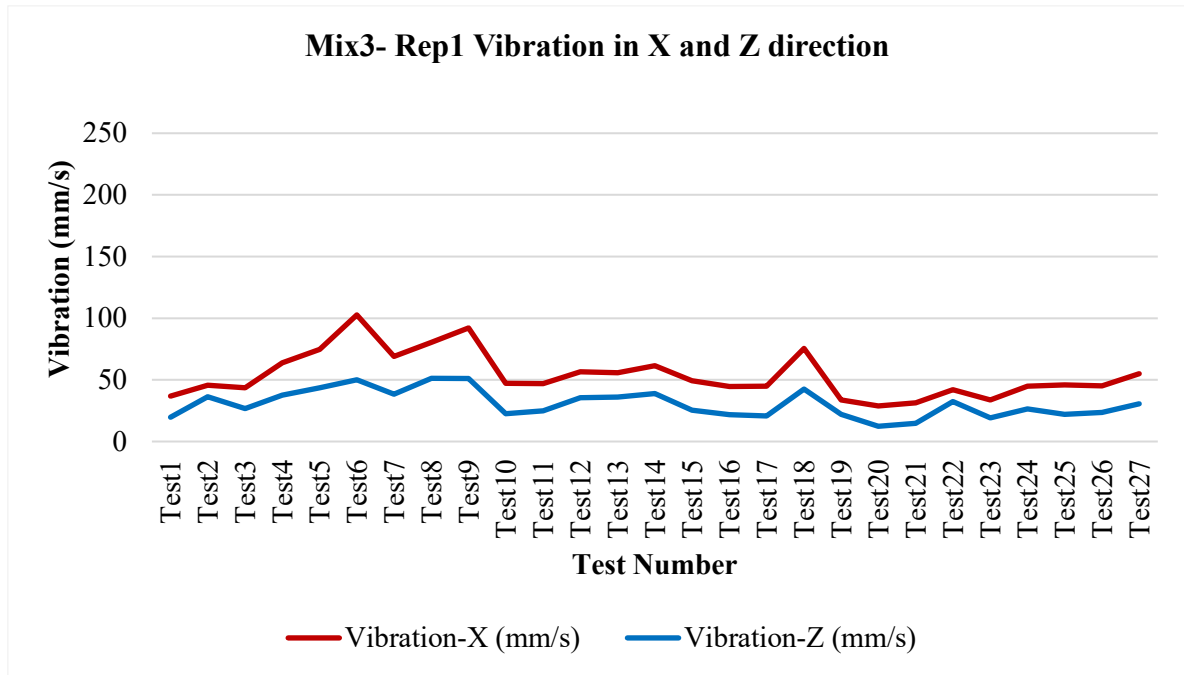


Figure 3.10. Mix 3-repeat 1: vibrations in the X and Z directions.

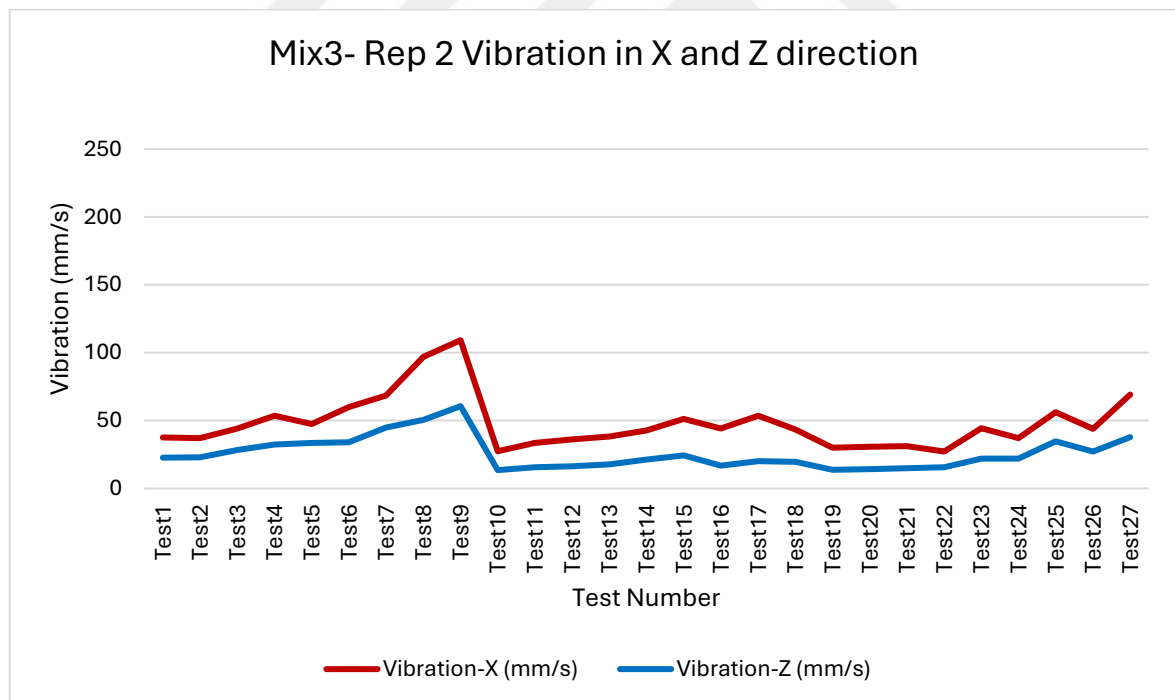


Figure 3.11. Mix 3-repeat 2: vibrations in the X and Z directions.

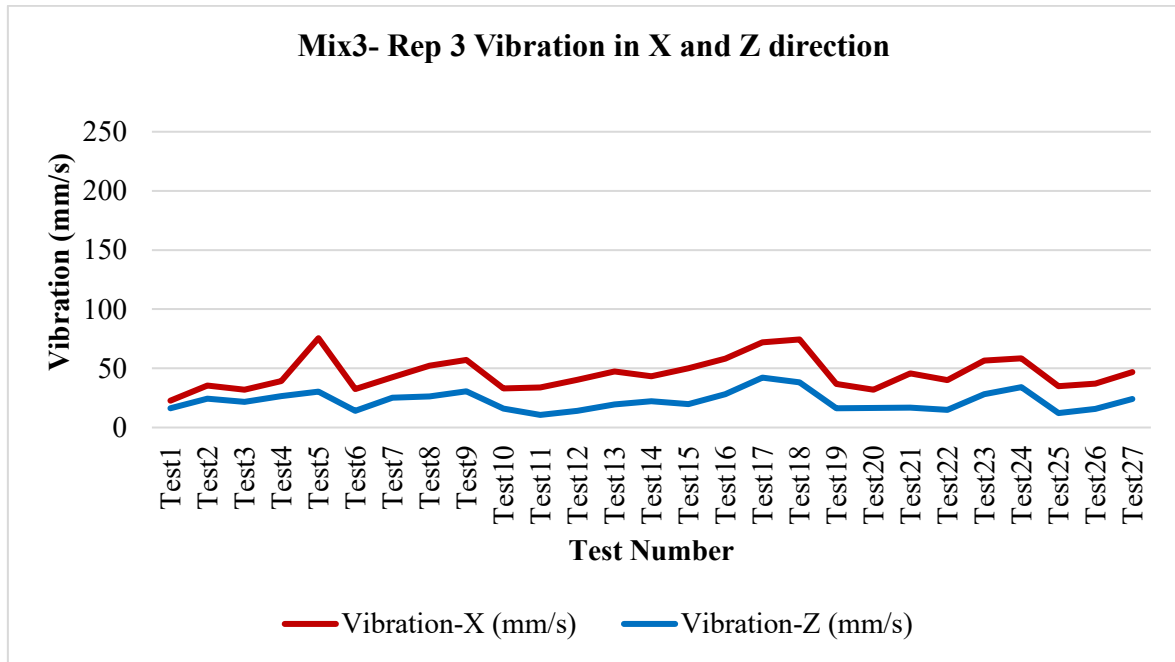


Figure 3.12. Mix 3-repeat 3: vibrations in the X and Z directions.

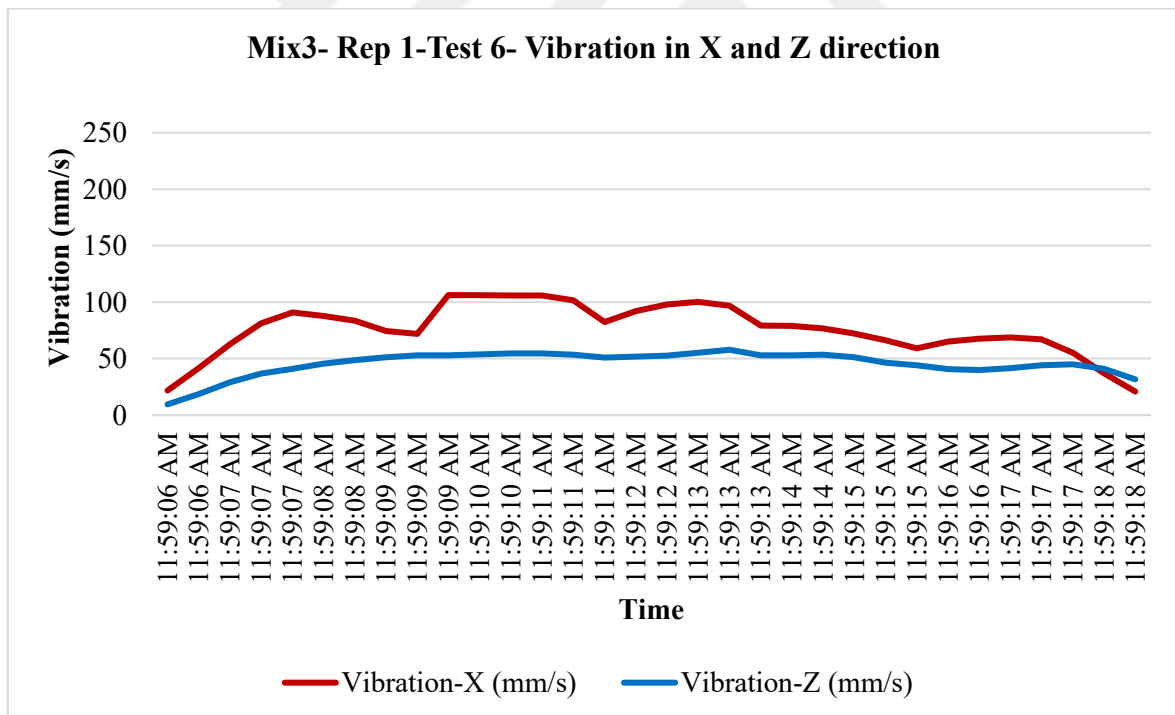


Figure 3.13. Vibrations in X and Z directions over time for Mix 3, repeat 1 Test 6.

These indicate that the dominant excitation acting on the machine–tool structure during cutting is aligned with the direction of chip formation and pick motion. This observation is mechanically consistent with how roadheader picks load the system. The rock reaction on each pick is commonly decomposed into cutting (tangential), normal, and side

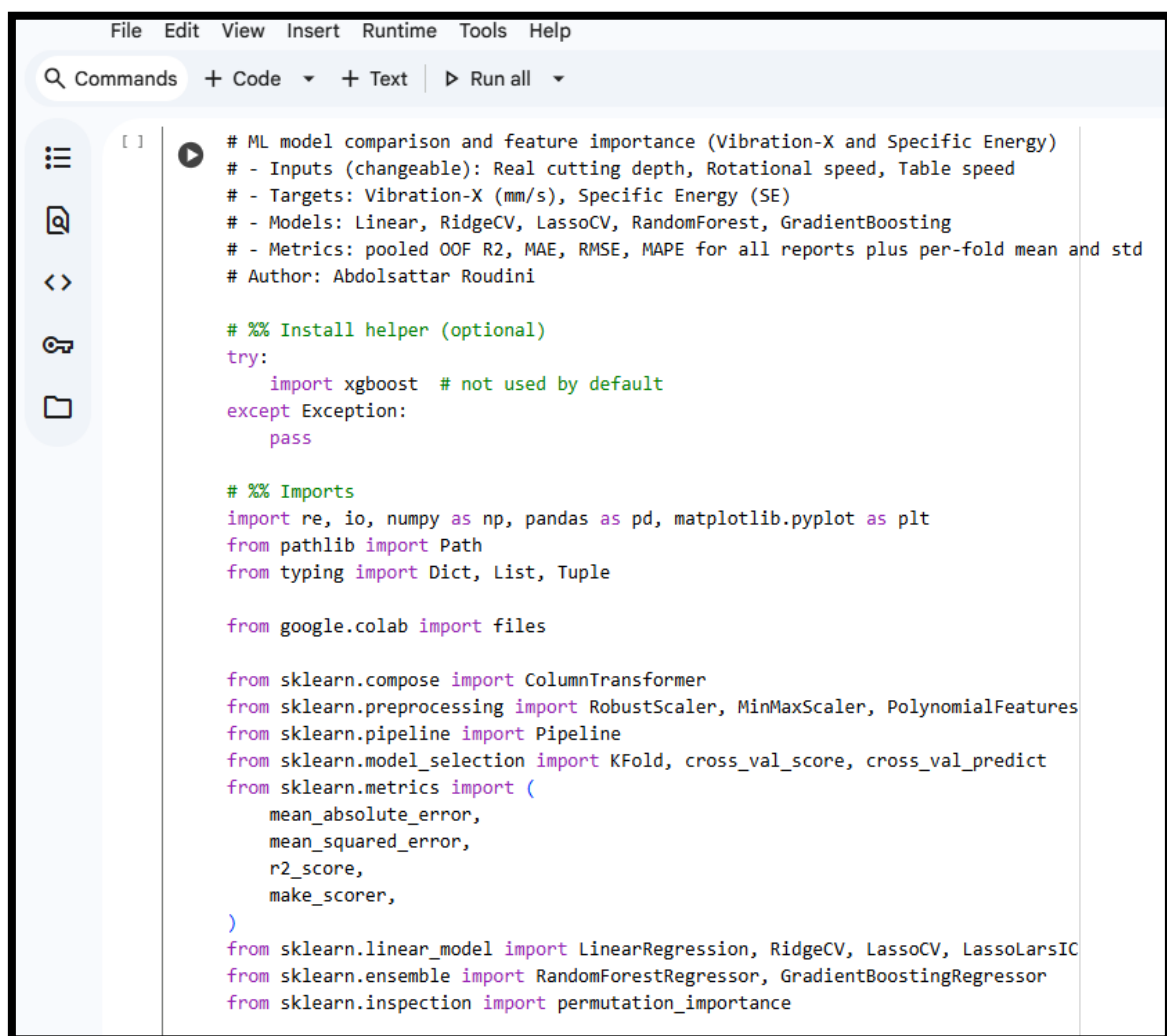
components aligned with the cutting direction and two orthogonal axes; this is the standard frame used in roadheader cutting models and tests (Q. Zhang et al., 2022; Zhu et al., 2022). In linear limestone cutting tests with a conical pick, the reported ratio of mean normal to mean cutting force FN_m/FC_m ranges from 0.16 to 0.38, directly implying that cutting (tangential) force dominates the average loading, an effect that would preferentially excite vibration along the cutting direction, as seen in the X-axis records (Kim et al., 2024).

At the machine scale, roadheader cutting-head dynamics further show strong directional responses tied to the applied cutting loads and operating regime. Depending on kinematics (e.g., cross-cutting vs drilling) and sensor placement, some studies report different axes having the largest acceleration during specific modes, but all attribute the response to the projection of pick forces into the machine frame (Z. Zhang et al., 2022b). Non-contact measurements with high-speed cameras confirm substantial, repeatable vibrations of the cutting head during cutting, consistent with force-driven excitation from the picks (Cheluszka & Mann, 2019).

3.3. Most Important Feature

In this study, Python was used throughout the analysis and implementation of the workflow. The software stack was organized by workflow stage. Excel-based datasets from Mix-1 to Mix-3 were ingested with pandas, which handled multi-sheet reading, coercion of measurement fields to numeric types, harmonization of heterogeneous headers, and table merges where required, with NumPy providing vectorized computation on the cleaned frames. Byte-level input and path resolution were managed using the io and pathlib modules. Regular expressions via re were used to standardize column names to the canonical schema adopted in this work (depth, rot, table, vibration_x, SE) and to parse sample identifiers such as Mix1, Mix2, and Mix3. In the Colab environment, Google Colab files were used to accept the three mixes of Excel workbooks. Preprocessing and pipelines were implemented with scikit-learn using ColumnTransformer and Pipeline to define feature flows per model, RobustScaler for linear estimators, MinMaxScaler for the symbolic regression stage, and PolynomialFeatures to generate sparse polynomial bases. Model training and validation used LinearRegression, RidgeCV, LassoCV, RandomForestRegressor, and GradientBoostingRegressor, evaluated with KFold, cross_val_score, and cross_val_predict, and scored using r2_score, mean_absolute_error, and make_scorer. Model inspection and

symbolic regression relied on `permutation_importance` for model-agnostic feature importance and on `LassoLarsIC` with the Akaike Information Criterion to obtain compact, human-readable formulas. Visualizations were produced with `matplotlib.pyplot`, including model comparison charts, permutation importance bar charts, and cross-validated predicted versus observed scatter plots. Output was exported as CSV files with `pandas` for reproducibility. The environment optionally installs `xgboost` for future extensions, although it was not used in the analyses reported here. Figure 3.14 shows the initial Colab cell that installs the optional `xgboost` package and imports all libraries used in the workflow.



```

File Edit View Insert Runtime Tools Help
Q Commands + Code + Text ▶ Run all
[ ] ▶
# ML model comparison and feature importance (Vibration-X and Specific Energy)
# - Inputs (changeable): Real cutting depth, Rotational speed, Table speed
# - Targets: Vibration-X (mm/s), Specific Energy (SE)
# - Models: Linear, RidgeCV, LassoCV, RandomForest, GradientBoosting
# - Metrics: pooled OOF R2, MAE, RMSE, MAPE for all reports plus per-fold mean and std
# Author: Abdolsattar Roudini

# %% Install helper (optional)
try:
    import xgboost # not used by default
except Exception:
    pass

# %% Imports
import re, io, numpy as np, pandas as pd, matplotlib.pyplot as plt
from pathlib import Path
from typing import Dict, List, Tuple

from google.colab import files

from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import RobustScaler, MinMaxScaler, PolynomialFeatures
from sklearn.pipeline import Pipeline
from sklearn.model_selection import KFold, cross_val_score, cross_val_predict
from sklearn.metrics import (
    mean_absolute_error,
    mean_squared_error,
    r2_score,
    make_scorer,
)
from sklearn.linear_model import LinearRegression, RidgeCV, LassoCV, LassoLarsIC
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
from sklearn.inspection import permutation_importance

```

Figure 3.14. Required Python libraries imported in the Google Colab environment, including data handling, preprocessing, modeling, evaluation, inspection, and visualization modules

3.3.1. Most Important Feature Recognition for Specific Energy

The Mix-1 performance table for SE shows that Gradient Boosting achieves the highest cross-validated accuracy ($R^2 = 0.929$, $MAE = 60.64$), with Random Forest a close second ($R^2 = 0.924$, $MAE = 62.66$) (Table 3.5). This ranking is visible in the model-comparison figure (Fig. 3.15), where the ensemble bars exceed those of the linear family. The feature-importance panel (Table 3.6, and Fig. 3.16) indicates a dominant effect of depth (importance 97.3 %) relative to traverse speed 2.2%) and rotational speed (0.5%). The predicted versus observed plot for the Gradient Boosting model (Fig. 3.17) shows a compact point cloud closely aligned with the 1:1 reference line, consistent with the reported R^2 and MAE, with only a slight tendency to underpredict at the upper end of the SE range.

Table 3.5. Mix-1 SE, model performance

Mix-1-SE: Model Comparison				
ML Model	R^2	MAE	RMSE	MAPE
Gradient Boosting	0.929	60.64	79.45	11.41%
Random Forest	0.924	62.66	82.14	11.64%
Lasso CV	0.853	88.18	114.03	16.28%
Linear	0.852	89.22	114.26	16.78%
Ridge CV	0.852	89.08	114.33	16.69%

Table 3.6. Mix-1 SE, permutation feature importance

Mix-1-SE: Feature importance	
Feature	Importance
Depth	97.30%
Traverse Speed	2.20%
linear Rotational Speed	0.50%

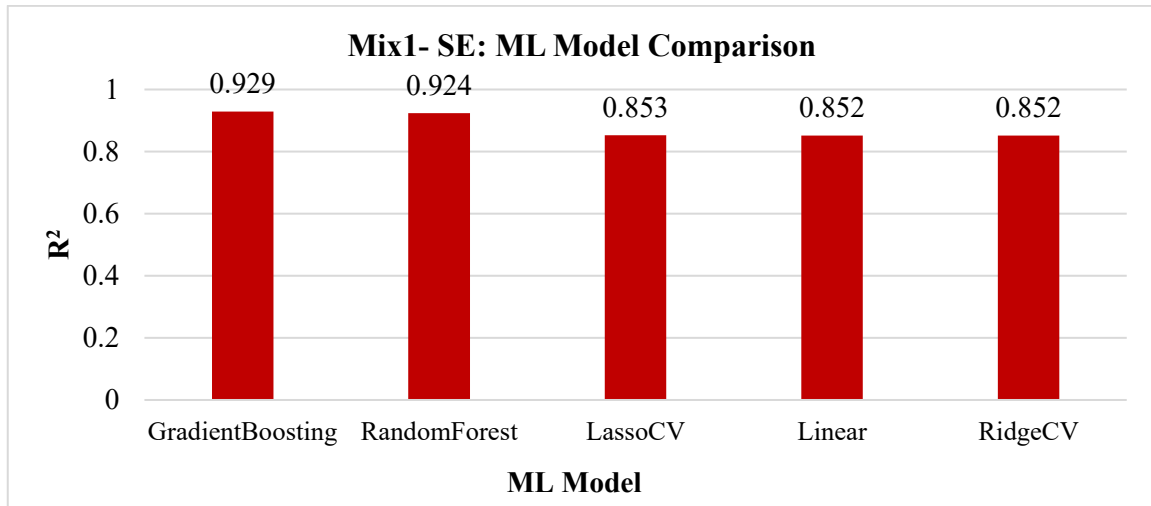


Figure 3.15. Mix-1 SE, model comparison

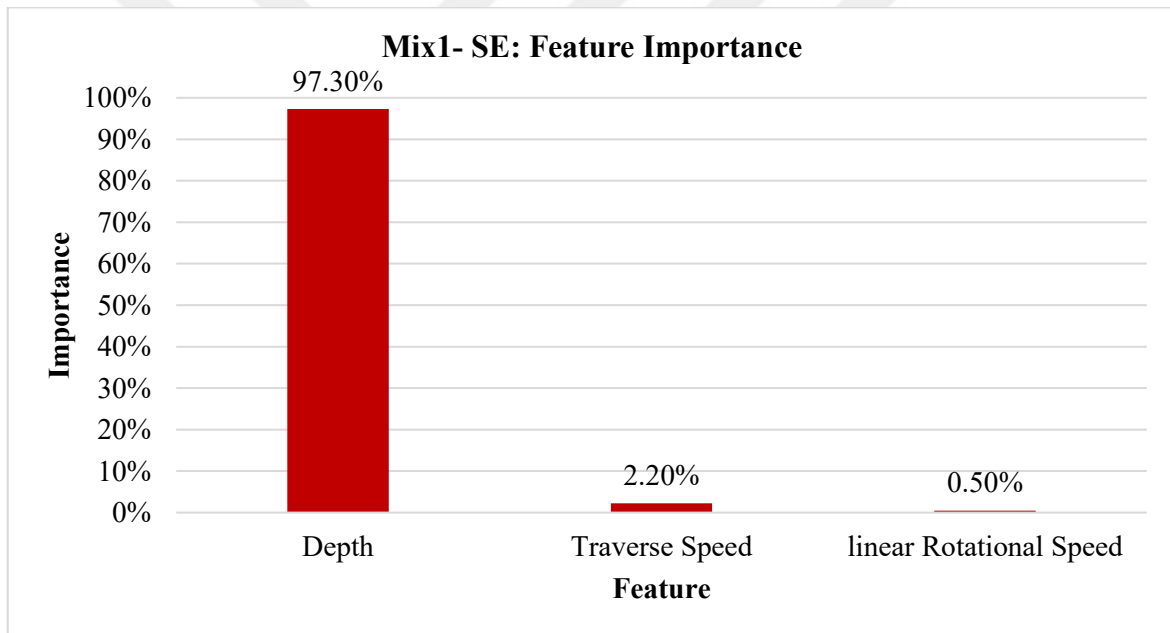


Figure 3.16. Mix-1 SE, feature importance

Excavation-Sample11.xlsx::Excavation data - GradientBoosting: R2=0.929, MAE=60.64, RMSE=79.45, MAPE=11.41%

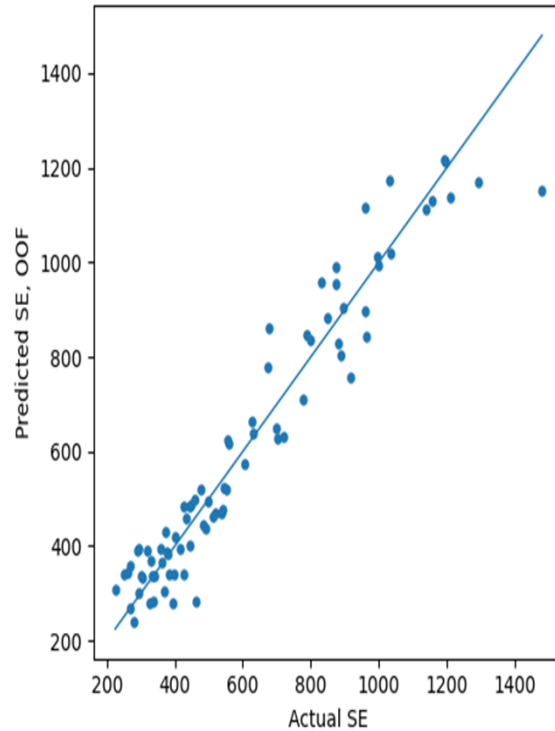


Figure 3.17. Predicted versus observed SE for Mix-1 using Gradient Boosting (K-fold cross validation)

The Mix-2 performance table for SE indicates that Random Forest delivers the highest cross validated accuracy ($R^2 = 0.946$, $MAE = 41.03$), followed closely by Gradient Boosting ($R^2 = 0.944$, $MAE = 43.94$) (Table 3.7). This ordering is visible in the model comparison figure (Fig. 3.18), where the ensemble bars exceed those of the linear family, which cluster around $R^2 = 0.881$ with $MAE \approx 67$. The permutation importance summary (Table 3.8 and Fig. 3.19) again identifies depth as the dominant driver of SE in this sample (importance 97.8%), while traverse speed (1.85%) and rotational speed (0.35 %) contribute only marginally. Consistent with these results, the predicted versus observed plot for the Random Forest model (Fig. 3.20) shows a compact point cloud aligned with the 1:1 reference line across the operating range, with only slight underprediction at the highest SE values.

Table 3.7. Mix-2 SE, model performance

Mix-2-SE: Model Comparison				
ML Model	R²	MAE	RMSE	MAPE
Random Forest	0.946	41.03	58.57	7.71%
Gradient Boosting	0.944	43.94	59.69	8.72%
Lasso CV	0.881	66.49	87.13	13.29%
Linear	0.881	66.96	87.13	13.48%
Ridge CV	0.881	66.89	87.14	13.45%

Table 3.8. Mix-2 SE, permutation feature importance

Mix-2-SE: Feature importance	
Feature	Importance
Depth	97.80%
Traverse Speed	1.85%
linear Rotational Speed	0.35%

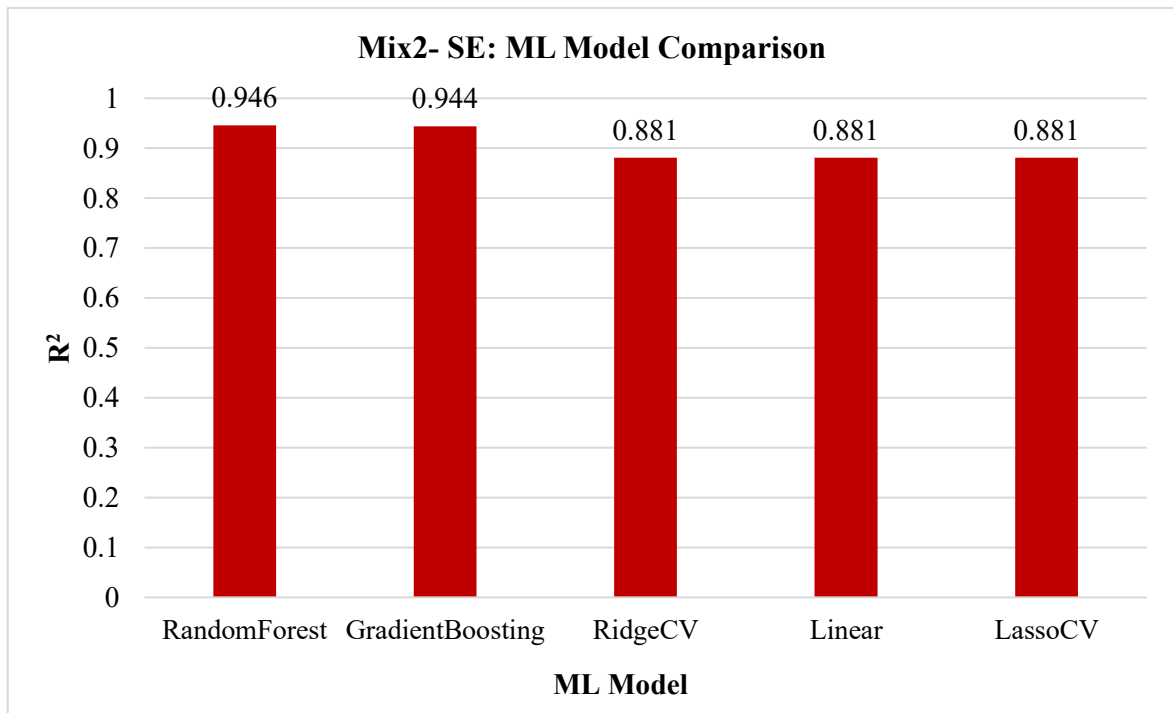


Figure 3.18. Mix-2 SE, model comparison

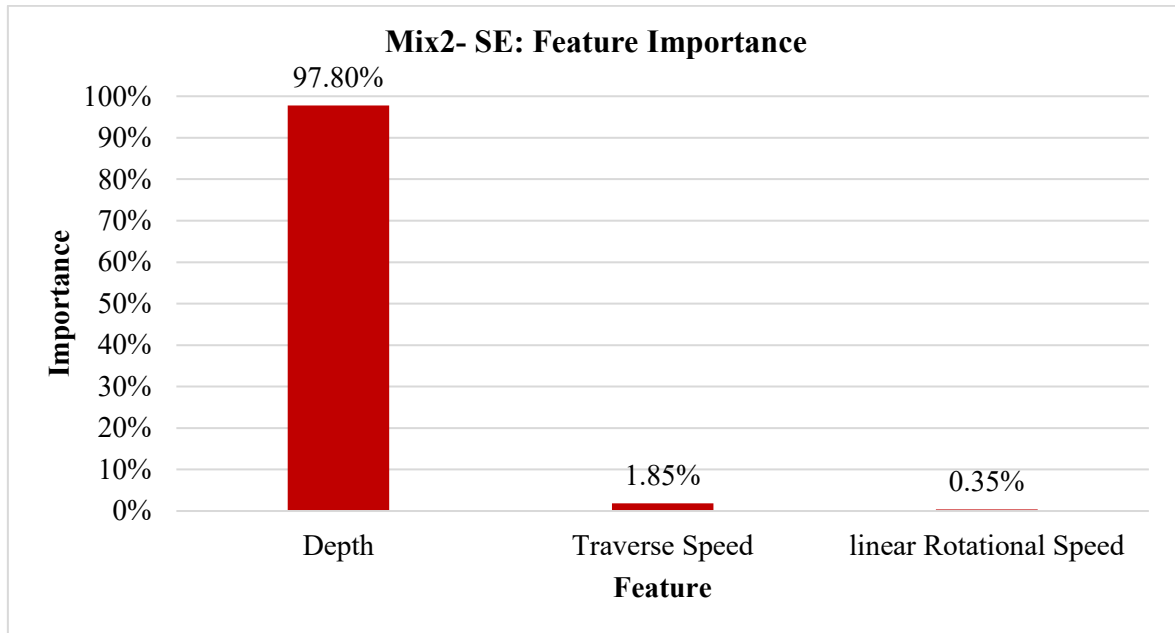


Figure 3.19. Mix-2 SE, feature importance

Excavation-Sample22.xlsx::Excavation Sample2 - RandomForest: R2=0.946, MAE=41.03, RMSE=58.57, MAPE=7.71%

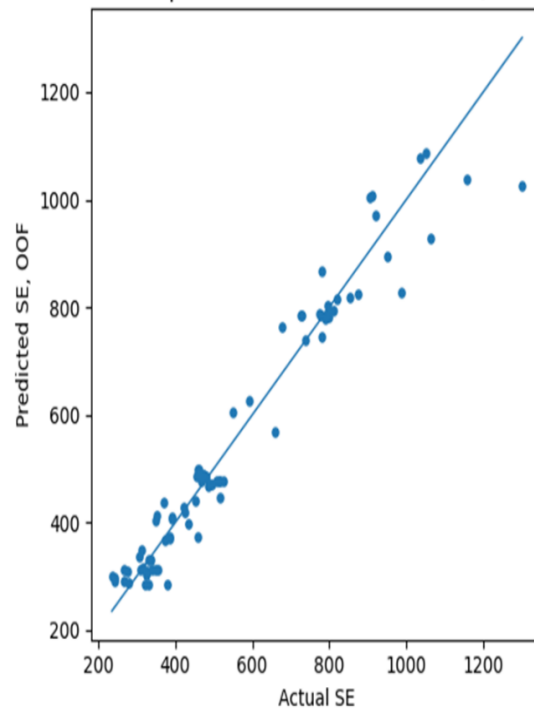


Figure 3.20. Predicted versus observed SE for Mix-2 using Random Forest (K-fold cross validation)

The Mix-3 performance table for SE indicates that Gradient Boosting provides the most accurate estimates, with cross-validated R^2 of 0.974 and MAE of 29.84, while Random Forest follows closely with R^2 of 0.966 and MAE of 32.65 (Table 3.9). This ordering is echoed in the model comparison graphic (Fig. 3.21), where the ensemble bars clearly exceed those for the linear group, which clusters near $R^2 \approx 0.872$ with MAE around 75. The important summary (Table 3.10 and Fig. 3.22) identifies penetration depth as the controlling predictor (importance 97.7 %), with traverse speed and rotational speed contributing only marginal effects (2.05 and 0.25 %). The predicted versus observed scatter for the Gradient Boosting model (Fig. 3.23) shows a tight cloud along the 1:1 line across the range of SE values, consistent with the very low error. Taken together, the table and figures show that SE in Mix-3 is governed primarily by cutting depth, with the two speed variables acting as secondary tuning parameters within the tested conditions.

Table 3.9. Mix-3 SE, model performance

Mix-3-SE: Model Comparison				
ML Model	R^2	MAE	RMSE	MAPE
Gradient Boosting	0.974	29.84	42.91	6.18%
Random Forest	0.966	32.65	49.47	6.63%
Ridge CV	0.872	75.35	96.05	19.61%
Linear	0.871	75.99	96.16	19.98%
Lasso CV	0.87	75.58	96.55	19.67%

Table 3.10. Mix-3 SE, permutation feature importance

Mix-3-SE: Feature importance	
Feature	Importance
Depth	97.70%
Traverse Speed	2.05%
linear Rotational Speed	0.25%

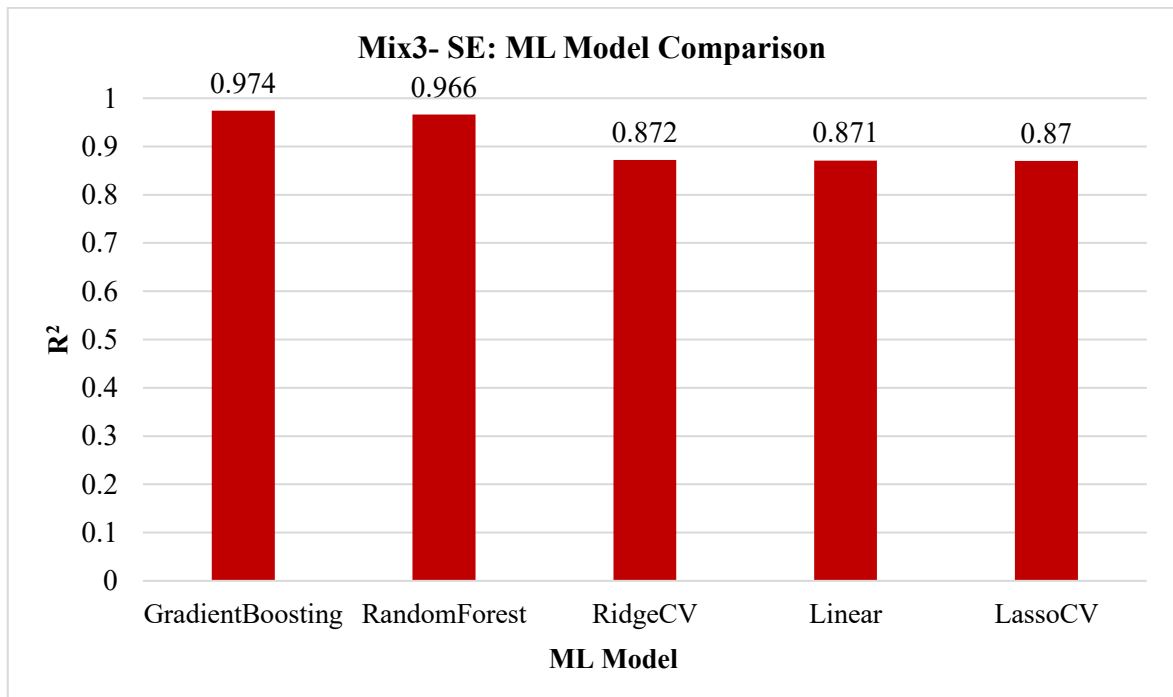


Figure 3.21. Mix-3 SE, model comparison

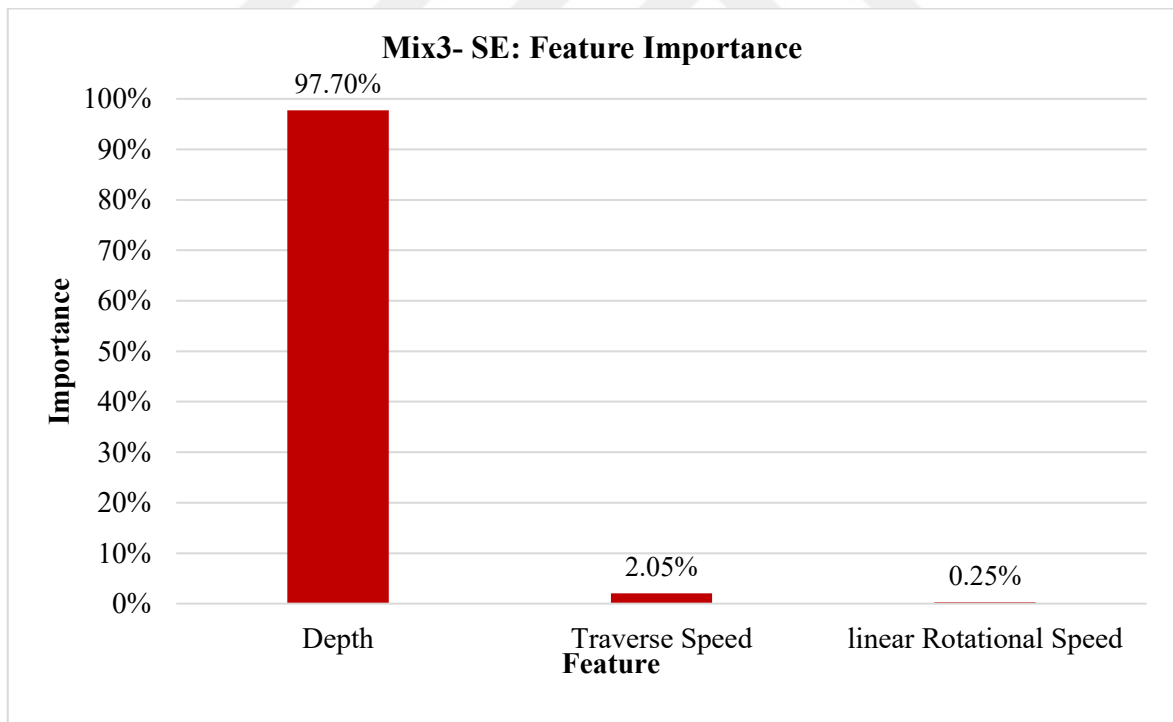


Figure 3.22. Mix-3 SE, feature importance

Excavation-Sample33.xlsx::Sheet1 - GradientBoosting: $R^2=0.974$, $MAE=29.84$, $RMSE=42.91$, $MAPE=6.18\%$

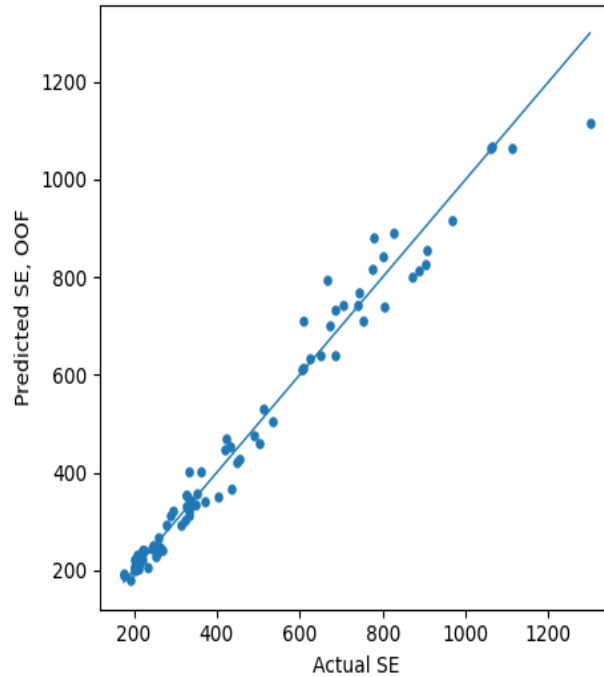


Figure 3.23. Predicted versus observed SE for Mix-3 using Gradient Boosting (K-fold cross validation)

3.3.2. Most Important Feature Recognition for Vibration-X

The Mix-1 performance table for Vibration-X shows LassoCV as the leading estimator with $R^2 = 0.338$ and $MAE = 37.29$ mm/s, with RidgeCV and Linear essentially tied just behind it ($R^2 \approx 0.338$ - 0.337 , $MAE \approx 37.2$ - 37.3 mm/s) (Table 3.11). This ordering is visible in the model comparison panel (Fig. 3.24), where the linear family outperforms the tree ensembles, which return much lower scores (Random Forest $R^2 = 0.06$, Gradient Boosting $R^2 = -0.096$). Permutation importance analysis (Table 3.12; Fig. 3.25) indicates that cutting depth is the primary factor influencing vibration response, accounting for approximately 65 percent of the total importance. Rotational speed also contributes meaningfully (33 percent), whereas traverse speed has minimal effect (about 2 percent) under the tested conditions. Consistent with these statistics, the predicted versus observed plot for RidgeCV (Fig. 3.26) shows moderate alignment with the 1:1 line and noticeable scatter at higher vibration levels, which matches the modest R^2 and the MAE near 37 mm/s. In practical terms, vibration mitigation for Mix-1 should prioritize accurate control of cutting depth, with rotational speed

used as a secondary tuning parameter, and traverse speed expected to have minimal impact under the present operating conditions.

Table 3.11. Mix-1 Vibration-X, model performance

Mix-1-Vibration-x: Model Comparison				
ML Model	R²	MAE	RMSE	MAPE
Lasso CV	0.338	37.29	45.22	24.29%
Ridge CV	0.338	37.45	45.22	24.34%
Linear	0.337	37.44	45.24	24.21%
Random Forest	0.06	43.69	53.87	28.95%
Gradient Boosting	-0.096	47.46	58.16	31.36%

Table 3.12. Mix-1 Vibration-X permutation feature importance.

Mix-1- Vibration-x: Feature importance	
Feature	Importance
Depth	65.30%
linear Rotational Speed	33.10%
Traverse Speed	1.60%

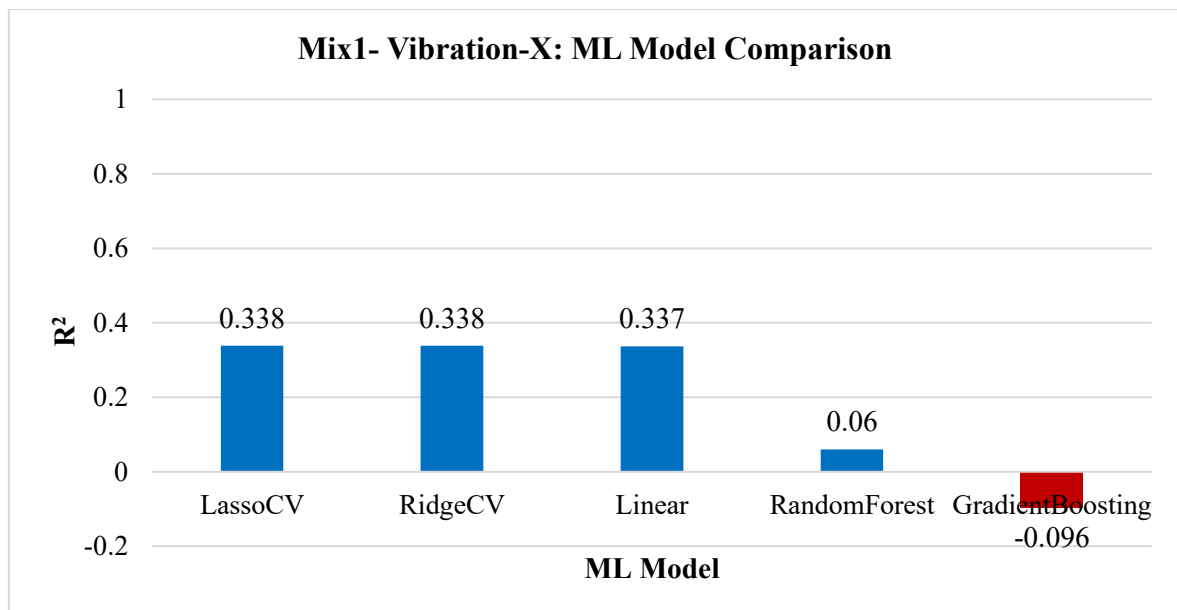


Figure 3.24. Mix-1 Vibration-X, model comparison

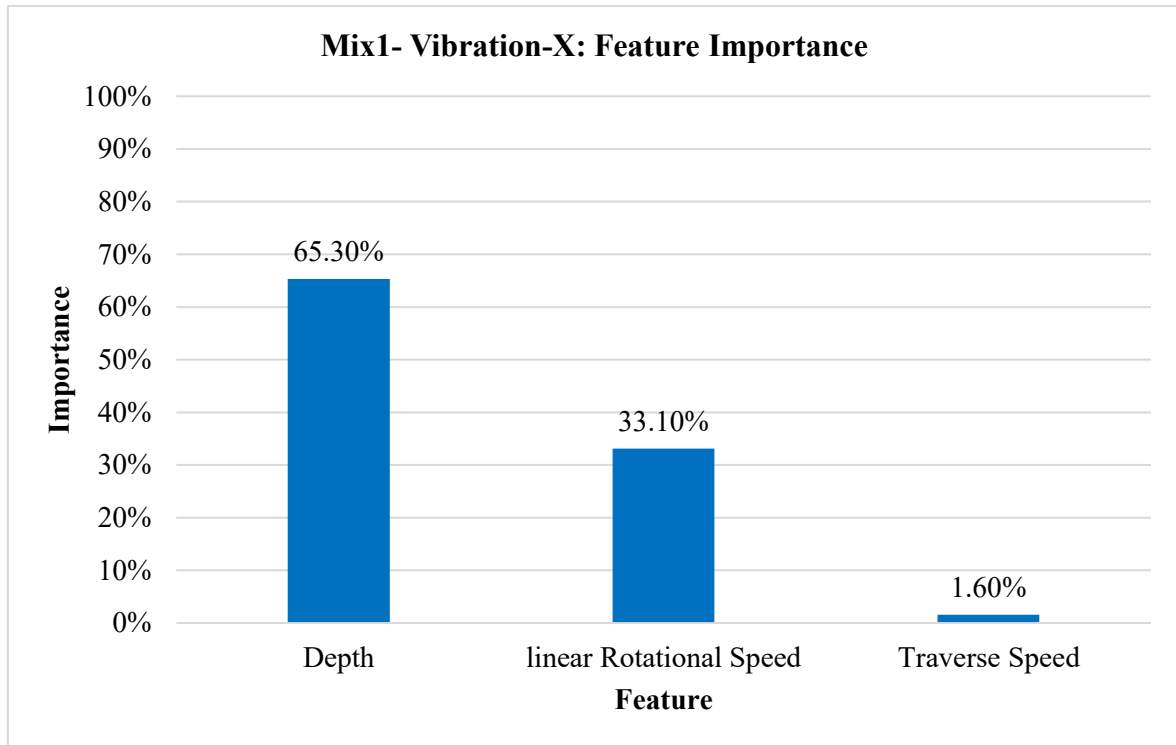


Figure 3.25. Mix-1 Vibration-X, feature importance

Excavation-Sample11.xlsx::Excavation data - LassoCV: $R^2=0.338$, $MAE=37.29$, $RMSE=45.22$, $MAPE=24.29\%$

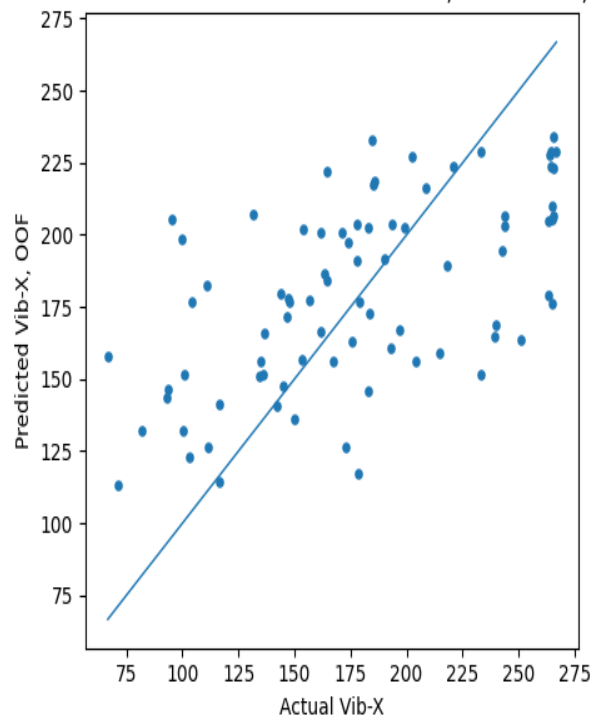


Figure 3.26. Mix-1 Vibration-X, predicted versus observed using Ridge CV

The Mix-2 performance table for Vibration-X shows that Random Forest attains the best cross validated accuracy ($R^2 = 0.659$, $MAE = 22.4$ mm/s), followed by Gradient Boosting ($R^2 = 0.626$, $MAE = 24.27$ mm/s) (Table 3.13). The linear family (RidgeCV, Linear, LassoCV) clusters around $R^2 \approx 0.51$ with $MAE \approx 28.3$ mm/s, indicating that nonlinear structure is present and is captured better by the ensembles. This ordering is evident in the model comparison panel (Fig. 3.27). Feature-importance analysis (Table 3.14; Fig. 3.28) confirms that cutting depth is the dominant factor influencing vibration (≈ 65 percent), followed by rotational speed (≈ 30 percent), while traverse speed plays only a minor role (< 6 percent). The predicted versus observed plot for Random Forest (Fig. 3.29) shows points concentrated near the 1:1 line across the operating range, consistent with the reported R^2 and MAE, with only mild dispersion at the upper tail of vibration values. Operationally, vibration control in Mix-2 should prioritize precise regulation of cutting depth, while rotational speed can be used as a secondary tuning lever and traverse speed is expected to have limited impact within the tested conditions.

Table 3.13. Mix-2 Vibration-X, model performance.

Mix-2-Vibration-x: Model Comparison				
ML Model	R^2	MAE	RMSE	MAPE
Random Forest	0.659	22.4	29.21	14.33%
Gradient Boosting	0.626	24.27	30.57	15.27%
Ridge CV	0.508	28.28	35.05	19.26%
Linear	0.508	28.27	35.07	19.22%
Lasso CV	0.506	28.33	35.12	19.27%

Table 3.14. Mix-2 Vibration-X permutation feature importance

Mix-2- Vibration-x: Feature importance	
Feature	Importance
Depth	64.70%
linear Rotational Speed	30.00%
Traverse Speed	5.30%

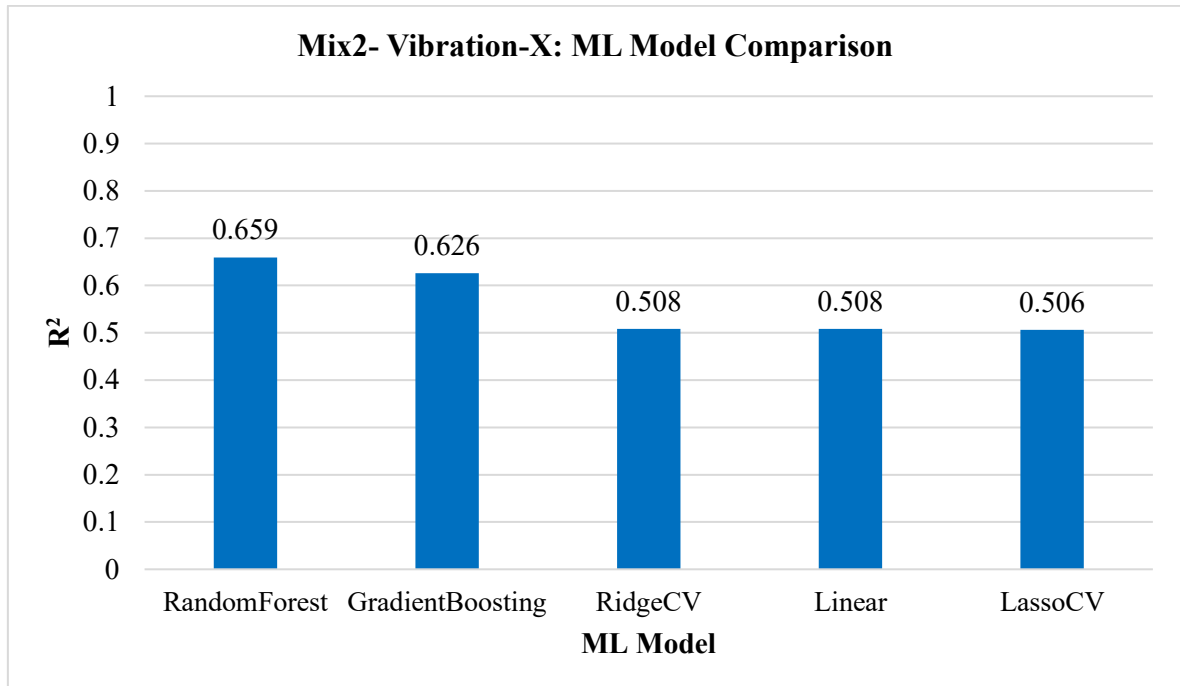


Figure 3.27. Mix-2 Vibration-X, model comparison

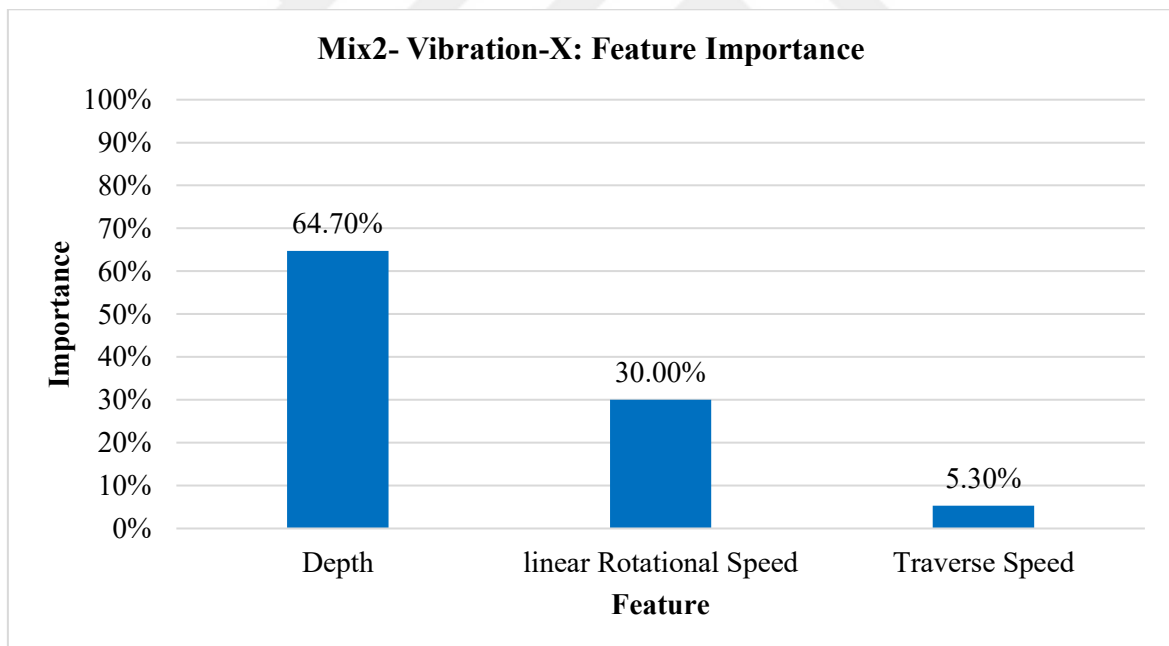


Figure 3.28. Mix-2 Vibration-X, feature importance

Excavation-Sample22.xlsx::Excavation Sample2 - RandomForest: R2=0.659, MAE=22.40, RMSE=29.21, MAPE=14.33%

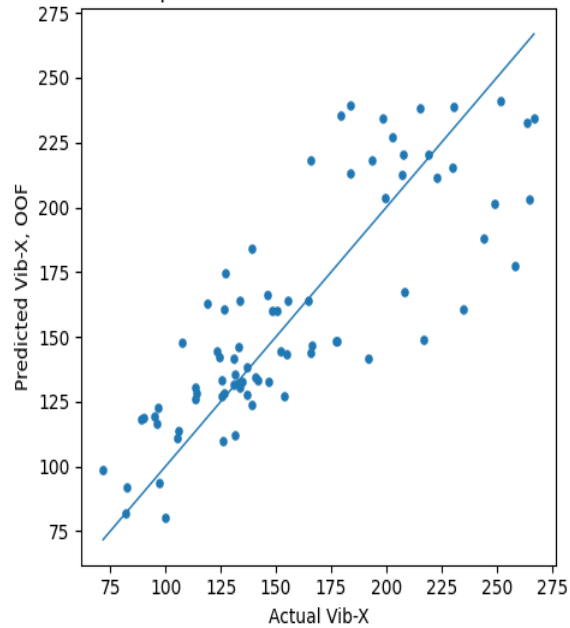


Figure 3.29. Mix-2 Vibration-X, predicted versus observed using Random Forest

The Mix-3 Vibration-X results (Table 3.15) show that the three linear models performed similarly and achieved the highest predictive accuracy. LassoCV, Linear, and RidgeCV yielded nearly identical R^2 values (≈ 0.46) and MAE values of about 10.1–10.2 mm/s. Random Forest and Gradient Boosting performed less well, with lower R^2 values (0.424 and 0.300) and noticeably higher error levels, as also reflected in Fig. 3.30. Permutation-importance analysis (Table 3.16; Fig. 3.31) indicates that cutting depth is the most influential factor in vibration response (≈ 61 percent), followed by rotational speed (≈ 28 percent), while traverse speed plays a smaller role (≈ 12 percent) within the tested range.

The prediction–observation trend for the linear models is aligned reasonably well with the 1:1 reference line (Fig. 3.32), consistent with the moderate R^2 and low MAE values. Overall, Mix-3 vibration behavior is governed mainly by cutting depth, with rotational speed serving as a secondary adjustment variable, and traverse speed contributing relatively little under these operating conditions.

Table 3.15. Mix-3 Vibration-X, model performance

Mix-3-Vibration-x: Model Comparison				
ML Model	R²	MAE	RMSE	MAPE
Lasso CV	0.466	10.13	12.76	20.82%
Linear	0.464	10.15	12.78	20.87%
Ridge CV	0.463	10.13	12.8	20.73%
Random Forest	0.424	11.05	13.25	23.55%
Gradient Boosting	0.3	11.74	14.61	24.73%

Table 3.16. Mix-3 Vibration-X permutation feature importance

Mix-3- Vibration-x: Feature importance	
Feature	Importance
Depth	60.70%
linear Rotational Speed	27.60%
Traverse Speed	11.70%

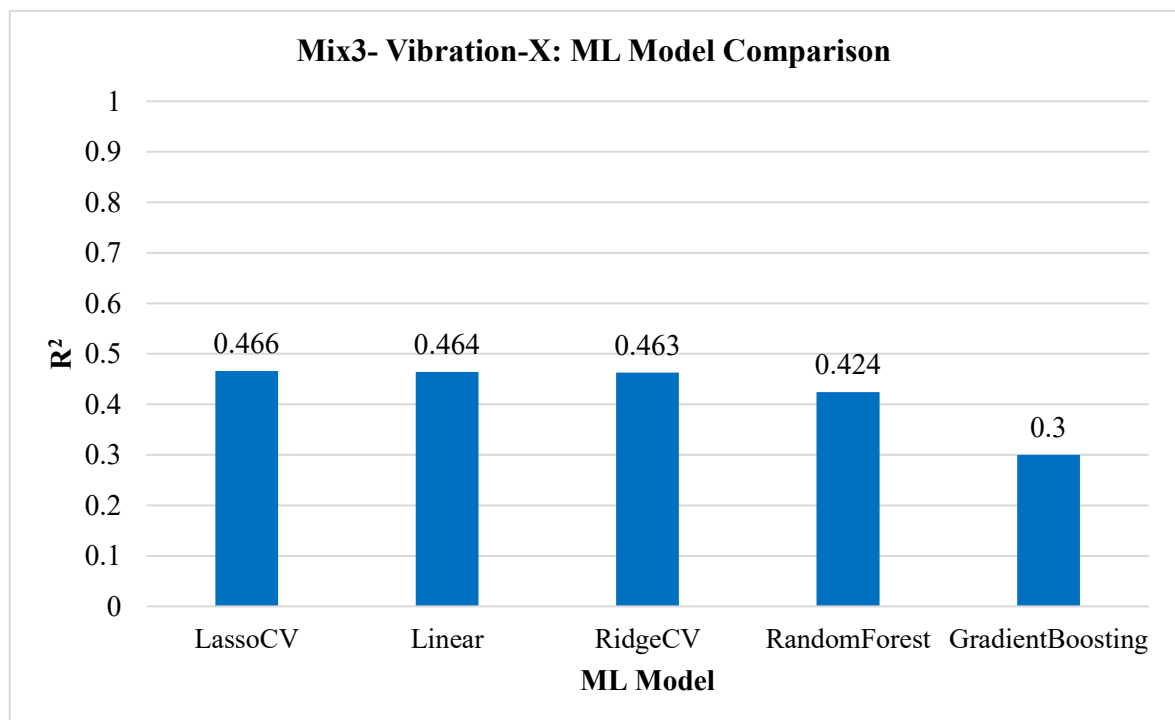


Figure 3.30. Mix-3 Vibration-X, model comparison

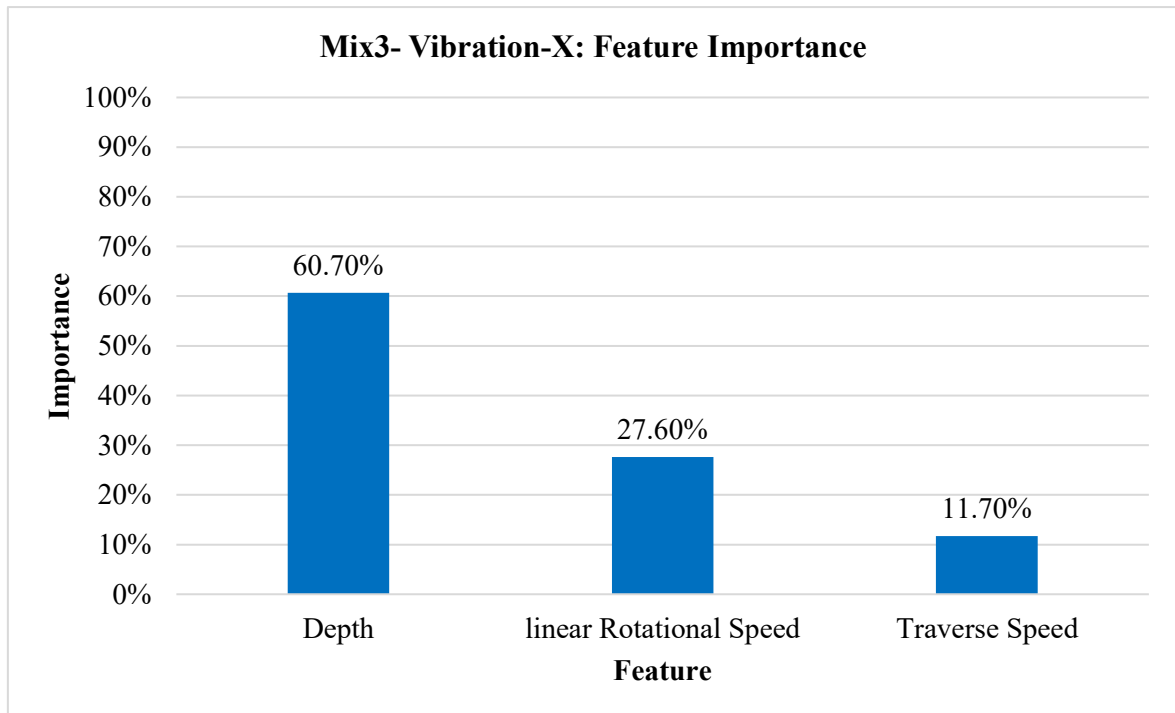


Figure 3.31. Mix-3 Vibration-X, feature importance.

Excavation-Sample33.xlsx::Sheet1 - LassoCV: $R^2=0.466$, $MAE=10.13$, $RMSE=12.76$, $MAPE=20.82\%$

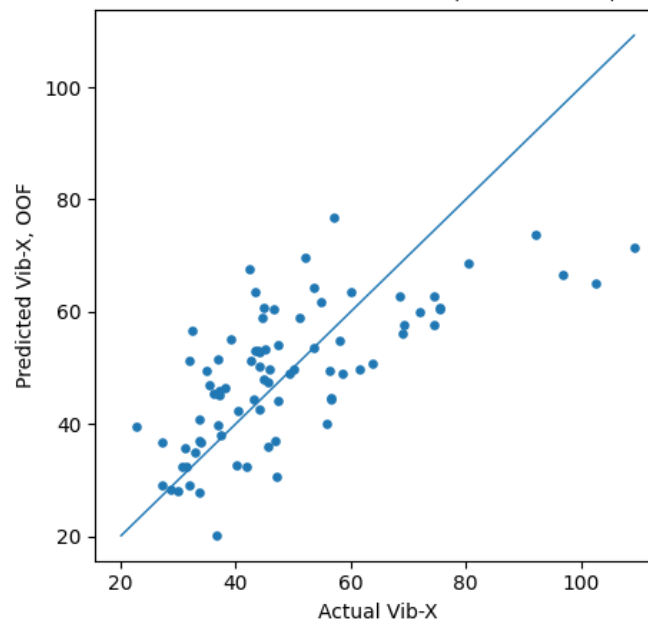


Figure 3.32. Mix-3 Vibration-X, predicted versus observed using Ridge CV

Taken together, the feature-importance and model-performance summaries show a consistent pattern across the three mixes. In all cases, depth is by far the dominant factor

influencing Specific Energy (Fig. 3.33), contributing close to 98 percent in Mix-1, Mix-2, and Mix-3, while rotational speed and traverse speed each account for only a very small share. For Vibration-X (Fig. 3.34), depth remains the primary driver, but rotational speed assumes a clearer secondary role, most noticeably in Mix-2. Traverse speed again shows only limited influence.

Model-performance trends (Fig. 3.35) reflect these relationships. Ensemble methods give the most accurate SE predictions: Gradient Boosting performs best in Mix-1 and Mix-3 ($R^2 = 0.93$ and 0.97), while Random Forest leads in Mix-2 ($R^2 = 0.95$). In contrast, Vibration-X is better captured by linear models in Mix-1 and Mix-3 (Lasso, $R^2 = 0.338$ and 0.466), with Random Forest performing best in Mix-2 ($R^2 = 0.66$).

Overall, the three plots and the corresponding results in Table 3.17 point to a clear control hierarchy: depth is the main governing parameter for both SE and vibration response; rotational speed can be used to adjust vibration, particularly in Mix-2; and traverse speed plays only a minor role within the tested operating conditions.

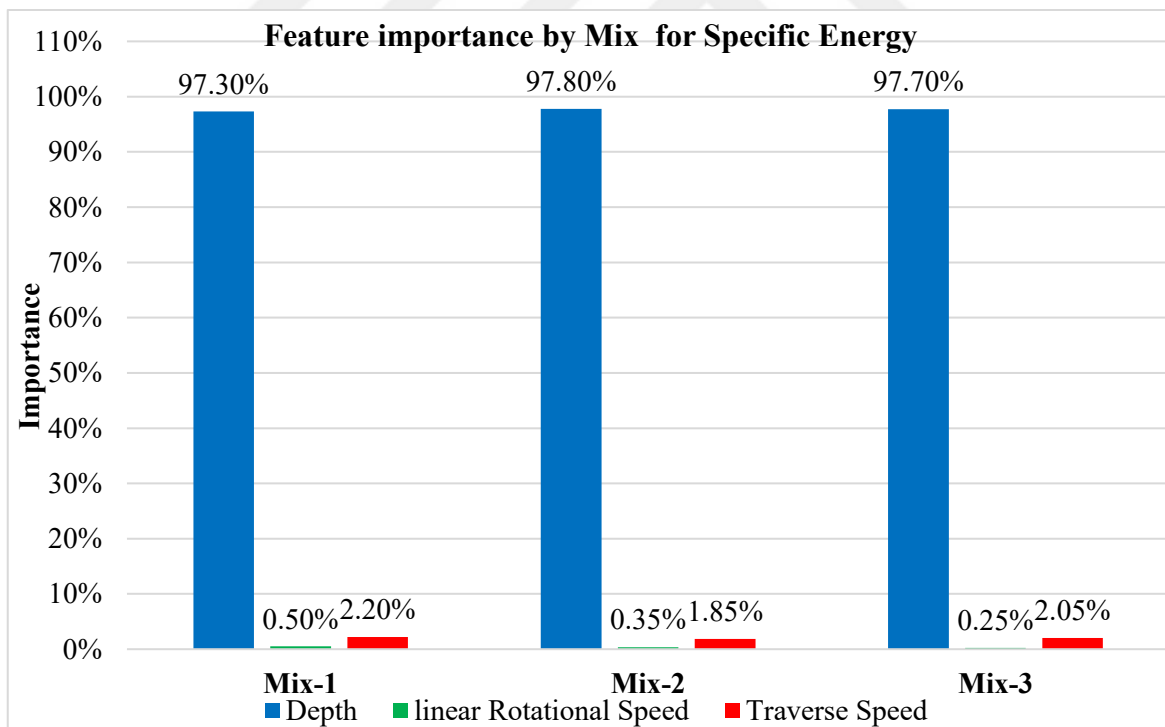


Figure 3.33. Feature importance by Mix for SE

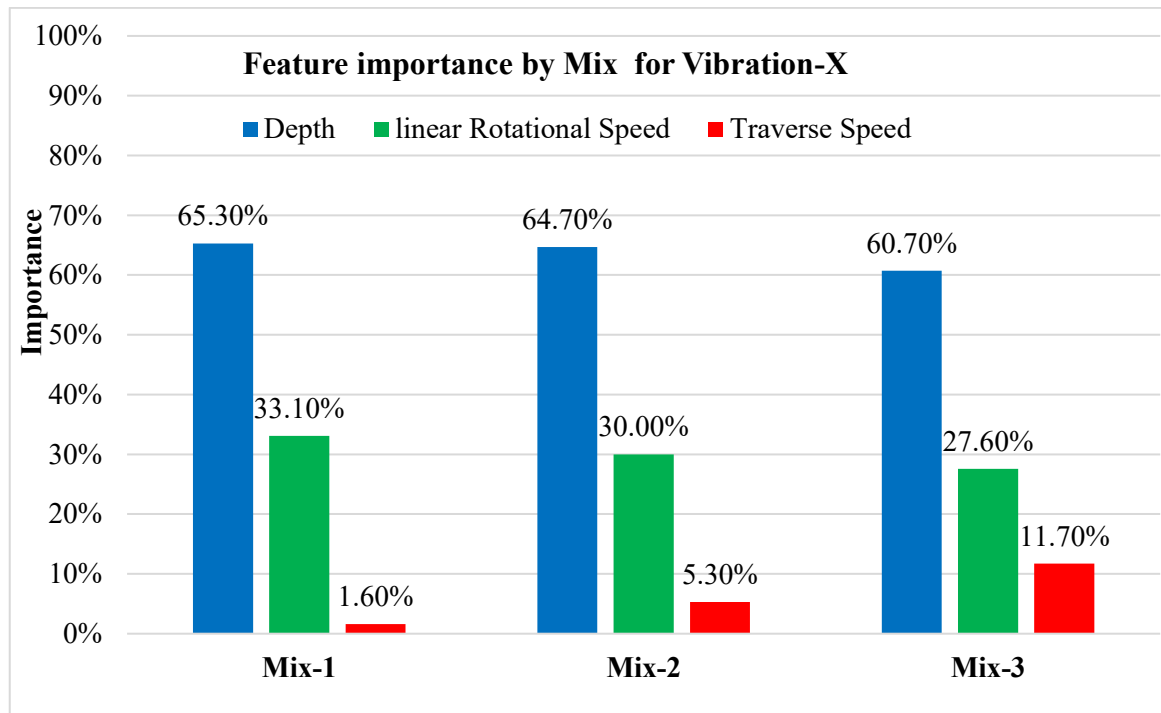


Figure 3.34. Feature importance by Mix for Vibration-X

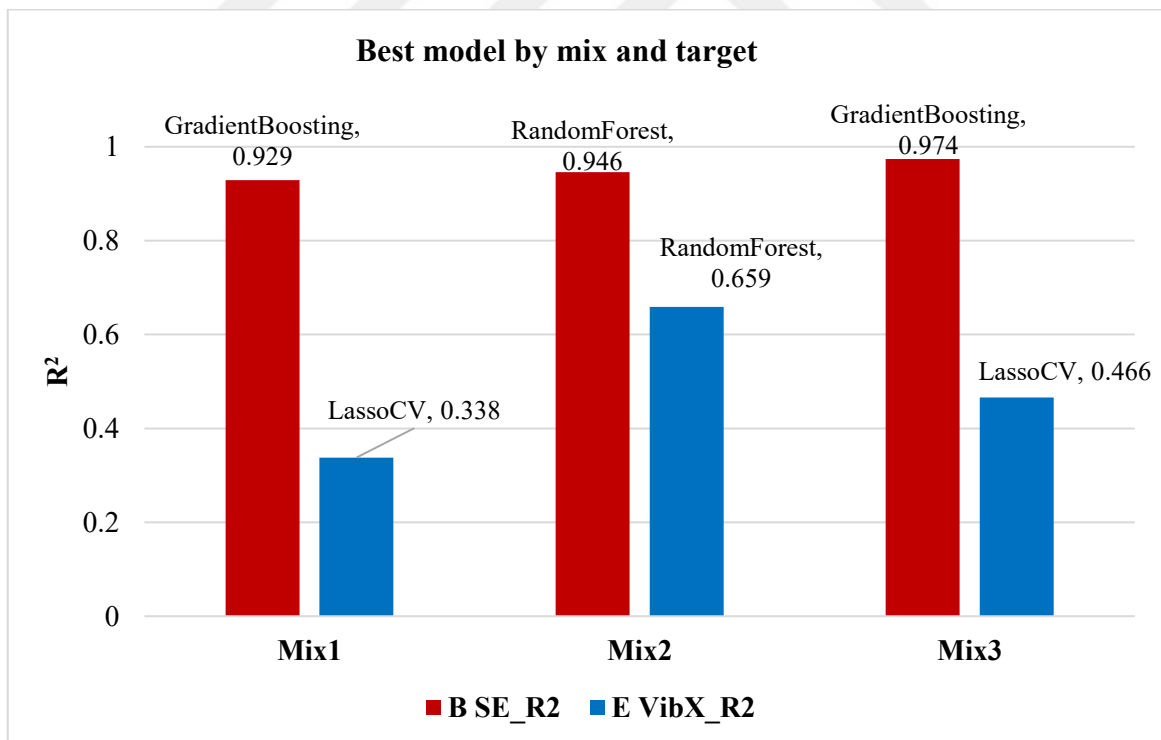


Figure 3.35. Best ML model by mix and target

Table 3.17. The performance comparison in best model by mix and target

Mix	Target	Best_Model	cv_R2	cv_MAE
Mix1	SE	GradientBoosting	0.929	60.64
Mix2	SE	RandomForest	0.946	41.03
Mix3	SE	GradientBoosting	0.974	29.84
Mix1	Vibration-X	LassoCV	0.338	37.29
Mix2	Vibration-X	RandomForest	0.659	22.4
Mix3	Vibration-X	LassoCV	0.466	10.13

3.4. Optimization of Cutting Variables

Figure 3.36 presents the initialization cell of the Google Colab notebook that implements the end-to-end optimization workflow. The environment relies on Python packages for data handling and modeling, specifically numpy and pandas for numerical and tabular operations, matplotlib for visualization, scikit-learn for supervised learning and cross-validation, and pyswarms for metaheuristic search. Auxiliary modules handle file I/O and schema harmonization, where openpyxl enables ingestion of Excel workbooks, and robust column pickers map heterogeneous headers to the canonical variable's depth, rot, table, Vibration-X, and SE. A fixed random seed is set for reproducibility.

Three Excel datasets, one per mix, are uploaded and parsed automatically. For each file, the notebook selects the sheet with the greatest number of rows, coerces measurement fields to numeric types, drops incomplete rows, and records observed bounds for depth and the discrete level sets for rot and table. This preserves the feasible region present in the experiments and allows the optimizer to return implementable set-points. The cleaned datasets then serve as training material for surrogate models that approximate the input response surfaces required for fast search.

For each mix and for each target, SE and Vibration-X, two ensemble regressors are compared, Random Forest and Gradient Boosting. Model selection is performed by K-fold cross-validation with the mean R^2 used as the primary criterion and mean absolute error as a secondary tiebreaker. The number of folds adapts to sample size in order to balance variance and bias in the estimate. The chosen regressors are then fitted and stored together with per-mix statistics, including Vibration-X caps defined from the empirical distribution, which are later used for penalty design.

The objective of optimization is a scalarization of both goals, namely the reduction of SE and the reduction of Vibration-X. Predicted SE and Vibration-X from the surrogates are normalized per mix, then combined with user-settable weights, with a balanced setting used by default. Several restrictions and preferences are encoded as differentiable penalties within the objective. Depths are constrained to satisfy a strict ordering across mixes, depth1 less than depth2 less than depth3, with a small margin. Rotational speed is constrained so that rot1 is not lower than rot2 or rot3. A soft preference encourages table1 to be lower than the minimum of table2 and table3. Values above the per-mix Vibration-X cap attract an additional penalty that increases with normalized depth to reflect operational caution at more aggressive cuts. A final term promotes the expected monotonic pattern of SE across mixes, highest for Class 1 and lowest for Class 3. The strength of each penalty is tunable. During evaluation, rot and table are rounded to the nearest observed discrete level, and depths are clipped to their empirical bounds, which enforces feasibility with respect to the experimental actuation grid.

Particle swarm optimization is the global search engine that explores the nine-dimensional decision vector, depth, rot, table for each of the three mixes. The first PSO run solves the balanced joint problem under the default weights and the full set of penalties, returning one recommended setting per class and writing them to `joint_pso_optimized_settings.csv`. Constraint checks are printed to document that the learned solution respects the ordering and preference structure. A second use of PSO implements a trade-off sweep in which the vibration and energy weights are varied, for example vibration-priority and energy-priority configurations. Each sweep re-optimizes the same nine-dimensional problem and records per-class recommendations and predicted responses in `tradeoff_summary_per_sample.csv`, together with the achieved objective value. In both uses, bounds are set from the empirical ranges, and the swarm size and iteration budget are chosen to provide stable convergence within practical runtime.

```

# %% [markdown]
# Joint PSO with cross-sample constraints, thresholds, and online control
# - Optimizes [depth, rot, table] for all 3 samples jointly (surrogate: RF or GB)
# - Enforces:
#   * depth1 < depth2 < depth3 (strict with small margin)
#   * rot1 >= rot2 and rot1 >= rot3
#   * table1 can be lower than others (soft preference, not hard constraint)
# - Penalizes high vibration by reducing depth (per-sample cap)
# - Learns Vib-X thresholds t1, t2 for 3-class categorization
# - Provides recommend_settings_from_vibx(v) for online use
# Author: Abdolsattar Roudini

# %% Install
!pip -q install openpyxl pyswarms

# %% Imports
import io, re, numpy as np, pandas as pd, matplotlib.pyplot as plt
from pathlib import Path
from typing import List, Tuple
from IPython.display import display
from google.colab import files
from sklearn.model_selection import KFold, cross_val_score
from sklearn.metrics import make_scorer, mean_absolute_error
from sklearn.pipeline import Pipeline
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
import pyswarms as psw # renamed to avoid collisions with any variable named `ps`

np.random.seed(42)

# ===== Tunable knobs =====
SWARM_SIZE = 100
PSO_ITERS = 180
CV_BIG = 5
CV_SMALL = 3

# Default objective weights
W_SE = 0.5
W_VIB = 0.5

```

Figure 3.36. Initialization cell of the Google Colab notebook that sets up the end-to-end optimization pipeline

To link offline optimization to prospective online control, the notebook learns two Vibration-X thresholds that partition the observed Vibration-X into three classes that align with the SE-based ranking of mixers. Thresholds are obtained by a simple misclassification minimization over candidate quantiles, with a local refinement step. The result, together with the distribution of Vibration-X for each class, is shown in the histogram titled “Vib-X distributions with learned thresholds”, and the values are stored in `vibx_thresholds.csv`. A helper function, `recommend_settings_from_vibx`, maps a real-time Vibration-X measurement to the detected class and returns the corresponding recommended set-points from the balanced PSO solution. This provides a direct rule for adjusting depth, rot, and table in response to measured vibration levels during operation.

3.4.1. Optimization Results and Trade-off Analysis

Table 3.18 lists the optimized settings obtained through PSO when different priorities are applied. The balanced case treats vibration and specific energy equally. Under this setting, all three mixes run at the same rotational speed of 4.17 m/s. Mix-1 operates with a slightly lower traverse speed than Mix-2 and Mix-3. Depth increases from Mix-1 to Mix-3. Predicted vibration and specific energy both decline across the mixes, which agrees with the expected ranking of mechanical difficulty.

The vibration-priority case shifts the solution toward reduced vibration. Depth becomes smaller for every mix, with the largest reduction occurring in Mix-1. Rotational and traverse speeds remain unchanged. Vibration drops slightly in Mix-1 and Mix-3, while Mix-2 shows little change. Specific energy becomes higher for all mixes, because optimization favors vibration reduction rather than energy savings.

The energy-priority case moves the solution in the opposite direction. Depth becomes larger for all mixes. Mix-1 again takes a lower traverse speed, while the other mixes remain at 187.37 cm/min. Specific energy drops for every mix, and vibration changes only moderately. This confirms that the optimizer trades deeper cutting to improve energy performance. Across all settings, the ordering of depth from Mix-1 to Mix-3 remains consistent, showing that the cross-sample structure is preserved.

Figure 3.37 presents the vibration distributions with learned thresholds $t_1 = 86$ mm/s and $t_2 = 141.2$ mm/s, which separate the three material classes. Mixes with vibration below t_1 are assigned to Class 3, values between t_1 and t_2 align with Class 2, and values above t_2 are associated with Class 1. This agrees with the specific energy-based ordering, in which Class 1 corresponds to the highest median SE of about 513.640 MJ/m³, Class 2 has a median of about 480.028 MJ/m³ and Class 3 has the lowest median SE of about 353.649 MJ/m³. By using these thresholds, the machine can identify the likely class directly from measured vibration and then adopt the appropriate PSO-based settings taken from Table 3.18.

Table 3.18. Trade-off sweep results from joint PSO for Mix-1 to Mix-3

Mix	Trade off	w_vib	w_se	Depth (mm)	linear Rotational Speed (m/s)	Traverse Speed (cm/min)	Pred_VibX (mm/s)	Pred_SE (MJ/m ³)
Mix-1	Balanced	0.5	0.50	6.43	4.17	162.80	109.67	452.68
Mix-2	Balanced	0.5	0.50	7.40	4.17	187.37	142.11	373.38
Mix-3	Balanced	0.5	0.50	7.77	4.17	187.37	44.31	245.73
Mix-1	Vibration- priority	0.8	0.20	4.61	4.17	187.37	103.77	813.09
Mix-2	Vibration- priority	0.8	0.20	5.92	4.17	187.37	112.03	517.50
Mix-3	Vibration- priority	0.8	0.20	6.17	4.17	187.37	32.94	422.81
Mix-1	Energy- priority	0.2	0.80	6.97	4.17	162.80	136.93	354.15
Mix-2	Energy- priority	0.2	0.80	8.14	4.17	187.37	130.04	331.62
Mix-3	Energy- priority	0.2	0.80	8.41	4.17	187.37	50.02	210.14

In practice, the balanced configuration provides a reasonable default. The vibration-priority configuration is useful when vibration control is dominant, and the energy-priority setting is preferred when energy efficiency is the main objective while vibration remains within acceptable limits.

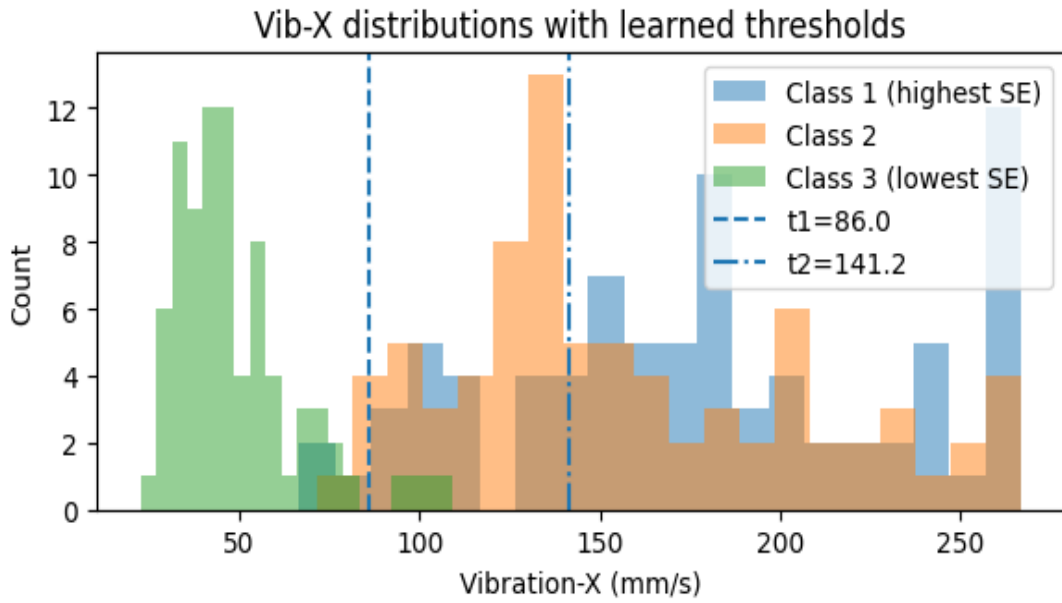


Figure 3.37. Vib-X distributions with learned thresholds t_1 and t_2 , supporting online class detection and set-point selection.

3.5. Pick Wear Assessment

During the cutting tests, a separate pick was used for each mix, and the same pick was kept in service only for the duration of its corresponding test. Pick wear was examined through mass-loss measurements, 3-D surface mapping of the tungsten-carbide tips, and visual inspection of the picks following excavation. The measured mass losses were 0.415 g for Pick 2, 0.337 g for Pick 1, and 0.272 g for Pick 3 (Table 3.19). These results indicate that Pick 2 experienced the most wear, followed by Pick 1, while Pick 3 was affected the least.

The 3-D surface maps (Figure 3.38) show the condition of the tungsten-carbide tips before and after cutting. All three picks initially exhibited smooth, dome-shaped geometry. After excavation, pick 2 displayed the most pronounced alteration of the carbide surface, characterized by a broader and more irregular worn region. Pick 1 showed moderate and relatively uniform changes, whereas the carbide of Pick 3 retained most of its original shape with only limited surface modification.

Table 3.19. Mass loss of Picks 1–3 after excavation.

Pick	Mix	Mass before cutting tests (g)	Mass after cutting tests(g)	Mass loss (g)	Mass loss (%)
1	Mix 1	208.9927	208.656	0.3368	0.1612
2	Mix 2	208.7466	208.3314	0.4152	0.1989
3	Mix 3	210.8315	210.5594	0.2721	0.1291

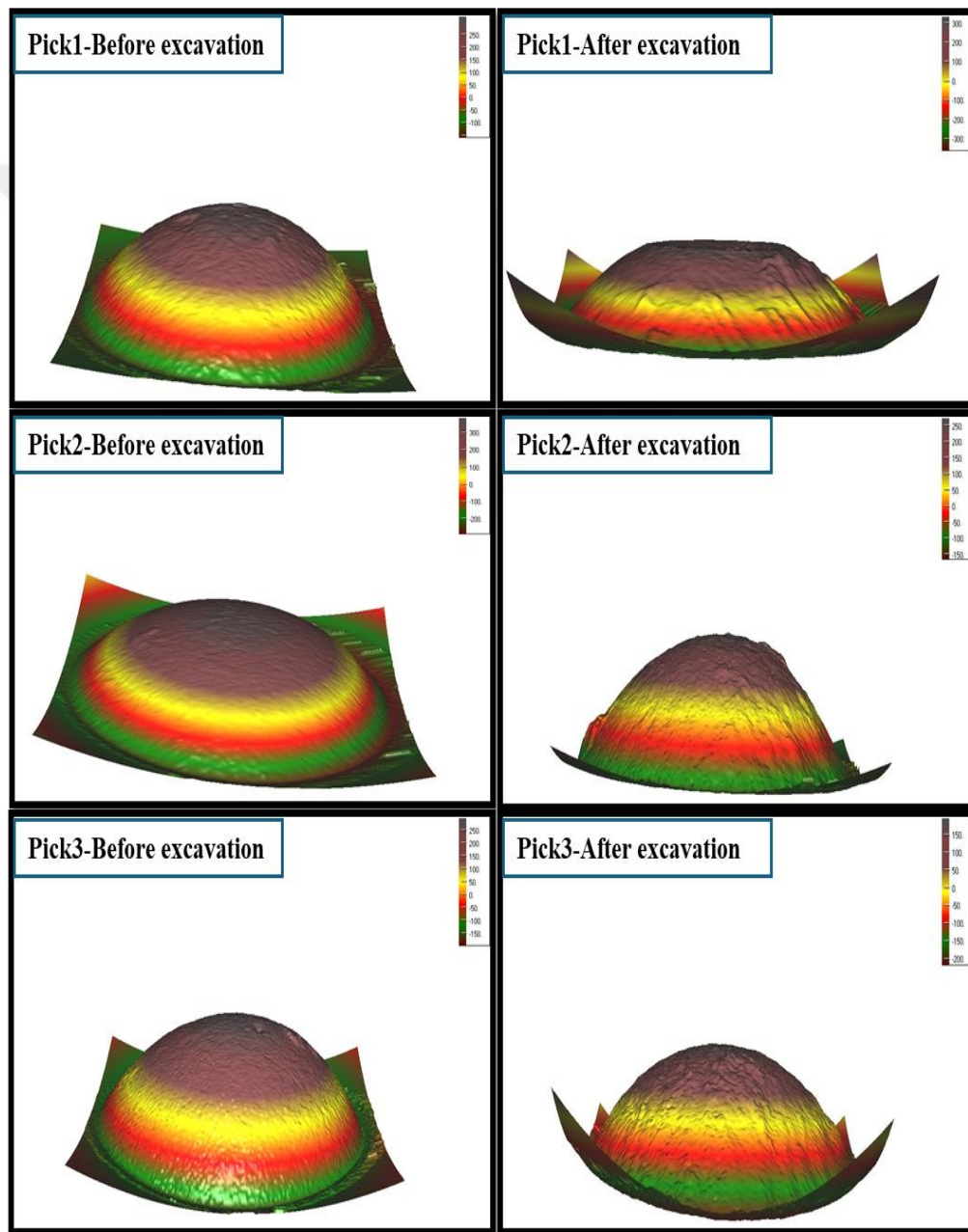


Figure 3.38. Three-dimensional surface maps of the tungsten-carbide tips of Picks 1–3 before and after excavation.

Visual inspection of the picks after testing (Figure 3.39) provides complementary evidence of wear. Beyond alterations to the carbide tip, visible wear marks were observed on the surrounding steel body. These marks were most noticeable on Pick 2, moderate on Pick 1 and minimal on Pick 3, which is consistent with the mass-loss and 3-D mapping results. The wear pattern did not reflect the compressive strength ranking of the mix designs. Instead, the trend aligned more closely with the sand-to-cement ratios.



Figure 3.39. Visual condition of Picks 1–3 after cutting tests.

4. CONCLUSIONS

This thesis sets out to reduce specific energy and vibration in a laboratory scale rotary rock cutting head using a data-driven pipeline that links sensing, prediction, and optimization. The work addressed three gaps identified at the outset, a reliable procedure to determine the dominant vibration direction during cutting, data-driven vibration thresholds that separate three strength classes, and supervised learning models that map controllable set-points to SE and vibration for subsequent optimization. The research was guided by four questions on vibration direction, the diagnostic value of vibration features, the joint influence of depth, rotation, and table speed on SE and vibration, and the choice and robustness of predictive models.

4.1. Answers to the Research Questions

RQ1, dominant vibration direction. Across mixes and operating conditions, the X-axis exhibited the largest vibration magnitudes during cutting, and this dominance was stable within the three UCS classes. This validates the starting hypothesis that the machine-frame X-axis is the most informative direction for real-time monitoring and control.

RQ2, vibration features as proxies. Vibration features were used as practical proxies for material response. The dominant vibration axis provided clearer separation among materials and operating regimes. Two data-driven vibration thresholds were learned to minimize misclassification across the three SE-based classes. These thresholds, $t_1 = 86 \text{ mm/s}$ and $t_2 = 141.2 \text{ mm/s}$, partitioned the measurements into bands consistent with expected hardness and energy levels. They were visualized and applied to trigger class-conditioned operating recommendations. These results show that directional vibration measurements offer a reliable indicator of both hardness and specific energy.

RQ3, joint effects of depth, rotation, and table speed. The fitted response surfaces and the trade-off analyses showed the expected interactions. Increasing depth within the mechanically stable range reduced SE and increased vibration, with diminishing returns at higher penetrations. Rotation affected both targets through chip formation and pick engagement, while table speed mediated exposure time and dynamic loading. The joint particle-swarm search, run over a 9-variable decision vector with cross-sample ordering constraints, consistently returned feasible, monotone settings. Under balanced weights the

recommended depths increased from Mix-1 to Mix-3, rotation clustered near the observed high level, and table remained at the nominal middle level, yielding decreasing SE and vibration from Mix-1 to Mix-3. When vibration was prioritized the optimizer reduced depths and, where allowed by the discrete grid, rotation, at a cost in SE. When energy was prioritized it increased depths and, for Mix-1, selected a slower table, accepting higher vibration where thresholds allowed. These patterns match rock-cutting theory and confirm that meaningful operating compromises can be computed within realistic actuator grids.

RQ4, predictive models and robustness. For specific energy, tree-based models showed the highest accuracy across all mixes. Gradient Boosting performed best for Mix-1 and Mix-3, with R^2 values of about 0.93 and 0.97, while Random Forest was best for Mix-2 with an R^2 of about 0.95. For Vibration-X, model performance was more modest. Lasso gave the best results for Mix-1 and Mix-3, with R^2 values near 0.34 and 0.47, and Random Forest performed best for Mix-2 with an R^2 of about 0.66. Although Vibration-X was harder to predict than specific energy, these surrogate models, combined with the learned vibration thresholds, were sufficiently accurate to support the multi-objective optimization and the mapping from vibration measurements to recommended control settings.

4.2. Limitations

The dataset is laboratory-scale with discrete actuator levels and controlled materials, so dynamics such as tool wear, thermal effects, and rock mass structure were only partially represented. The vibration model used a single dominant axis, which simplifies sensing but ignores cross-axis coupling.

4.3. Future Research

1. Field-scale validation and scaling laws. Collect multi-modal data on a production roadheader to assess transfer from laboratory to site and to calibrate depth, rotation, and traverse scaling.
2. Closed-loop control. Embed the vibration-to-setting logic inside a real controller, for example model predictive control informed by the surrogates, and compare against operator baselines for energy, vibration, and advance.

3. Pick wear and health modeling. Add wear-state estimators and bearing-health diagnostics to decouple geological from mechanical causes of vibration, then include a wear-cost term in the optimization objective.
4. Richer sensing and fusion. Extend beyond the dominant axis to tri-axial vibration, motor current spectra, and torque, using sensor fusion to improve Vibration-X prediction and threshold stability.
5. Uncertainty and safety envelopes. Quantify prediction and optimization uncertainty, for example with conformal prediction or Bayesian ensembles, and enforce chance constraints on vibration and power.
6. Hybrid physics-ML models. Combine chip-formation mechanics with learned components to regularize extrapolation, especially when actuator settings or materials leave the training manifold.
7. Adaptive domain transfer. Use domain adaptation to move models between rock types and machines, benchmarking how much site-specific data is needed to meet accuracy targets.
8. Trajectory and sequencing. Couple set-point optimization with cut-sequence and path planning so that energy and vibration are minimized at the cycle level, not only at a static operating point.

Taken together, the thesis demonstrates that vibration-informed machine learning, combined with constrained swarm search, can compute feasible, high-quality operating settings that reduce energy while respecting vibration limits. The proposed future program focuses on transferring these gains to the field and closing the loop for practical, safe, and efficient excavation.

5. REFERENCES

- Akbasli, I. T. (2024, June 24). *Particle Swarm Optimization Benchmark tests*. <https://www.kaggle.com/code/zzettrkalpakbal/particle-swarm-optimization-benchmark-tests?scriptVersionId=99242386&cellId=1>
- Akhlat, Y., Manzali, Y., Chahhou, M., & Zinedine, A. (2021). A New Noisy Random Forest Based Method for Feature Selection. *Cybernetics and Information Technologies*, 21(2), 10–28. <https://doi.org/10.2478/cait-2021-0016>
- Arooj, S., Rehman, S. ur, Imran, A., Almuhaimeed, A., Alzahrani, A. K., & Alzahrani, A. (2022). A Deep Convolutional Neural Network for the Early Detection of Heart Disease. *Biomedicines*, 10(11), 2796. <https://doi.org/10.3390/biomedicines10112796>
- Arulkumaran, K., Deisenroth, M. P., Brundage, M., & Bharath, A. A. (2017). Deep Reinforcement Learning: A Brief Survey. *IEEE Signal Processing Magazine*, 34(6), 26–38. <https://doi.org/10.1109/MSP.2017.2743240>
- ASTM. (2021). *Test Method for Compressive Strength of Hydraulic Cement Mortars (Using 2-in. or [50mm] Cube Specimens)*. https://doi.org/10.1520/C0109_C0109M-21
- Averin, E., Zhabin, A., Polyakov, A., Linnik, Y., & Linnik, V. (2021). Transition between relieved and unrelieved modes when cutting rocks with conical picks. *Journal of Mining Institute*, 249, 329–333. <https://doi.org/10.31897/PMI.2021.3.1>
- Aydin, G. (2012). *Kesici Disklerin Kayaç Kesme Performansının Araştırılması ve Performans Tahminine Yönelik Modellerin Geliştirilmesi* [PhD Thesis, Karadeniz Technical University].
- Balci, C., Demircin, M. A., Copur, H., & Tuncdemir, H. (2004). Estimation of optimum specific energy based on rock properties for assessment of roadheader performance. *The South African Institute of Mining and Metallurgy*, 104, 633–642.
- Bilgin, N., Copur, H., & Balci, C. (2013). *Mechanical Excavation in Mining and Civil Industries*. CRC Press. <https://doi.org/10.1201/b16083>
- Bilgin, N., Demircin, M. A., Copur, H., Balci, C., Tuncdemir, H., & Akcin, N. (2006). Dominant rock properties affecting the performance of conical picks and the comparison of some experimental and theoretical results. *International Journal of Rock Mechanics and Mining Sciences*, 43(1), 139–156. <https://doi.org/10.1016/j.ijrmms.2005.04.009>
- Bilgin, N., Dincer, T., Copur, H., & Erdogan, M. (2004). Some geological and geotechnical factors affecting the performance of a roadheader in an inclined tunnel. *Tunnelling and Underground Space Technology*, 19(6), 629–636. <https://doi.org/10.1016/j.tust.2004.04.004>

- Bilgin, N., Seyrek, T., Shahriar, K., & Erding, E. (1991). Roadheaders glean valuable tips for Istanbul Metro. *International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts*, 28(2–3), A190. [https://doi.org/10.1016/0148-9062\(91\)93166-4](https://doi.org/10.1016/0148-9062(91)93166-4)
- Box, G. E. P., Hunter, J. Stuart., & Hunter, W. Gordon. (2005). *Statistics for experimenters : design, innovation, and discovery*. Wiley-Interscience.
- Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2), 123–140. <https://doi.org/10.1007/BF00058655>
- Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), 5–32. <https://doi.org/10.1023/A:1010933404324>
- Bühlmann, P., & Hothorn, T. (2007). Boosting algorithms: Regularization, prediction and model fitting. *Statistical Science*, 22(4), 477–505. <https://doi.org/10.1214/07-STS242>
- Bühlmann, P., & van de Geer, S. (2011). *Statistics for High-Dimensional Data*. Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-642-20192-9>
- Cheluszka, P. (2021a). Numerical studies of the dynamics of the roadheader equipped with an automatic control system. *Energies*, 14, 3892. <https://doi.org/10.3390/en14133892>
- Cheluszka, P. (2021b). Numerical studies of the dynamics of the roadheader equipped with an automatic control system during cutting of rocks with different mechanical properties. *Energies*, 14(21). <https://doi.org/10.3390/en14217353>
- Cheluszka, P., & Jagieła-Zajac, A. (2022). Validation of a Method for Measuring the Position of Pick Holders on a Robotically Assisted Mining Machine's Working Unit. *Energies*, 15(1). <https://doi.org/10.3390/en15010295>
- Cheluszka, P., & Mann, R. (2019). Vibration identification of the roadheader cutting head using high-speed cameras. *MATEC Web of Conferences*, 252, 03018. <https://doi.org/10.1051/mateconf/201925203018>
- Cheluszka, P., Mikuła, S., Mikuła, J., & Fries, J. (2024). Improving the operational durability of tangential-rotary picks for rock cutting with central spraying and hydrodynamic rotation. *Scientific Reports*, 14(1). <https://doi.org/10.1038/s41598-024-63733-1>
- Cheluszka, P., Remiorz, E., & Rostami, J. (2022). The Use of a Roadheader Simulator in Research of Dynamics and Energy-Consumption of Excavating Underground Roadways and Tunnels. *Energies*, 15(18), 6673. <https://doi.org/10.3390/en15186673>
- Cheluszka, P., Sobota, P., & Głuszek, G. (2019). Studies of behaviour of the automatic control system of roadheader cutting heads movement. *MATEC Web of Conferences*, 252, 09002. <https://doi.org/10.1051/mateconf/201925209002>
- Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. <https://doi.org/10.1145/2939672.2939785>

- Choudhary, B., Sharma, D., & Das, M. (2017). A Study to Investigate the Influence of Coal Properties on Bit Wear-ability of Roadheader. *Modelling, Measurement and Control B*, 86(3), 727–739. https://doi.org/10.18280/mmc_b.860308
- Chourib, I., Guillard, G., Farah, I. R., & Solaiman, B. (2022). Stroke Treatment Prediction Using Features Selection Methods and Machine Learning Classifiers. *IRBM*, 43(6), 678–686. <https://doi.org/10.1016/j.irbm.2022.02.002>
- Comakli, R. (2019). Performance of roadheaders in low strength pyroclastic rocks, a case study of cold storage caverns in Cappadocia. *Tunnelling and Underground Space Technology*, 89, 179–188. <https://doi.org/10.1016/j.tust.2019.04.007>
- Çomaklı, R., Kahraman, S., & Balci, C. (2014). Performance prediction of roadheaders in metallic ore excavation. *Tunnelling and Underground Space Technology*, 40, 38–45. <https://doi.org/10.1016/j.tust.2013.09.006>
- Çopur, H., Özdemir, L., & Rostami, J. (1998). *Roadheader Applications In Mining And Tunneling Industries*.
- Deng, H., Zhou, Y., Wang, L., & Zhang, C. (2021). Ensemble learning for the early prediction of neonatal jaundice with genetic features. *BMC Medical Informatics and Decision Making*, 21(1), 338. <https://doi.org/10.1186/s12911-021-01701-9>
- Deshmukh, S., Raina, A. K., Murthy, V. M. S. R., Trivedi, R., & Vajre, R. (2020). Roadheader—A comprehensive review. *Tunnelling and Underground Space Technology*, 95, 103148. <https://doi.org/10.1016/j.tust.2019.103148>
- Deyi, Z., Songyong, L., Shihang, L., Hao, L., Qixin, Z., & Xuemei, N. (2024). Compensation control for roadheader's pitch and yaw position pose deviation based on cutting boom motion. *Tunnelling and Underground Space Technology*, 150. <https://doi.org/10.1016/j.tust.2024.105839>
- Dibavar, B., Kahraman, S., Rostami, M., & Fener, M. (2023). A new rock mass cuttability classification for roadheaders used in coal mining. *Mining, Metallurgy & Exploration*, 40, 1141–1152. <https://doi.org/10.1007/s42461-023-00788-0>
- Ding, J., Du, J., Wang, H., & Xiao, S. (2025). A novel two-stage feature selection method based on random forest and improved genetic algorithm for enhancing classification in machine learning. *Scientific Reports*, 15(1), 16828. <https://doi.org/10.1038/s41598-025-01761-1>
- Dolipski, M., Cheluszka, P., & Sobota, P. (2015). Investigating the simulated control of the rotational speed of roadheader cutting heads, relating to the reduction of energy consumption during the cutting process. *Journal of Mining Science*, 51(2), 298–308. <https://doi.org/10.1134/S106273911502012X>

- Dong, Z., Zhang, X., Yang, W., Lei, M., Zhang, C., Wan, J., & Han, L. (2022). Automatic Cutting Speed Control System of Boom-Type Roadheader Based on Geological Strength Index. *Minerals*, 12(12). <https://doi.org/10.3390/min12121582>
- Eberhart, R., & Kennedy, J. (1995). A new optimizer using particle swarm theory. *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, 39–43. <https://doi.org/10.1109/MHS.1995.494215>
- Ebrahimabadi, A., Azimipour, M., & Bahreini, A. (2015). Prediction of roadheaders' performance using artificial neural network approaches (MLP and KOSFM). *Journal of Rock Mechanics and Geotechnical Engineering*, 7(5), 573–583. <https://doi.org/10.1016/j.jrmge.2015.06.008>
- Ebrahimabadi, A., Goshtasbi, K., Shahriar, K., & Cheraghi Seifabad, M. (2011a). A model to predict the performance of roadheaders based on the rock mass brittleness index. *Journal of the Southern African Institute of Mining and Metallurgy*, 111(5), 355–364.
- Ebrahimabadi, A., Goshtasbi, K., Shahriar, K., & Cheraghi Seifabad, M. (2011b). Predictive Models for Roadheaders' Cutting Performance in Coal Measure Rocks. *Yerbilimleri*, 32(2), 141–168. <https://dergipark.org.tr/tr/pub/yerbilimleri/issue/13646/165202>
- Ebrahimabadi, A., Goshtasbi, K., Shahriar, K., & Seifabad, M. C. (2012). A Universal Model to Predict Roadheaders' Cutting Performance / Uniwersalny Model Do Prognozowania Postępu Prac Kombajnów Do Drażenia Tuneli. *Archives of Mining Sciences*, 57(4), 1015–1026. <https://doi.org/10.2478/v10267-012-0067-5>
- Elsheikh, A. H., & Abd Elaziz, M. (2019). Review on applications of particle swarm optimization in solar energy systems. *International Journal of Environmental Science and Technology*, 16(2), 1159–1170. <https://doi.org/10.1007/s13762-018-1970-x>
- Emmert-Streib, F., & Dehmer, M. (2019). High-Dimensional LASSO-Based Computational Regression Models: Regularization, Shrinkage, and Selection. *Machine Learning and Knowledge Extraction*, 1(1), 359–383. <https://doi.org/10.3390/make1010021>
- Ericsson, L., Gouk, H., Loy, C. C., & Hospedales, T. M. (2022). Self-Supervised Representation Learning: Introduction, advances, and challenges. *IEEE Signal Processing Magazine*, 39(3), 42–62. <https://doi.org/10.1109/MSP.2021.3134634>
- Fan, Q., Chen, C., Zhang, Q., & Liu, G. (2023). A dynamic analysis of a conical pick during rock-cutting process based on the smooth finite element method. *Wear*, 512–513, 204523. <https://doi.org/10.1016/j.wear.2022.204523>
- Feng, L., & Changqing, Y. (2021). Self-adaptive Adjustment System for Swing Speed of Boom-type Roadheader. *2021 IEEE 2nd International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA)*, 1020–1027. <https://doi.org/10.1109/ICIBA52610.2021.9688228>

- Friedman, J. H. (2001). 999 REITZ LECTURE GREEDY FUNCTION APPROXIMATION: A GRADIENT BOOSTING MACHINE 1. In *The Annals of Statistics* (Vol. 29, Issue 5).
- Friedman, J. H. (2002). Stochastic gradient boosting. In *Computational Statistics & Data Analysis* (Vol. 38). www.elsevier.com/locate/cstda
- Gad, A. G. (2022). Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review. *Archives of Computational Methods in Engineering*, 29(5), 2531–2561. <https://doi.org/10.1007/s11831-021-09694-4>
- GeeksforGeeks. (2023). *Machine learning: Linear regression*. GeeksforGeeks. <https://www.geeksforgeeks.org/machine-learning/ml-linear-regression/>
- GeeksforGeeks. (2025, September 1). *Random Forest Classifier using Scikit-learn*. <https://www.geeksforgeeks.org/dsa/random-forest-classifier-using-scikit-learn/>
- Gehring, K. H. (1989). Cutting comparison. *Tunnels and Tunnelling*, 21(11), 27–30.
- Guo, D., Song, Z., Liu, N., Xu, T., Wang, X., Zhang, Y., Su, W., & Cheng, Y. (2024). Performance Study of Hard Rock Cantilever Roadheader Based on PCA and DBN. *Rock Mechanics and Rock Engineering*, 57(4), 2605–2623. <https://doi.org/10.1007/s00603-023-03698-1>
- Hastie, Trevor., Friedman, J. H. ., & Tibshirani, Robert. (2009). *The elements of statistical learning : data mining, inference, and prediction*. Springer.
- Hekimoglu, O. Z., & Fowell, R. J. (1991). Theoretical and practical aspects of circumferential pick spacing on boom tunnelling machine cutting heads. *Mining Science and Technology*, 13(3), 257–270. [https://doi.org/10.1016/0167-9031\(91\)90397-U](https://doi.org/10.1016/0167-9031(91)90397-U)
- Heyduk, A., & Joostberens, J. (2020). Automatic control of roadheader cutting head speed and load torque. *IOP Conference Series: Earth and Environmental Science*, 609(1). <https://doi.org/10.1088/1755-1315/609/1/012081>
- Hoerl, R. W. (2020). Ridge Regression: A Historical Context. *Technometrics*, 62(4), 420–425. <https://doi.org/10.1080/00401706.2020.1742207>
- Huang, S.-C., Pareek, A., Jensen, M., Lungren, M. P., Yeung, S., & Chaudhari, A. S. (2023). Self-supervised learning for medical image classification: a systematic review and implementation guidelines. *Npj Digital Medicine*, 6(1), 74. <https://doi.org/10.1038/s41746-023-00811-0>
- Jasiulek, D., & Świder, J. (2013). Mechatronic systems in mining roadheaders – examples of solutions. In *Pomiary Automatyka Robotyka nr* (Vol. 1).
- Jeong, H., Choi, S., Lee, S., & Jeon, S. (2020). Rock Cutting Simulation of Point Attack Picks Using the Smooth Particle Hydrodynamics Technique and the Cumulative Damage Model. *Applied Sciences*, 10(15), 5314. <https://doi.org/10.3390/app10155314>

- Joostberens, J., & Heyduk, A. (2018). Laboratory Tests of a Speed Control System for Roadheader Cutting Heads. *Multidisciplinary Aspects of Production Engineering*, 1(1), 153–159. <https://doi.org/10.2478/mape-2018-0020>
- Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245), 255–260. <https://doi.org/10.1126/science.aaa8415>
- Jurczuk, K., Czajkowski, M., & Kretowski, M. (2023). From Random Forest to an interpretable decision tree - An evolutionary approach. *Proceedings of the Companion Conference on Genetic and Evolutionary Computation*, 291–294. <https://doi.org/10.1145/3583133.3590732>
- Kahraman, E., & Kahraman, S. (2016). The performance prediction of roadheaders from easy testing methods. *Bulletin of Engineering Geology and the Environment*, 75(4), 1585–1596. <https://doi.org/10.1007/s10064-015-0801-2>
- Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017, December 4). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. *31st Conference on Neural Information Processing Systems (NIPS 2017)*. <https://github.com/Microsoft/LightGBM>.
- Keleş, S. (2005). *Cutting performance assessment of a medium weight roadheader at Çayirhan coal mine*. Middle East Technical University.
- Kennedy, J. (2011). Particle Swarm Optimization. In *Encyclopedia of Machine Learning* (pp. 760–766). Springer US. https://doi.org/10.1007/978-0-387-30164-8_630
- Kim, H., Hwang, S., Yoo, W., Kim, W., Kim, C., & Yoo, H. (2024). Laboratory-Scale Limestone Rock Linear Cutting Tests with a Conical Pick: Predicting Optimal Cutting Conditions from Tool Forces. *Buildings*, 14(9), 2772. <https://doi.org/10.3390/buildings14092772>
- Kotwica, K. (2018). Atypical and innovative tool, holder and mining head designed for roadheaders used to tunnel and gallery drilling in hard rock. *Tunnelling and Underground Space Technology*, 82, 493–503. <https://doi.org/10.1016/j.tust.2018.08.017>
- Krauze, M., Wodecki, J., Gołębiowski, P., Mertuszka, M., & Jakubowski, J. (2022). The use of a roadheader simulator in research of dynamics and energy-consumption of excavating underground roadways and tunnels. *Energies*, 15, 6673.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>
- Li, C., Zhao, D., Cao, C., Lyu, F., Zhang, M., & Wu, M. (2022). The coupling relationship analysis and control of the roadheader's cutting and supporting structures. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 44(9), 437. <https://doi.org/10.1007/s40430-022-03671-z>

- Li, F., Li, Y., Yan, C., Ma, C., Liu, C., & Suo, Q. (2022). Swing Speed Control Strategy of Fuzzy PID Roadheader Based on PSO-BP Algorithm. *2022 IEEE 6th Information Technology and Mechatronics Engineering Conference (ITOEC)*, 827–833. <https://doi.org/10.1109/ITOEC53115.2022.9734429>
- Liu, Q., Liu, S., Dai, Q., Yu, X., Teng, D., & Wei, M. (2021). Data-Driven Approaches for Diagnosis of Incipient Faults in Cutting Arms of the Roadheader. *Shock and Vibration*, 2021(1). <https://doi.org/10.1155/2021/8865068>
- Liu, Y., Wang, Y., & Zhang, J. (2012). *New Machine Learning Algorithm: Random Forest* (pp. 246–252). https://doi.org/10.1007/978-3-642-34062-8_32
- Manzali, Y., Akhiat, Y., Chahhou, M., Elmohajir, M., & Zinedine, A. (2023). Reducing the number of trees in a forest using noisy features. *Evolving Systems*, 14(2), 157–174. <https://doi.org/10.1007/s12530-022-09441-5>
- Manzali, Y., & Elfar, M. (2023). Random Forest Pruning Techniques: A Recent Review. *Operations Research Forum*, 4(2), 43. <https://doi.org/10.1007/s43069-023-00223-6>
- Millonas, M. M. (1993). *Swarms, Phase Transitions, and Collective Intelligence*.
- Montgomery, D. (2017). *Design and analysis of experiments*. John Wiley & Sons, Inc.
- Montgomery, D., Peck, E., & Vining, G. (2012). *Introduction to Linear Regression Analysis, 5th Edition*. John Wiley & Sons.
- Muthukrishnan, R., & Rohini, R. (2016). LASSO: A feature selection technique in predictive modeling for machine learning. *2016 IEEE International Conference on Advances in Computer Applications (ICACA)*, 18–20. <https://doi.org/10.1109/ICACA.2016.7887916>
- Myers, R., Montgomery, D., & Anderson-Cook, C. (2009). *Response surface methodology : process and product optimization using designed experiments*. Wiley.
- Pawlik, J., Pytel, M., & Góral, T. (2022). The volumetric wear assessment of a mining conical pick using the photogrammetric approach. *Materials*, 15, 6016. <https://doi.org/10.3390/ma15176016>
- Peinsitt, T., Haubmann, H., Kargl, H., & Kary, C. (2020). Recent developments towards autonomous tunneling and mining machinery. In *Tunnels and Underground Cities: Engineering and Innovation Meet Archaeology, Architecture and Art (WTC 2019)* (pp. 2849–2858). CRC Press. <https://doi.org/10.4324/9781003031635-32>
- Poli, R., Kennedy, J., & Blackwell, T. (2007). Particle swarm optimization. *Swarm Intelligence*, 1(1), 33–57. <https://doi.org/10.1007/s11721-007-0002-0>

- Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2019). CatBoost: unbiased boosting with categorical features. *Advances in Neural Information Processing Systems*, 31, 6638–6648.
- Qi, P., Chang, J., Chen, X., Wang, T., & Wu, M. (2022). Identification of Rock Properties of Rock Wall Cut by Roadheader Based on PSO-VMD-LSSVM. *Frontiers in Earth Science*, 10. <https://doi.org/10.3389/feart.2022.884633>
- Qu, X., & Zhang, Y. (2023). Fault Diagnosis Method of Roadheader Bearing Based on VMD and Domain Adaptive Transfer Learning. *Sensors*, 23(11). <https://doi.org/10.3390/s23115134>
- Qu, Y., Yang, T., Li, T., Zhan, Y., & Fu, S. (2022). Path Tracking of Underground Mining Boom Roadheader Combining BP Neural Network and State Estimation. *Applied Sciences*, 12(10), 5165. <https://doi.org/10.3390/app12105165>
- Restner, U., & Plinninger, R. J. (2015). Rock mechanical aspects of roadheader excavation. *ISRM Regional Symposium, EUROCK 2015*, 249–254. <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85055162560&partnerID=40&md5=cb90c736069a8bcc66e68d6cb7a08fb0>
- Rostami, J., Özdemir, L., & Neil, D. M. (1994). Performance prediction: A key issue in mechanical hard rock mining. *Mining Engineering*, 11, 1263–1267.
- Salsani, A., Daneshian, J., Shariati, S., Yazdani-Chamzini, A., & Taheri, M. (2014). Predicting roadheader performance by using artificial neural network. *Neural Computing and Applications*, 24(7–8), 1823–1831. <https://doi.org/10.1007/s00521-013-1434-7>
- Sarker, I. H. (2021). Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Computer Science*, 2(3), 160. <https://doi.org/10.1007/s42979-021-00592-x>
- Seker, S. E., & Ocak, I. (2019). Performance prediction of roadheaders using ensemble machine learning techniques. *Neural Computing and Applications*, 31(4), 1103–1116. <https://doi.org/10.1007/s00521-017-3141-2>
- Sifferlinger, N. A., Hartlieb, P., & Moser, P. (2017). The Importance of Research on Alternative and Hybrid Rock Extraction Methods. *BHM Berg- Und Hüttenmännische Monatshefte*, 162(2), 58–66. <https://doi.org/10.1007/s00501-017-0574-y>
- Škvareková, D., Rudnó, J., & Andrassyová, Z. (2021). Evaluation and risk factors of roadheaders in coal mines. *Management Systems in Production Engineering*, 29(4), 314–319. <https://doi.org/10.2478/mspe-2021-0030>
- Slouka, S., Brune, J., & Rostami, J. (2022). Characterization of respirable dust generated from full-scale laboratory igneous rock cutting tests with conical picks at two stages of wear. *Mining, Metallurgy & Exploration*, 39, 1801–1809. <https://doi.org/10.1007/s42461-022-00625-w>

- Tang, C., Abbatematteo, B., Hu, J., Chandra, R., Martín-Martín, R., & Stone, P. (2025). Deep Reinforcement Learning for Robotics: A Survey of Real-World Successes. *Annual Review of Control, Robotics, and Autonomous Systems*, 8(1), 153–188. <https://doi.org/10.1146/annurev-control-030323-022510>
- Tatiya, Ratan. (2017). *Civil excavations and tunnelling : a practical guide*. ICE Publishing.
- Teale, R. (1965). The concept of specific energy in rock drilling. *International Journal of Rock Mechanics and Mining Sciences*, 2(1), 57–73. [https://doi.org/10.1016/0148-9062\(65\)90022-7](https://doi.org/10.1016/0148-9062(65)90022-7)
- Thuro, K., & Plinninger, R. J. (1999). Roadheader excavation performance-geological and geotechnical influences. *9th ISRM Congress*, 0–4.
- Ulusay, R., & Hudson, J. A. (2007). *The complete ISRM suggested methods for rock characterization, testing and monitoring: 1974–2006*. ISRM Turkish National Group Ankara.
- Wang, D., Wang, P., Li, Y., Guo, M., Zheng, W., Shen, Y., & Wu, M. (2022). Research on Adaptive Cutting Control Strategy of Roadheader Cutting Arms. *China Mechanical Engineering*, 33(20), 2492–2501. <https://www.cmemo.org.cn/CN/10.3969/j.issn.1004-132X.2022.20.013>
- Wang, H. (2023). Research on the Application of Random Forest-based Feature Selection Algorithm in Data Mining Experiments. *International Journal of Advanced Computer Science and Applications*, 14(10). <https://doi.org/10.14569/IJACSA.2023.0141054>
- Wang, H., Sun, D., & Qin, D. (2017). A new continuously variable transmission system applied to transmission system of the roadheader's cutting unit. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 231(19), 3590–3600. <https://doi.org/10.1177/0954406216649404>
- Wang, P., Shen, Y., Ji, X., Zong, K., Zheng, W., Wang, D., & Wu, M. (2021a). Multiparameter Control Strategy and Method for Cutting Arm of Roadheader. *Shock and Vibration*, 2021(1). <https://doi.org/10.1155/2021/9918988>
- Wang, P., Shen, Y., Ji, X., Zong, K., Zheng, W., Wang, D., & Wu, M. (2021b). Multiparameter Control Strategy and Method for Cutting Arm of Roadheader. *Shock and Vibration*, 2021(1). <https://doi.org/10.1155/2021/9918988>
- Wang, P., Shen, Y., Li, R., Zong, K., Fu, S., & Wu, M. (2021a). Multisensor information-based adaptive control method for cutting head speed of roadheader. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 235(11), 1941–1955. <https://doi.org/10.1177/0954406220952494>
- Wang, P., Shen, Y., Li, R., Zong, K., Fu, S., & Wu, M. (2021b). Multisensor information-based adaptive control method for cutting head speed of roadheader. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 235(11), 1941–1955. <https://doi.org/10.1177/0954406220952494>

- Wang, P., Yang, Y., & Wang, D. (2021). Intelligent cutting control system and method of coal and gangue in robotic roadheader. *Journal of China Coal Society*, 46(S2), 1124–1134. <https://www.mtxb.com.cn/cn/article/Y2021/IS2/1124>
- Wang, S., & Wu, M. (2019). Cutting trajectory planning of sections with complex composition for roadheader. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 233(4), 1441–1452. <https://doi.org/10.1177/0954406218768840>
- Wang, X., & Su, O. (2019). Specific energy analysis of rock cutting based on fracture mechanics: A case study using a conical pick on sandstone. *Engineering Fracture Mechanics*, 213, 197–205. <https://doi.org/10.1016/j.engfracmech.2019.04.010>
- Wang, X., Su, O., & Wang, Q. (2022). Distribution characteristics of rock chips under relieved and unrelieved cutting conditions. *International Journal of Rock Mechanics and Mining Sciences*, 151, 105048. <https://doi.org/10.1016/j.ijrmms.2022.105048>
- Wang, X., Wang, Q.-F., Liang, Y.-P., Su, O., & Yang, L. (2018). Dominant Cutting Parameters Affecting the Specific Energy of Selected Sandstones when Using Conical Picks and the Development of Empirical Prediction Models. *Rock Mechanics and Rock Engineering*, 51(10), 3111–3128. <https://doi.org/10.1007/s00603-018-1522-1>
- Wu, J., Xu, Z., Fang, X., Shi, G., & Wang, H. (2023). Research on Path Planning and Control Method for Secondary Autonomous Cutting of Cantilever Roadheader in a Large-Section Coal Roadway. *Sustainability (Switzerland)*, 15(1). <https://doi.org/10.3390/su15010560>
- Yang, J., Tang, Z., Wang, X., Wang, Z., & Wu, M. (2019). Roadheader Anomaly Detection Method Based on VSAPSO-BP Under the Single Category Learning. *Journal of Vibration, Measurement and Diagnosis*, 39(1), 130–135.
- Zhang, C., Zhang, X., Yang, W., Zhang, G., Wan, J., Lei, M., & Dong, Z. (2025). Safe Path Planning Method Based on Collision Prediction for Robotic Roadheader in Narrow Tunnels. *Mathematics*, 13(3). <https://doi.org/10.3390/math13030522>
- Zhang, J., Wan, C., Wang, J., Chen, C., Wang, T., Zhang, R., & Guo, H. (2024). Research on the “shape-performance-control” integrated digital twin system for boom-type roadheaders. *Scientific Reports*, 14(1). <https://doi.org/10.1038/s41598-024-56539-8>
- Zhang, L., Tian, M., Song, J., He, Y., Feng, J., & Liu, X. (2020). Hardness recognition method of roadheader’s cutting rock wall based on multi-source data fusion. *Journal of Vibration and Shock*, 39(13), 7–15.
- Zhang, M., Cai, X., Lv, F., Zong, K., & Wu, M. (2018). Research on roadheader auto rectification in limited roadway space based on regional grid. *Chinese Journal of Scientific Instrument*, 39(3), 62–70.

- Zhang, M., Cheng, R., & Zhu, Y. (2021). Study on roadheader rectification running performance and control in the in-cline coalmine roadway. *Journal of China Coal Society*, 46(S1), 549–557.
- Zhang, M., Zhao, L., & Shi, B. (2023). Analysis and construction of the coal and rock cutting state identification system in coal mine intelligent mining. *Scientific Reports*, 13(1). <https://doi.org/10.1038/s41598-023-30617-9>
- Zhang, Q., Fan, Q., Gao, H., Wu, Y., & Xu, F. (2022). A study on pick cutting properties with full-scale rotary cutting experiments and numerical simulations. *PLOS ONE*, 17(4), e0266872. <https://doi.org/10.1371/journal.pone.0266872>
- Zhang, X., Zheng, X., Cheng, R., Qiu, J., & Jin, Y. (2018). A competitive mechanism based multi-objective particle swarm optimizer with fast convergence. *Information Sciences*, 427, 63–76. <https://doi.org/10.1016/j.ins.2017.10.037>
- Zhang, Z., Chen, S., Liu, Y., Wang, H., & Cao, C. (2022a). Cutting Mechanism Rotor System Dynamic Characteristics of Cantilever Roadheader under Random Hard Rock Load. *Actuators*, 11(3), 83. <https://doi.org/10.3390/act11030083>
- Zhang, Z., Chen, S., Liu, Y., Wang, H., & Cao, C. (2022b). Cutting Mechanism Rotor System Dynamic Characteristics of Cantilever Roadheader under Random Hard Rock Load. *Actuators*, 11(3), 83. <https://doi.org/10.3390/act11030083>
- Zhao, Z. M., Liu, R. H., & Zhao, J. (2014). Research on the Pattern of Boom-Type Roadheader Intelligent Control System. *Applied Mechanics and Materials*, 599–601, 1048–1051. <https://doi.org/10.4028/www.scientific.net/AMM.599-601.1048>
- Zhou, J., Dai, Y., Huang, S., Armaghani, D. J., & Qiu, Y. (2023). Proposing several hybrid SSA—machine learning techniques for estimating rock cuttability by conical pick with relieved cutting modes. *Acta Geotechnica*, 18(3), 1431–1446. <https://doi.org/10.1007/s11440-022-01685-4>
- Zhu, Y., Xie, M., & Wang, H. (2022). Prediction method of cutting stability of roadheader based on the Newton–Lagrange mixed discrete method. *Scientific Reports*, 12(1), 20760. <https://doi.org/10.1038/s41598-022-25409-6>
- Zong, K., Fu, S., & Wu, M. (2021). Simulation of control strategy for swing speed of roadheader's cutting arm based on GA-BP network. *Journal of China Coal Society*, 46(S1), 511–519.

6. APPENDIX

Rock Cutting Dataset for All Three Mixes

Mix-1-Rep1

Test Number	d _{real} (mm)	W _{eff} (mm)	Rotational Speed (m/s)	Table Speed (cm/min)	MAX Vibration-X (mm/s)	MAX Vibration-Z (mm/s)	MAX Power (W)	SE (MJ/m3)
Test1	3.86	6.48	3.4	162.80	161.66	65.62	877.00	1292.63
Test2	4.56	7.65	3.4	187.37	146.79	71.99	869.00	797.44
Test3	4.55	7.64	3.4	211.94	156.48	78.75	831.37	677.43
Test4	6.59	11.06	3.4	162.80	171.46	87.44	945.00	477.87
Test5	6.34	10.64	3.4	187.37	193.68	101.95	1097.00	520.75
Test6	6.8	11.41	3.4	211.94	208.49	127.41	1098.00	400.57
Test7	8.28	13.90	3.4	162.80	264.08	133.88	1177.21	377.09
Test8	7.29	12.23	3.4	187.37	185.58	74.54	1056.00	379.15
Test9	7.7	12.92	3.4	211.94	184.61	81.66	1024.26	291.42
Test10	4.39	7.37	3.78	162.80	93.83	55.05	878.00	1000.50
Test11	4.49	7.54	3.78	187.37	101.15	57.13	877.00	830.07
Test12	4.9	8.22	3.78	211.94	134.99	70.43	959.00	673.78
Test13	6.2	10.40	3.78	162.80	104.45	64.32	953.00	544.45
Test14	6.49	10.89	3.78	187.37	164.33	74.29	943.00	427.20
Test15	7	11.75	3.78	211.94	99.79	50.75	855.00	294.35
Test16	7.96	13.36	3.78	162.80	95.54	61.81	879.00	304.66
Test17	8.19	13.74	3.78	187.37	131.48	66.41	918.00	261.14
Test18	8.41	14.11	3.78	211.94	164.48	75.74	943.00	224.91
Test19	3.61	6.06	4.17	162.80	71.37	67.52	878.00	1479.55
Test20	4.61	7.74	4.17	187.37	82.17	64.94	878.00	788.31
Test21	4.29	7.20	4.17	211.94	100.16	70.39	952.91	873.43
Test22	6.51	10.93	4.17	162.80	66.72	45.90	862.40	446.89
Test23	6.71	11.26	4.17	187.37	214.55	88.34	1027.44	435.43
Test24	6.93	11.63	4.17	211.94	178.74	81.09	1017.59	357.44
Test25	8.54	14.33	4.17	162.80	242.37	99.86	1113.00	335.14
Test26	8.33	13.98	4.17	187.37	178.06	79.79	1185.00	325.86
Test27	8.1	13.59	4.17	211.94	190.00	84.17	1137.95	292.58

Mix-1-Rep2

Test Number	d_{real} (mm)	W_{eff} (mm)	Rotational Speed (m/s)	Table Speed (cm/min)	MAX Vibration-X (mm/s)	MAX Vibration-Z (mm/s)	MAX Power (W)	SE (MJ/m ³)
Test1	4	6.71	3.4	162.80	167.21	95.60	869.00	1192.75
Test2	4.28	7.18	3.4	187.37	183.37	111.99	920.00	958.31
Test3	4.49	7.54	3.4	211.94	265.20	144.88	1149.28	961.67
Test4	6.15	10.32	3.4	162.80	161.46	82.11	949.00	551.02
Test5	6.29	10.56	3.4	187.37	182.77	97.06	1113.00	536.78
Test6	6.39	10.72	3.4	211.94	198.89	126.03	1075.00	444.12
Test7	8.05	13.51	3.4	162.80	233.24	129.79	1097.00	371.76
Test8	8	13.43	3.4	187.37	266.80	142.93	1332.17	397.18
Test9	7.93	13.31	3.4	211.94	202.32	114.26	1196.00	320.83
Test10	4.35	7.30	3.78	162.80	93.22	63.34	859.00	996.93
Test11	4.56	7.65	3.78	187.37	135.93	76.78	959.00	880.02
Test12	4.3	7.22	3.78	211.94	153.10	76.00	930.00	848.47
Test13	6.62	11.11	3.78	162.80	263.01	107.43	1025.00	513.64
Test14	6.32	10.61	3.78	187.37	143.70	97.55	958.00	457.65
Test15	5.53	9.28	3.78	211.94	197.12	113.71	1005.97	554.92
Test16	8.5	14.26	3.78	162.80	265.41	142.33	1088.55	330.87
Test17	8.87	14.89	3.78	187.37	264.63	148.42	1580.00	383.19
Test18	8.85	14.85	3.78	211.94	264.46	146.52	1679.00	361.62
Test19	4.5	7.55	4.17	162.80	173.06	83.82	953.72	1034.30
Test20	4	6.71	4.17	187.37	178.21	78.65	955.00	1138.91
Test21	4.73	7.94	4.17	211.94	149.74	91.36	1032.43	778.45
Test22	5.77	9.68	4.17	162.80	182.92	90.23	956.00	630.60
Test23	6.82	11.45	4.17	187.37	193.01	87.09	1038.00	425.83
Test24	6.13	10.29	4.17	211.94	175.48	98.07	1112.00	499.20
Test25	8.19	13.74	4.17	162.80	163.26	91.39	1032.89	338.17
Test26	8.89	14.92	4.17	187.37	177.75	96.30	1110.00	267.99
Test27	8.29	13.91	4.17	211.94	174.05	96.19	1097.00	269.27

Mix-1-Rep3

Test Number	d _{real} (mm)	W _{eff} (mm)	Rotational Speed (m/s)	Table Speed (cm/min)	MAX Vibration-X (mm/s)	MAX Vibration-Z (mm/s)	MAX Power (W)	SE (MJ/m ³)
Test1	4.00	6.71	3.40	162.80	136.78	74.46	872.00	1196.87
Test2	4.48	7.52	3.40	187.37	147.39	78.88	942.00	895.57
Test3	4.10	6.88	3.40	211.94	147.86	94.08	955.00	958.36
Test4	6.68	11.21	3.40	162.80	244.02	144.12	1098.00	540.38
Test5	6.74	11.31	3.40	187.37	243.52	131.65	1153.84	484.65
Test6	6.50	10.91	3.40	211.94	263.30	133.61	1233.00	492.30
Test7	7.62	12.79	3.40	162.80	185.01	116.94	1175.00	444.40
Test8	7.65	12.84	3.40	187.37	221.08	123.89	1277.00	416.36
Test9	8.00	13.43	3.40	211.94	265.67	153.64	1754.00	462.32
Test10	4.90	8.22	3.78	162.80	233.33	97.83	1002.00	916.49
Test11	4.70	7.89	3.78	187.37	204.10	112.12	1028.00	887.98
Test12	3.73	6.26	3.78	211.94	145.14	96.42	955.00	1157.92
Test13	5.62	9.43	3.78	162.80	250.87	118.69	1037.00	721.04
Test14	5.40	9.06	3.78	187.37	239.09	122.75	1072.44	701.76
Test15	5.64	9.47	3.78	211.94	239.67	115.97	1139.96	604.54
Test16	8.10	13.59	3.78	162.80	265.22	134.43	1102.16	368.91
Test17	8.43	14.15	3.78	187.37	265.16	122.41	1466.00	393.63
Test18	8.46	14.20	3.78	211.94	265.69	127.87	1274.98	300.51
Test19	4.70	7.89	4.17	162.80	111.55	65.96	878.00	872.87
Test20	3.70	6.21	4.17	187.37	116.50	56.89	868.00	1209.82
Test21	3.76	6.31	4.17	211.94	103.00	55.52	864.39	1031.39
Test22	5.48	9.20	4.17	162.80	142.12	93.65	955.00	698.38
Test23	5.36	9.00	4.17	187.37	116.57	77.41	942.00	625.64
Test24	5.37	9.01	4.17	211.94	134.25	68.20	957.00	559.83
Test25	7.85	13.17	4.17	162.80	110.97	74.86	957.00	341.05
Test26	8.83	14.82	4.17	187.37	153.93	84.49	1029.02	251.83
Test27	8.42	14.13	4.17	211.94	217.81	125.05	1176.00	279.82

Mix-2-Rep1

Test Number	d _{real} (mm)	W _{eff} (mm)	Rotational Speed (m/s)	Table Speed (cm/min)	MAX Vibration-X (mm/s)	MAX Vibration-Z (mm/s)	MAX Power (W)	SE (MJ/m3)
Test1	4.24	7.12	3.40	162.80	81.66	56.57	870.00	1062.77
Test2	4.27	7.17	3.40	187.37	71.72	59.50	872.00	912.57
Test3	4.20	7.05	3.40	211.94	96.00	80.93	945.60	904.28
Test4	6.50	10.91	3.40	162.80	133.75	93.83	955.00	496.39
Test5	6.25	10.49	3.40	187.37	125.65	94.86	1037.00	506.55
Test6	6.68	11.21	3.40	211.94	127.00	90.77	980.00	370.48
Test7	8.14	13.66	3.40	162.80	193.35	105.46	1162.00	385.13
Test8	7.65	12.84	3.40	187.37	179.12	92.63	1082.00	352.78
Test9	8.39	14.08	3.40	211.94	215.30	121.99	1115.00	267.20
Test10	4.74	7.95	3.78	162.80	133.72	73.44	872.00	852.34
Test11	3.96	6.65	3.78	187.37	141.83	68.75	863.08	1050.19
Test12	4.80	8.06	3.78	211.94	176.93	102.73	1008.00	738.02
Test13	5.63	9.45	3.78	162.80	153.99	89.32	950.00	658.20
Test14	5.53	9.28	3.78	187.37	137.15	82.33	950.00	592.76
Test15	6.23	10.46	3.78	211.94	234.98	106.13	1038.00	451.14
Test16	8.07	13.54	3.78	162.80	222.90	91.53	1112.00	374.98
Test17	8.38	14.06	3.78	187.37	249.14	101.43	1187.00	322.53
Test18	8.25	13.85	3.78	211.94	230.48	97.27	1336.00	331.12
Test19	4.69	7.87	4.17	162.80	95.18	78.28	876.00	874.60
Test20	4.59	7.70	4.17	187.37	96.49	57.23	863.00	781.61
Test21	4.44	7.45	4.17	211.94	113.98	62.39	958.00	819.77
Test22	6.57	11.03	4.17	162.80	124.36	74.89	944.00	480.28
Test23	6.47	10.86	4.17	187.37	130.95	96.36	1038.00	473.14
Test24	6.65	11.16	4.17	211.94	152.23	82.38	1112.00	424.18
Test25	8.54	14.33	4.17	162.80	146.31	86.70	1080.00	325.21
Test26	7.50	12.59	4.17	187.37	146.58	86.79	1033.00	350.42
Test27	8.27	13.88	4.17	211.94	164.80	101.90	1079.00	266.14

Mix-2-Rep2

Test Number	d _{real} (mm)	W _{eff} (mm)	Rotational Speed (m/s)	Table Speed (cm/min)	MAX Vibration-X (mm/s)	MAX Vibration-Z (mm/s)	MAX Power (W)	SE (MJ/m ³)
Test1	4.35	7.30	3.40	162.80	82.51	53.28	793.00	920.34
Test2	4.63	7.77	3.40	187.37	105.11	64.46	877.00	780.63
Test3	4.66	7.82	3.40	211.94	125.74	73.78	872.00	677.39
Test4	6.50	10.91	3.40	162.80	166.47	89.76	882.00	458.45
Test5	6.60	11.08	3.40	187.37	154.61	117.46	1118.00	489.73
Test6	6.69	11.23	3.40	211.94	165.57	98.69	1150.00	433.45
Test7	8.10	13.59	3.40	162.80	207.19	110.92	1153.00	385.93
Test8	8.65	14.52	3.40	187.37	219.15	113.68	1338.00	341.22
Test9	8.75	14.68	3.40	211.94	266.85	135.77	1722.00	379.41
Test10	4.04	6.78	3.78	162.80	136.80	80.68	861.00	1158.49
Test11	4.78	8.02	3.78	187.37	131.31	65.32	873.00	729.06
Test12	4.55	7.64	3.78	211.94	155.20	88.87	950.00	774.09
Test13	6.64	11.14	3.78	162.80	139.27	87.81	937.00	466.72
Test14	6.82	11.45	3.78	187.37	165.76	96.34	1028.00	421.73
Test15	6.80	11.41	3.78	211.94	258.38	120.50	1076.00	392.54
Test16	8.79	14.75	3.78	162.80	229.95	125.71	1225.00	348.18
Test17	8.20	13.76	3.78	187.37	264.70	114.85	1185.00	336.28
Test18	8.00	13.43	3.78	211.94	263.90	144.01	1345.00	354.52
Test19	4.62	7.75	4.17	162.80	125.82	90.63	959.00	986.70
Test20	4.78	8.02	4.17	187.37	131.40	95.80	958.00	800.05
Test21	4.62	7.75	4.17	211.94	133.04	89.15	1009.00	797.44
Test22	6.75	11.33	4.17	162.80	133.01	88.63	1006.00	484.89
Test23	6.27	10.52	4.17	187.37	130.98	104.96	989.00	480.03
Test25	8.92	14.97	4.17	162.80	191.74	119.88	1186.00	327.34
Test26	7.96	13.36	4.17	187.37	134.78	80.90	1034.00	311.39
Test27	8.80	14.77	4.17	211.94	148.43	94.21	1115.00	242.89

Mix-2-Rep3

Test Number	d _{real} (mm)	W _{eff} (mm)	Rotational Speed (m/s)	Table Speed (cm/min)	MAX Vibration-X (mm/s)	MAX Vibration-Z (mm/s)	MAX Power (W)	SE (MJ/m3)
Test1	3.85	6.46	3.4	162.80	97.13	82.22	878.00	1300.84
Test2	4.34	7.28	3.4	187.37	99.64	78.39	940.00	952.26
Test3	4.5	7.55	3.4	211.94	126.52	96.19	933.00	777.23
Test4	6.81	11.43	3.4	162.80	199.29	108.85	1089.00	515.68
Test5	6.25	10.49	3.4	187.37	107.75	78.73	945.00	461.61
Test7	7.54	12.65	3.4	162.80	243.95	116.74	1185.00	457.75
Test8	8.43	14.15	3.4	187.37	207.70	109.26	1181.00	317.10
Test9	8.6	14.43	3.4	211.94	251.76	121.62	1353.00	308.60
Test10	4.92	8.26	3.78	162.80	113.41	84.38	872.03	791.13
Test11	4	6.71	3.78	187.37	140.66	83.62	869.00	1036.35
Test12	4.52	7.59	3.78	211.94	177.72	104.74	957.00	790.18
Test13	6.43	10.79	3.78	162.80	208.09	105.07	971.00	515.76
Test14	6.55	10.99	3.78	187.37	217.00	113.19	1151.80	512.27
Test15	6.93	11.63	3.78	211.94	183.98	113.88	1116.00	392.00
Test16	8.9	14.94	3.78	162.80	203.00	107.37	1108.00	307.19
Test17	8.33	13.98	3.78	187.37	183.95	91.94	1000.00	274.99
Test18	8.56	14.37	3.78	211.94	198.67	95.26	1024.00	235.75
Test19	4.92	8.26	4.17	162.80	89.05	68.66	877.00	795.65
Test20	4.75	7.97	4.17	187.37	113.68	79.02	958.00	810.18
Test21	4.69	7.87	4.17	211.94	138.87	70.80	945.00	724.73
Test22	6.03	10.12	4.17	162.80	90.26	67.26	868.00	524.25
Test23	5.73	9.62	4.17	187.37	105.82	77.10	947.00	550.36
Test24	5.9	9.90	4.17	211.94	123.11	71.05	943.00	456.98
Test25	8.23	13.81	4.17	162.80	118.68	86.60	1020.00	330.71
Test26	8.31	13.95	4.17	187.37	126.48	78.08	1011.00	279.35
Test27	8.74	14.67	4.17	211.94	150.32	85.37	1098.89	242.67

Mix-3-Rep1

Test Number	d _{real} (mm)	W _{eff} (mm)	Rotational Speed (m/s)	Table Speed (cm/min)	MAX Vibration-X (mm/s)	MAX Vibration-Z (mm/s)	MAX Power (W)	SE (MJ/m3)
Test1	4.22	7.08	3.4	162.80	36.92	19.62	719.00	886.66
Test2	4.34	7.28	3.4	187.37	45.70	36.36	794.00	804.35
Test3	4.45	7.47	3.4	211.94	43.47	26.57	762.60	649.63
Test4	6.35	10.66	3.4	162.80	63.69	37.54	798.00	434.62
Test5	6.69	11.23	3.4	187.37	74.59	43.64	796.00	339.36
Test6	6.64	11.14	3.4	211.94	102.65	50.14	868.00	332.11
Test7	7.33	12.30	3.4	162.80	68.97	38.41	790.00	322.90
Test8	8.21	13.78	3.4	187.37	80.43	51.24	863.00	244.30
Test9	8.46	14.20	3.4	211.94	92.16	51.15	875.00	206.23
Test10	3.84	6.44	3.78	162.80	47.15	22.48	715.00	1064.86
Test11	4.18	7.01	3.78	187.37	46.95	24.77	796.97	870.35
Test12	4.63	7.77	3.78	211.94	56.60	35.56	794.00	624.81
Test13	5.71	9.58	3.78	162.80	55.78	36.00	796.00	536.16
Test14	6.57	11.03	3.78	187.37	61.49	38.96	792.00	350.11
Test15	5.36	9.00	3.78	211.94	49.40	25.37	774.00	454.47
Test16	8.47	14.21	3.78	162.80	44.66	21.79	706.26	216.20
Test17	8.24	13.83	3.78	187.37	44.93	20.63	798.00	224.26
Test18	7.52	12.62	3.78	211.94	75.44	42.62	865.42	258.16
Test19	4.83	8.11	4.17	162.80	33.82	22.12	799.00	752.15
Test20	4.1	6.88	4.17	187.37	28.84	12.36	799.00	906.95
Test21	4.1	6.88	4.17	211.94	31.40	14.69	797.00	799.80
Test22	5.85	9.82	4.17	162.80	42.03	32.30	797.00	511.44
Test23	6.4	10.74	4.17	187.37	33.68	19.05	798.00	371.75
Test24	6.84	11.48	4.17	211.94	44.88	26.30	798.00	287.73
Test25	8.31	13.95	4.17	162.80	45.95	22.00	796.58	253.33
Test26	8.68	14.57	4.17	187.37	45.08	23.56	791.00	200.33
Test27	8.76	14.70	4.17	211.94	54.96	30.47	874.00	192.13

Mix-3-Rep2

Test Number	d _{real} (mm)	W _{eff} (mm)	Rotational Speed (m/s)	Table Speed (cm/min)	MAX Vibration-X (mm/s)	MAX Vibration-Z (mm/s)	MAX Power (W)	SE (MJ/m ³)
Test1	3.77	6.33	3.4	162.80	37.50	22.67	719.00	1110.96
Test2	4.29	7.20	3.4	187.37	37.12	22.87	717.00	743.38
Test3	4.42	7.42	3.4	211.94	44.16	28.44	792.67	684.45
Test4	6.86	11.51	3.4	162.80	53.69	32.48	715.65	333.97
Test5	6	10.07	3.4	187.37	47.50	33.61	796.12	421.97
Test6	6.38	10.71	3.4	211.94	59.95	33.96	790.00	327.40
Test7	8.65	14.52	3.4	162.80	68.41	44.82	797.00	233.93
Test8	7.88	13.22	3.4	187.37	96.92	50.53	877.00	269.50
Test9	8.4	14.10	3.4	211.94	109.23	60.63	879.00	210.15
Test10	3.83	6.43	3.78	162.80	27.38	13.53	709.00	1061.45
Test11	4.12	6.91	3.78	187.37	33.63	15.70	692.00	777.89
Test12	4.7	7.89	3.78	211.94	36.11	16.32	796.00	607.87
Test13	6.64	11.14	3.78	162.80	38.21	17.83	710.00	353.65
Test14	6.58	11.04	3.78	187.37	42.63	21.23	737.50	325.03
Test15	6.94	11.65	3.78	211.94	51.14	24.43	797.00	279.15
Test16	7.33	12.30	3.78	162.80	44.09	16.85	717.00	293.06
Test17	8.59	14.42	3.78	187.37	53.69	20.06	798.00	206.36
Test18	8	13.43	3.78	211.94	43.44	19.73	794.48	209.41
Test19	4.88	8.19	4.17	162.80	30.00	13.83	661.00	609.55
Test20	4.75	7.97	4.17	187.37	30.81	14.11	795.00	672.33
Test21	4.35	7.30	4.17	211.94	31.23	14.94	790.00	704.27
Test22	6.24	10.47	4.17	162.80	27.23	15.58	794.00	447.82
Test23	6.76	11.34	4.17	187.37	44.28	21.99	781.05	326.13
Test24	6.27	10.52	4.17	211.94	37.16	21.96	774.00	332.12
Test25	8.9	14.94	4.17	162.80	56.30	34.64	795.00	220.41
Test26	8.62	14.47	4.17	187.37	43.88	27.07	780.00	200.30
Test27	8.38	14.06	4.17	211.94	69.17	37.77	845.00	202.98

Mix-3-Rep3

Test Number	d _{real} (mm)	W _{eff} (mm)	Rotational Speed (m/s)	Table Speed (cm/min)	MAX Vibration-X (mm/s)	MAX Vibration-Z (mm/s)	MAX Power (W)	SE (MJ/m3)
Test1	4.03	6.76	3.4	162.80	22.70	16.38	715.00	966.82
Test2	4.41	7.40	3.4	187.37	35.45	24.46	699.00	685.81
Test3	4.24	7.12	3.4	211.94	32.06	21.77	710.00	666.22
Test4	6.54	10.98	3.4	162.80	39.18	26.64	703.00	360.95
Test5	6.66	11.18	3.4	187.37	75.52	30.37	781.00	335.98
Test6	5.27	8.84	3.4	211.94	32.52	13.98	710.00	431.25
Test7	8.8	14.77	3.4	162.80	42.46	25.10	710.00	201.35
Test8	8.39	14.08	3.4	187.37	52.16	26.26	792.00	214.69
Test9	8.75	14.68	3.4	211.94	57.09	30.69	798.00	175.82
Test10	4.62	7.75	3.78	162.80	32.96	16.09	719.00	739.77
Test11	4	6.71	3.78	187.37	33.93	10.65	692.00	825.26
Test12	4.16	6.98	3.78	211.94	40.45	14.09	796.00	775.92
Test13	6.6	11.08	3.78	162.80	47.47	19.55	799.00	402.82
Test14	5.57	9.35	3.78	187.37	43.20	22.16	796.00	489.56
Test15	5.65	9.48	3.78	211.94	50.15	19.81	792.00	418.52
Test16	8.15	13.68	3.78	162.80	58.17	28.15	796.00	263.18
Test17	8.57	14.38	3.78	187.37	72.09	42.15	787.00	204.46
Test18	7.81	13.11	3.78	211.94	74.44	38.29	794.00	219.59
Test19	3.67	6.16	4.17	162.80	36.83	16.25	797.00	1299.50
Test20	4.11	6.90	4.17	187.37	32.05	16.55	799.00	902.54
Test21	4.71	7.90	4.17	211.94	45.63	16.91	795.00	604.53
Test22	5.9	9.90	4.17	162.80	40.14	15.03	797.00	502.81
Test23	6.71	11.26	4.17	187.37	56.61	28.20	782.20	331.50
Test24	6.82	11.45	4.17	211.94	58.46	34.15	868.00	314.81
Test25	8.28	13.90	4.17	162.80	34.87	12.27	799.00	255.94
Test26	7.85	13.17	4.17	187.37	37.07	15.62	794.92	246.14
Test27	8.73	14.65	4.17	211.94	46.75	24.20	799.00	176.85

Script for Most Important Feature Recognition

```
# ML model comparison and feature importance (Vibration-X and Specific
Energy)
# - Inputs (changeable): Real cutting depth, Rotational speed, Table
speed
# - Targets: Vibration-X (mm/s), Specific Energy (SE)
# - Models: Linear, RidgeCV, LassoCV, RandomForest, GradientBoosting
# - Metrics: pooled OOF R2, MAE, RMSE, MAPE for all reports plus per-
fold mean and std
# Author: Abdolsattar Roudini

# %% Install helper (optional)
try:
    import xgboost # not used by default
except Exception:
    pass

# %% Imports
import re, io, numpy as np, pandas as pd, matplotlib.pyplot as plt
from pathlib import Path
from typing import Dict, List, Tuple

from google.colab import files

from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import RobustScaler, MinMaxScaler,
PolynomialFeatures
from sklearn.pipeline import Pipeline
from sklearn.model_selection import KFold, cross_val_score,
cross_val_predict
from sklearn.metrics import (
    mean_absolute_error,
    mean_squared_error,
    r2_score,
    make_scorer,
)
from sklearn.linear_model import LinearRegression, RidgeCV, LassoCV,
LassoLarsIC
from sklearn.ensemble import RandomForestRegressor,
GradientBoostingRegressor
from sklearn.inspection import permutation_importance

# ----- Small helpers -----
-

def norm(s: str) -> str:
    return re.sub(r'^a-z0-9+', '_', str(s).strip().lower())

def read_all_sheets_from_bytes(b: bytes) -> List[Tuple[str,
pd.DataFrame]]:
    xls = pd.ExcelFile(io.BytesIO(b))
    out = []
    for sh in xls.sheet_names:
        try:
```

```

        df = pd.read_excel(io.BytesIO(b), sheet_name=sh)
        out.append((sh, df))
    except Exception:
        pass
    return out

def _pick_col(row: pd.DataFrame, preferences: List[str],
must_contains: List[str]=None) -> str:
    nm = {norm(c): c for c in row.columns}
    for p in preferences:
        for k, orig in nm.items():
            if p in k:
                return orig
    if must_contains:
        for c in row.columns:
            lc = c.lower()
            if all(m in lc for m in must_contains):
                return c
    return None

def pick_depth(row):
    return _pick_col(row,
['real_cutting_depth_mm_', 'cutting_depth_mm_', 'real_cutting_depth', 'c
utting_depth', 'depth'])

def pick_rot(row):
    return _pick_col(row,
['rotational_speed_m_s_', 'rotational_speed', 'rotation_speed', 'rpm', 'r
ot'])

def pick_table(row):
    return _pick_col(row,
['table_speed_cm_min_', 'traverse_speed', 'feed_rate', 'table_speed', 'fe
ed', 'table'])

def pick_vibration_x(row):
    return _pick_col(row, [
        'vibration_-x_mm_s_', 'vibration__x_mm_s_', 'vibration_x_mm_s_',
        'vibration_-x', 'vibration_x', 'panel_vbr_1', 'vbr_1', 'vibration1'
    ], must_contains=['vibration', 'x']) or _pick_col(row,
['vibration'], must_contains=['vibration'])

def pick_specific_energy(row):
    return _pick_col(row, [
        'specific_energy', 'mechanical_specific_energy', 'mse', 'se_mj',
        'se_mj_m3', 'se_(mj/m3)', 'se'
    ])

def prepare_df(row: pd.DataFrame) -> Tuple[pd.DataFrame,
Dict[str, str]]:
    depth = pick_depth(row); rot = pick_rot(row); table =
pick_table(row)
    vibx = pick_vibration_x(row); secol = pick_specific_energy(row)

    needed = {'depth': depth, 'rot': rot, 'table': table}
    missing = [k for k, v in needed.items() if v is None]
```

```

    if missing:
        raise ValueError(f"Missing key input columns: {missing}\nFound:
{list(raw.columns)}")

    cols = [depth, rot, table]
    ren = {depth:'depth', rot:'rot', table:'table'}
    if vibx is not None:
        cols.append(vibx); ren[vibx] = 'vibration_x'
    if secol is not None:
        cols.append(secol); ren[secol] = 'SE'

    df = raw[cols].rename(columns=ren).apply(pd.to_numeric,
errors='coerce').dropna().copy()
    return df, ren

def build_models(feats_cols: List[str]) -> Dict[str, Pipeline]:
    feat_lin = ColumnTransformer([('scale', RobustScaler(),
feats_cols)], remainder='drop')
    return {
        'Linear': Pipeline([('scaler', feat_lin), ('m',
LinearRegression())]),
        'RidgeCV': Pipeline([('scaler', feat_lin), ('m',
RidgeCV(alphas=np.logspace(-3,3,25)))]),
        'LassoCV': Pipeline([('scaler', feat_lin), ('m', LassoCV(cv=5,
max_iter=10000, random_state=42))]),
        'RandomForest': Pipeline([('id','passthrough'), ('m',
RandomForestRegressor(n_estimators=500, random_state=42, n_jobs=-
1))]),
        'GradientBoosting': Pipeline([('id','passthrough'), ('m',
GradientBoostingRegressor(random_state=42))]),
    }

# ----- Metrics for older sklearn -----
# -----

def rmse(y_true, y_pred):
    return float(np.sqrt(mean_squared_error(y_true, y_pred)))

def mape_safe(y_true, y_pred, eps=1e-8):
    y_true = np.asarray(y_true, dtype=float)
    y_pred = np.asarray(y_pred, dtype=float)
    denom = np.where(np.abs(y_true) < eps, eps, np.abs(y_true))
    return float(np.mean(np.abs((y_true - y_pred) / denom)))

# ----- Evaluation with pooled OOF metrics -----
# -----

def evaluate_models_oof(X: pd.DataFrame, y: pd.Series, feats_cols:
List[str], n_splits: int) -> pd.DataFrame:
    """
    Return pooled OOF R2, MAE, RMSE, MAPE for each model.
    Also include mean and std per fold for transparency.
    """
    kf = KFold(n_splits=n_splits, shuffle=True, random_state=42)

    mae_scorer = make_scorer(mean_absolute_error,
greater_is_better=False)

```

```

        rmse_scorer = make_scorer(lambda yt, yp:
np.sqrt(mean_squared_error(yt, yp)), greater_is_better=False)
        mape_scorer = make_scorer(lambda yt, yp: mape_safe(yt, yp),
greater_is_better=False)

    rows = []
    for mname, pipe in build_models(feats_cols).items():
        # pooled metrics from OOF predictions
        yhat = cross_val_predict(pipe, X, y, cv=kf)
        r2_pooled = r2_score(y, yhat)
        mae_pooled = mean_absolute_error(y, yhat)
        rmse_pooled = rmse(y, yhat)
        mape_pooled = mape_safe(y, yhat)

        # per fold scores, flip sign back to positive
        r2_folds = cross_val_score(pipe, X, y, scoring='r2', cv=kf)
        mae_folds = -cross_val_score(pipe, X, y,
scoring=mae_scorer, cv=kf)
        rmse_folds = -cross_val_score(pipe, X, y, scoring=rmse_scorer,
cv=kf)
        mape_folds = -cross_val_score(pipe, X, y, scoring=mape_scorer,
cv=kf)

        rows.append([
            mname,
            r2_pooled, mae_pooled, rmse_pooled, mape_pooled,
            r2_folds.mean(), r2_folds.std(),
            mae_folds.mean(), mae_folds.std(),
            rmse_folds.mean(), rmse_folds.std(),
            mape_folds.mean(), mape_folds.std()
        ])

    perf = pd.DataFrame(rows, columns=[
        'model',
        'cv_R2', 'cv_MAE', 'cv_RMSE', 'cv_MAPE',
        'fold_R2_mean', 'fold_R2_std',
        'fold_MAE_mean', 'fold_MAE_std',
        'fold_RMSE_mean', 'fold_RMSE_std',
        'fold_MAPE_mean', 'fold_MAPE_std'
    ])

    perf = perf.sort_values(['cv_R2', 'cv_MAE'], ascending=[False,
True]).reset_index(drop=True)
    return perf

def best_model_name(perf: pd.DataFrame) -> str:
    top_r2 = perf['cv_R2'].max()
    cand = perf[perf['cv_R2'] == top_r2]
    return cand.sort_values('cv_MAE', ascending=True).iloc[0]['model']

def fit_pipeline(name: str, X: pd.DataFrame, y: pd.Series, feat_cols:
List[str]) -> Pipeline:
    pipe = build_models(feats_cols)[name]
    pipe.fit(X, y)
    return pipe

def permutation_importance_pct(pipe: Pipeline, X: pd.DataFrame, y:
pd.Series, n_repeats:int=15) -> pd.DataFrame:

```

```

"""
Return permutation importance as percentage contributions that sum
to 100.
Uses absolute importances for normalization to avoid negative
totals.
"""
pi = permutation_importance(pipe, X, y, n_repeats=n_repeats,
random_state=42, scoring='r2')
imp = pi.importances_mean
denom = np.sum(np.abs(imp))
pct = 100.0 * (np.abs(imp) / denom) if denom > 0 else
np.zeros_like(imp)
return (
    pd.DataFrame({'feature': X.columns, 'importance_pct': pct})
    .sort_values('importance_pct', ascending=False)
    .reset_index(drop=True)
)

def add_value_labels(ax, bars, fmt="{:0.3f}", dy_frac=0.02):
    ymax = ax.get_ylim()[1] if ax.get_ylim()[1] > 0 else 1.0
    for b in bars:
        h = b.get_height()
        ax.text(b.get_x()+b.get_width()/2, h + dy_frac*ymax,
fmt.format(h), ha='center', va='bottom', fontsize=9)

def fit_symbolic_sparse_poly(X: pd.DataFrame, y: pd.Series,
degree:int=3) -> str:
    scaler = MinMaxScaler()
    Xn = scaler.fit_transform(X.values)
    poly = PolynomialFeatures(degree=degree, include_bias=True)
    Xp = poly.fit_transform(Xn)
    larsic = LassoLarsIC(criterion="aic")
    larsic.fit(Xp, y.values)
    coefs = larsic.coef_.astype(float)
    intercept = float(larsic.intercept_)
    terms = poly.get_feature_names_out(['depth', 'rot', 'table'])
    nz = np.where(np.abs(coefs) > 1e-10)[0]
    pieces = [f"{intercept:.6g}"]
    for i in nz:
        pieces.append("{:0.6g}*{}".format(coefs[i], terms[i]))
    return "y = " + " ".join(pieces)

# ----- Upload and parse -----
----

print("Upload your 2-3 Excel files, for example Sample1, Sample2,
Sample3.")
uploaded = files.upload()
assert len(uploaded) >= 1, "Please upload at least one Excel file."

datasets = []
for fname, b in uploaded.items():
    sheets = read_all_sheets_from_bytes(b)
    if not sheets:
        print(f"[WARN] No readable sheets in {fname}, skipping.")
        continue
    best_sh, best_df, best_n = None, None, -1

```

```

for sh, df in sheets:
    if df is not None and df.shape[0] > best_n:
        best_sh, best_df, best_n = sh, df, df.shape[0]
label = f"{Path(fname).name}::{best_sh}"
try:
    df, ren = prepare_df(best_df)
    datasets.append((label, df))
    print(f"[OK] {label}: rows={len(df)} | cols used:
depth={ren.get('depth')}, rot={ren.get('rot')},
table={ren.get('table')}, vibX={ren.get('vibration_x', '(missing)')},
SE={ren.get('SE', '(missing)')}")
except Exception as e:
    print(f"[WARN] Skipping {label}: {e}")

assert datasets, "No valid datasets detected."

# ----- Analysis -----

summary_best = [] # [label, target, best_model, cv_R2, cv_MAE,
cv_RMSE, cv_MAPE]
summary_import = [] # [label, target, feature, importance_pct]
symbolic_eqs = [] # [label, target, degree, equation]

for label, df in datasets:
    print("\n" + "="*100)
    print("DATASET:", label)
    print("="*100)
    feat_cols = ['depth', 'rot', 'table']
    n_splits = 5 if len(df) >= 60 else 3

    # ----- Target 1: Vibration-X -----
    if 'vibration_x' in df.columns:
        X = df[feat_cols]; y = df['vibration_x']

        perf = evaluate_models_oof(X, y, feat_cols, n_splits)
        display(perf.style.format({
            'cv_R2': '{:.3f}', 'cv_MAE': '{:.2f}', 'cv_RMSE': '{:.2f}', 'cv
_MAPE': '{:.2%}',
            'fold_R2_mean': '{:.3f}', 'fold_R2_std': '{:.3f}',
            'fold_MAE_mean': '{:.2f}', 'fold_MAE_std': '{:.2f}',
            'fold_RMSE_mean': '{:.2f}', 'fold_RMSE_std': '{:.2f}',
            'fold_MAPE_mean': '{:.2%}', 'fold_MAPE_std': '{:.2%}',
        })).set_caption(f"{label} - Vibration-X: model comparison,
pooled OOF")

        plt.figure(figsize=(6, 3.6))
        bars = plt.bar(perf['model'], perf['cv_R2'])
        plt.ylabel("Pooled CV R2"); plt.title(f"{label} - Vibration-X,
pooled R2")
        plt.xticks(rotation=30, ha='right'); plt.ylim(min(0.0,
perf['cv_R2'].min()-0.05), max(0.01, perf['cv_R2'].max()*1.12))
        add_value_labels(plt.gca(), bars, fmt="{:.3f}")
        plt.tight_layout(); plt.show()

        bm = best_model_name(perf)
        bm_pipe = fit_pipeline(bm, X, y, feat_cols)

```

```

pi_df = permutation_importance_pct(bm_pipe, X, y, n_repeats=15)
display(pi_df.style.format({'importance_pct': '{:.1f}%'}).set_
caption(f"{label} - Vibration-X: permutation importance, percent
contribution (best model: {bm})"))

plt.figure(figsize=(5,3))
bars = plt.bar(pi_df['feature'], pi_df['importance_pct'])
plt.ylabel("Importance, percent of total")
plt.title(f"{label} - Vib-X feature importance, best: {bm}")
plt.xticks(rotation=0, ha='center'); plt.ylim(0, max(1.0,
pi_df['importance_pct'].max()*1.15))
add_value_labels(plt.gca(), bars, fmt="{:.1f}%")
plt.tight_layout(); plt.show()

# Pred vs Actual using pooled OOF predictions for the best
model
kf = KFold(n_splits=n_splits, shuffle=True, random_state=42)
yhat_cv = cross_val_predict(build_models(feat_cols)[bm], X, y,
cv=kf)

r2_pooled = r2_score(y, yhat_cv)
mae_pooled = mean_absolute_error(y, yhat_cv)
rmse_pooled = rmse(y, yhat_cv)
mape_pooled = mape_safe(y, yhat_cv)

plt.figure(figsize=(4.8,4.8))
plt.scatter(y, yhat_cv, s=14)
lo, hi = float(min(y.min(), yhat_cv.min())), float(max(y.max(),
yhat_cv.max()))
plt.plot([lo, hi], [lo, hi], lw=1)
plt.xlabel("Actual Vib-X"); plt.ylabel("Predicted Vib-X, OOF")
plt.title(f"{label} - {bm}: R2={r2_pooled:.3f},
MAE={mae_pooled:.2f}, RMSE={rmse_pooled:.2f}, MAPE={mape_pooled:.2%}")
plt.tight_layout(); plt.show()

summary_best.append([label, "Vibration-X", bm,
float(perf.iloc[0]['cv_R2']), float(perf.iloc[0]['cv_MAE']),
float(perf.iloc[0]['cv_RMSE']), float(perf.iloc[0]['cv_MAPE'])])
for _, r in pi_df.iterrows():
    summary_import.append([label, "Vibration-X", r['feature'],
float(r['importance_pct'])])

# Symbolic regression
try:
    eq = fit_symbolic_sparse_poly(X, y, degree=3)
    symbolic_eqs.append([label, "Vibration-X", 3, eq])
    print("Symbolic equation (Vibration-X, degree 3):")
    print(eq)
except Exception as e:
    print(f"[WARN] Symbolic regression failed for Vib-X: {e}")

# ----- Target 2: Specific Energy -----
if 'SE' in df.columns:
    X = df[feat_cols]; y = df['SE']

perf = evaluate_models_oof(X, y, feat_cols, n_splits)
display(perf.style.format({

```

```

        'cv_R2': '{:.3f}', 'cv_MAE': '{:.2f}', 'cv_RMSE': '{:.2f}', 'cv_
_MAPE': '{:.2%}',
        'fold_R2_mean': '{:.3f}', 'fold_R2_std': '{:.3f}',
        'fold_MAE_mean': '{:.2f}', 'fold_MAE_std': '{:.2f}',
        'fold_RMSE_mean': '{:.2f}', 'fold_RMSE_std': '{:.2f}',
        'fold_MAPE_mean': '{:.2%}', 'fold_MAPE_std': '{:.2%}',
    )).set_caption(f"{label} - SE: model comparison, pooled OOF"))

plt.figure(figsize=(6,3.6))
bars = plt.bar(perf['model'], perf['cv_R2'])
plt.ylabel("Pooled CV R2"); plt.title(f"{label} - SE, pooled
R2")

    plt.xticks(rotation=30, ha='right'); plt.ylim(min(0.0,
perf['cv_R2'].min()-0.05), max(0.01, perf['cv_R2'].max()*1.12))
    add_value_labels(plt.gca(), bars, fmt="{:.3f}")
    plt.tight_layout(); plt.show()

bm = best_model_name(perf)
bm_pipe = fit_pipeline(bm, X, y, feat_cols)

pi_df = permutation_importance_pct(bm_pipe, X, y, n_repeats=15)
display(pi_df.style.format({'importance_pct': '{:.1f}%'}).set_
caption(f"{label} - SE: permutation importance, percent contribution
(best model: {bm})"))

plt.figure(figsize=(5,3))
bars = plt.bar(pi_df['feature'], pi_df['importance_pct'])
plt.ylabel("Importance, percent of total")
plt.title(f"{label} - SE feature importance, best: {bm}")
    plt.xticks(rotation=0, ha='center'); plt.ylim(0, max(1.0,
pi_df['importance_pct'].max()*1.15))
    add_value_labels(plt.gca(), bars, fmt="{:.1f}%")
    plt.tight_layout(); plt.show()

kf = KFold(n_splits=n_splits, shuffle=True, random_state=42)
yhat_cv = cross_val_predict(build_models(feat_cols)[bm], X, y,
cv=kf)

r2_pooled = r2_score(y, yhat_cv)
mae_pooled = mean_absolute_error(y, yhat_cv)
rmse_pooled = rmse(y, yhat_cv)
mape_pooled = mape_safe(y, yhat_cv)

plt.figure(figsize=(4.8,4.8))
plt.scatter(y, yhat_cv, s=14)
lo, hi = float(min(y.min(), yhat_cv.min())), float(max(y.max(),
yhat_cv.max()))
plt.plot([lo, hi], [lo, hi], lw=1)
plt.xlabel("Actual SE"); plt.ylabel("Predicted SE, OOF")
    plt.title(f"{label} - {bm}: R2={r2_pooled:.3f},
MAE={mae_pooled:.2f}, RMSE={rmse_pooled:.2f}, MAPE={mape_pooled:.2%}")
    plt.tight_layout(); plt.show()

        summary_best.append([label, "SE", bm,
float(perf.iloc[0]['cv_R2']), float(perf.iloc[0]['cv_MAE']),
float(perf.iloc[0]['cv_RMSE']), float(perf.iloc[0]['cv_MAPE'])])
        for _, r in pi_df.iterrows():

```

```

summary_import.append([label, "SE", r['feature'],
float(r['importance_pct'])])

try:
    eq = fit_symbolic_sparse_poly(X, y, degree=3)
    symbolic_eqs.append([label, "SE", 3, eq])
    print("Symbolic equation (SE, degree 3):")
    print(eq)
except Exception as e:
    print(f"[WARN] Symbolic regression failed for SE: {e}")

# ----- Summary tables and CSV exports -----
# -----

summary_best_df = pd.DataFrame(summary_best,
columns=['dataset', 'target', 'best_model', 'cv_R2', 'cv_MAE', 'cv_RMSE', '
cv_MAPE'])
summary_imp_df = pd.DataFrame(summary_import,
columns=['dataset', 'target', 'feature', 'importance_pct'])
symbolic_df = pd.DataFrame(symbolic_eqs,
columns=['dataset', 'target', 'degree', 'equation'])

if not summary_best_df.empty:
    display(summary_best_df.sort_values(['target', 'dataset']).style.f
ormat({
        'cv_R2': '{:.3f}', 'cv_MAE': '{:.2f}', 'cv_RMSE': '{:.2f}', 'cv_MAP
E': '{:.2%}'
    })).set_caption("Best model per dataset and target, pooled OOF")
if not summary_imp_df.empty:
    display(summary_imp_df.sort_values(['target', 'dataset', 'feature']
).style.format({'importance_pct': '{:.1f}%'}).set_caption("Permutation
importance of best model, percent contribution"))
if not symbolic_df.empty:
    display(symbolic_df.style.set_caption("Symbolic regression
equations, sparse polynomial on MinMax scaled inputs"))

summary_best_df.to_csv("summary_best_models.csv", index=False)
summary_imp_df.to_csv("summary_feature_importances.csv", index=False)
symbolic_df.to_csv("summary_symbolic_equations.csv", index=False)
print("\nSaved CSVs: summary_best_models.csv,
summary_feature_importances.csv, summary_symbolic_equations.csv")

# ----- Summary plots across samples -----
# -----

def short_name(s: str) -> str:
    base = Path(s.split("::")[0]).stem
    m = re.search(r'(sample\s*\d+)', base.lower())
    return m.group(1).title().replace(" ", "") if m else base

# (A) Best pooled CV R2 per sample for each target
for trg in ['Vibration-X', 'SE']:
    sub = summary_best_df[summary_best_df['target']==trg].copy()
    if sub.empty:
        continue
    sub['sample'] = sub['dataset'].apply(short_name)
    sub = sub.sort_values('sample')

```

```

plt.figure(figsize=(6,3.6))
bars = plt.bar(sub['sample'], sub['cv_R2'])
for i, b in enumerate(bars):
    plt.text(b.get_x()+b.get_width()/2, b.get_height()*1.02,
sub['best_model'].iloc[i], ha='center', va='bottom', fontsize=9)
plt.ylabel("Best pooled CV R2"); plt.title(f"Best model performance
by sample - {trg}")
plt.ylim(min(0.0, sub['cv_R2'].min()-0.05), max(0.01,
sub['cv_R2'].max()*1.12))
plt.tight_layout(); plt.show()

# (B) Feature importance grouped bars per target, shown as percentages
for trg in ['Vibration-X', 'SE']:
    sub = summary_imp_df[summary_imp_df['target']==trg].copy()
    if sub.empty:
        continue
    sub['sample'] = sub['dataset'].apply(short_name)
    features = ['depth', 'rot', 'table']
    samples = sorted(sub['sample'].unique())
    M = np.zeros((len(samples), len(features)))
    for i, s in enumerate(samples):
        ss = sub[sub['sample']==s]
        for j, f in enumerate(features):
            vals = ss[ss['feature']==f]['importance_pct']
            M[i, j] = float(vals.mean()) if len(vals) else 0.0
    x = np.arange(len(samples)); w = 0.25
    plt.figure(figsize=(max(6, 1.6*len(samples)), 4))
    bars = []
    for j, f in enumerate(features):
        bj = plt.bar(x + (j-1)*w, M[:, j], width=w, label=f)
        bars.append(bj)
    plt.xticks(x, samples); plt.ylabel("Importance, percent of total")
    plt.title(f"Feature importance by sample - {trg} (best model per
sample)")
    plt.legend()
    ymax = plt.gca().get_ylim()[1] if plt.gca().get_ylim()[1] > 0 else
1.0
    for bj in bars:
        for b in bj:
            h = b.get_height()
            plt.text(b.get_x()+b.get_width()/2, h + 0.02*ymax,
f"{h:.1f}%", ha='center', va='bottom', fontsize=8)
    plt.tight_layout(); plt.show()

```

Optimization Script

```
# Joint PSO with cross-sample constraints, thresholds, and online
control
# - Optimizes [depth, rot, table] for all 3 samples jointly (surrogate:
RF or GB)
# - Enforces:
#     * depth1 < depth2 < depth3 (strict with small margin)
#     * rot1 >= rot2 and rot1 >= rot3
#     * table1 can be lower than others (soft preference, not hard
constraint)
# - Penalizes high vibration by reducing depth (per-sample cap)
# - Learns Vib-X thresholds t1, t2 for 3-class categorization
# - Provides recommend_settings_from_vibx(v) for online use
# Author: Abdolsattar Roudini

# %% Install
!pip -q install openpyxl pyswarms

# %% Imports
import io, re, numpy as np, pandas as pd, matplotlib.pyplot as plt
from pathlib import Path
from typing import List, Tuple
from IPython.display import display
from google.colab import files
from sklearn.model_selection import KFold, cross_val_score
from sklearn.metrics import make_scorer, mean_absolute_error
from sklearn.pipeline import Pipeline
from sklearn.ensemble import RandomForestRegressor,
GradientBoostingRegressor
import pyswarms as psw # renamed to avoid collisions with any variable
named `ps`

np.random.seed(42)

# ===== Tunable knobs =====
SWARM_SIZE    = 100
PSO_ITERS     = 180
CV_BIG        = 5
CV_SMALL      = 3

# Default objective weights
W_SE          = 0.5
W_VIB         = 0.5

# High Vib-X penalty strength (per sample)
PENALTY_ALPHA = 1.0

# Auto Vib-X cap per sample = this percentile of observed vib-x
VIB_CAP_PERCENTILE = 66

# Cross-sample constraint margins
MARGIN_DEPTH_FRAC = 0.02
MARGIN_ROT_FRAC   = 0.00

# Penalty strengths for cross-sample constraints
GAMMA_DEPTH_ORDER = 6.0
```

```

GAMMA_ROT_ORDER    = 3.0
GAMMA_TABLE_PREF   = 1.0
GAMMA_SE_ORDER     = 1.5

# ===== Helpers: parsing =====
def norm_colname(s: str) -> str:
    return re.sub(r'^a-z0-9+', '_', str(s).strip().lower())

def read_all_sheets_from_bytes(b: bytes) -> List[Tuple[str,
pd.DataFrame]]:
    xls = pd.ExcelFile(io.BytesIO(b))
    out = []
    for sh in xls.sheet_names:
        try:
            df = pd.read_excel(io.BytesIO(b), sheet_name=sh)
            out.append((sh, df))
        except Exception:
            pass
    return out

def _pick_col(raw: pd.DataFrame, preferences: List[str],
must_contains: List[str]=None) -> str:
    mapping = {norm_colname(c): c for c in raw.columns}
    for p in preferences:
        for k, orig in mapping.items():
            if p in k:
                return orig
    if must_contains:
        for c in raw.columns:
            lc = c.lower()
            if all(m in lc for m in must_contains):
                return c
    return None

def pick_depth(raw): return _pick_col(raw, ['real_cutting_depth_mm_',
'cutting_depth_mm_', 'real_cutting_depth', 'cutting_depth', 'depth'])
def pick_rot(raw): return _pick_col(raw, ['rotational_speed_m_s_',
'rotational_speed', 'rotation_speed', 'rpm', 'rot'])
def pick_table(raw): return _pick_col(raw, ['table_speed_cm_min_',
'traverse_speed', 'feed_rate', 'table_speed', 'feed', 'table'])
def pick_vibx(raw): return _pick_col(raw, [
'vibration_x_mm_s_', 'vibration_x_mm_s_', 'vibration_x_mm_s_',
'vibration_x', 'vibration_x'
], must_contains=['vibration', 'x']) or _pick_col(raw, ['vibration'],
must_contains=['vibration'])
def pick_se(raw): return _pick_col(raw,
['specific_energy', 'mechanical_specific_energy', 'mse', 'se_mj', 'se_mj_
m3', 'se_(mj/m3)', 'se'])

def prepare_df(raw: pd.DataFrame):
    depth = pick_depth(raw); rot = pick_rot(raw); table =
pick_table(raw)
    vibx = pick_vibx(raw); secol = pick_se(raw)
    need = {'depth': depth, 'rot': rot, 'table': table}
    miss = [k for k,v in need.items() if v is None]
    if miss:

```

```

        raise ValueError(f"Missing key inputs: {miss}\nFound columns:
{list(raw.columns)}")
    cols = [depth, rot, table]; ren = {depth:'depth', rot:'rot',
table:'table'}
    if vibx is not None: cols.append(vibx); ren[vibx] = 'vibration_x'
    if secol is not None: cols.append(secol); ren[secol] = 'SE'
    df = raw[cols].rename(columns=ren).apply(pd.to_numeric,
errors='coerce').dropna().copy()
    return df

# ===== Upload and parse =====
print("Upload your 3 Excel files (Sample1, Sample2, Sample3).")
uploaded = files.upload()
assert len(uploaded) >= 3, "Please upload at least three Excel files."

datasets = [] # list of (label, df)
for fname, b in uploaded.items():
    sheets = read_all_sheets_from_bytes(b)
    if not sheets:
        print(f"[WARN] No readable sheets in {fname}, skipping.")
        continue
    best_sh, best_df, best_n = None, None, -1
    for sh, df in sheets:
        if df is not None and df.shape[0] > best_n:
            best_sh, best_df, best_n = sh, df, df.shape[0]
    label = f"{Path(fname).name}:{best_sh}"
    try:
        df = prepare_df(best_df)
        datasets.append((label, df))
        print(f"[OK] {label}: rows={len(df)} | columns used:
{list(df.columns)}")
    except Exception as e:
        print(f"[WARN] Skipping {label}: {e}")

assert len(datasets) == 3, f"Expected 3 valid datasets, got
{len(datasets)}"

# ===== Surrogate models per sample =====
def best_regressor(X: pd.DataFrame, y: pd.Series, cv_splits:int=5):
    from sklearn.pipeline import Pipeline
    models = {
        'RandomForest': Pipeline([('m',
RandomForestRegressor(n_estimators=450, random_state=42, n_jobs=-
1))]),
        'GradientBoosting': Pipeline([('m',
GradientBoostingRegressor(random_state=42))]),
    }
    kf = KFold(n_splits=cv_splits, shuffle=True, random_state=42)
    scores = []
    for name, pipe in models.items():
        r2 = cross_val_score(pipe, X, y, scoring='r2', cv=kf).mean()
        mae = -cross_val_score(pipe, X, y,
scoring=make_scorer(mean_absolute_error, greater_is_better=False),
cv=kf).mean()
        scores.append((name, r2, mae))
    scores.sort(key=lambda t: (-t[1], t[2]))

```

```

    best_name = scores[0][0]
    best_pipe = models[best_name].fit(X, y)
    return best_name, best_pipe, pd.DataFrame(scores,
columns=['model', 'cv_R2', 'cv_MAE'])

per_sample = {} # keyed by label
for label, df in datasets:
    feats = ['depth', 'rot', 'table']
    depth_lo, depth_hi = df['depth'].min(), df['depth'].max()
    rot_levels = sorted(df['rot'].unique().tolist())
    table_levels = sorted(df['table'].unique().tolist())
    cvspl = CV_BIG if len(df) >= 60 else CV_SMALL
    results = {}
    # Require both targets
    if 'vibration_x' in df and len(df) >= 12:
        name_v, mod_v, tab_v = best_regressor(df[feats],
df['vibration_x'], cvspl)
        results.update(dict(v_model_name=name_v, v_model=mod_v,
v_scores=tab_v))
    if 'SE' in df and len(df) >= 12:
        name_s, mod_s, tab_s = best_regressor(df[feats], df['SE'],
cvspl)
        results.update(dict(s_model_name=name_s, s_model=mod_s,
s_scores=tab_s))

    assert 'v_model' in results and 's_model' in results, f"{label}
needs both Vib-X and SE columns with enough rows."

    v_min, v_max = (df['vibration_x'].min(), df['vibration_x'].max())
    s_min, s_max = (df['SE'].min(), df['SE'].max())
    vib_cap = np.percentile(df['vibration_x'].values,
VIB_CAP_PERCENTILE)
    per_sample[label] = dict(
        df=df, feats=feats,
        depth_bounds=(float(depth_lo), float(depth_hi)),
        rot_levels=rot_levels, table_levels=table_levels,
        v_min=float(v_min), v_max=float(v_max),
        s_min=float(s_min), s_max=float(s_max),
        vib_cap=float(vib_cap),
        **results
    )

# ===== Map labels to classes by SE median (Class1 highest to Class3
lowest) =====
stats = [(lab, d['df']['SE'].median()) for lab, d in
per_sample.items()]
stats.sort(key=lambda t: -t[1])
class_ordered_labels = [lab for lab, _ in stats]
class_map = {lab: (i+1) for i, lab in enumerate(class_ordered_labels)}
print("\nClass mapping by SE median (desc):")
for lab, med in stats:
    print(f" Class {class_map[lab]} <= {lab} | median(SE)={med:.3f}")
S1, S2, S3 = class_ordered_labels

# ===== Joint PSO (9D) with constraints
=====
def round_to_nearest(vals, allowed):

```

```

allowed = np.asarray(allowed, dtype=float)
idx = np.abs(vals[:,None] - allowed[None,:]).argmin(axis=1)
return allowed[idx]

def joint_objective_factory(W_se=W_SE, W_vib=W_VIB):
    # Precompute ranges and margins
    d1 = per_sample[S1]['depth_bounds']; d2 =
per_sample[S2]['depth_bounds']; d3 = per_sample[S3]['depth_bounds']
    rot1_levels = np.array(per_sample[S1]['rot_levels'], dtype=float)
    rot2_levels = np.array(per_sample[S2]['rot_levels'], dtype=float)
    rot3_levels = np.array(per_sample[S3]['rot_levels'], dtype=float)
    tab1_levels = np.array(per_sample[S1]['table_levels'],
dtype=float)
    tab2_levels = np.array(per_sample[S2]['table_levels'],
dtype=float)
    tab3_levels = np.array(per_sample[S3]['table_levels'],
dtype=float)

    def norm01(x, lo, hi): return (x - lo) / (hi - lo + 1e-9)

    depth_ranges = (d1[1]-d1[0], d2[1]-d2[0], d3[1]-d3[0])
    rot1_lo, rot1_hi = min(rot1_levels), max(rot1_levels)
    rot2_lo, rot2_hi = min(rot2_levels), max(rot2_levels)
    rot3_lo, rot3_hi = min(rot3_levels), max(rot3_levels)
    m1 = MARGIN_DEPTH_FRAC * depth_ranges[0]
    m2 = MARGIN_DEPTH_FRAC * depth_ranges[1]

    vcap1 = per_sample[S1]['vib_cap']; vcap2 =
per_sample[S2]['vib_cap']; vcap3 = per_sample[S3]['vib_cap']
    vmin1,vmax1 = per_sample[S1]['v_min'], per_sample[S1]['v_max']
    vmin2,vmax2 = per_sample[S2]['v_min'], per_sample[S2]['v_max']
    vmin3,vmax3 = per_sample[S3]['v_min'], per_sample[S3]['v_max']
    smin1,smax1 = per_sample[S1]['s_min'], per_sample[S1]['s_max']
    smin2,smax2 = per_sample[S2]['s_min'], per_sample[S2]['s_max']
    smin3,smax3 = per_sample[S3]['s_min'], per_sample[S3]['s_max']

    vm1, sm1 = per_sample[S1]['v_model'], per_sample[S1]['s_model']
    vm2, sm2 = per_sample[S2]['v_model'], per_sample[S2]['s_model']
    vm3, sm3 = per_sample[S3]['v_model'], per_sample[S3]['s_model']

    def objective(X):
        X = np.asarray(X, dtype=float)

        # depths with bounds
        d1v = np.clip(X[:,0], d1[0], d1[1])
        d2v = np.clip(X[:,3], d2[0], d2[1])
        d3v = np.clip(X[:,6], d3[0], d3[1])

        # discrete rounding for rot and table
        r1v = round_to_nearest(np.clip(X[:,1], rot1_lo, rot1_hi),
rot1_levels)
        r2v = round_to_nearest(np.clip(X[:,4], rot2_lo, rot2_hi),
rot2_levels)
        r3v = round_to_nearest(np.clip(X[:,7], rot3_lo, rot3_hi),
rot3_levels)
        t1v = round_to_nearest(X[:,2], tab1_levels)
        t2v = round_to_nearest(X[:,5], tab2_levels)

```

```

t3v = round_to_nearest(X[:,8], tab3_levels)

# predictions per sample
f1 = pd.DataFrame({'depth':d1v,'rot':r1v,'table':t1v})
f2 = pd.DataFrame({'depth':d2v,'rot':r2v,'table':t2v})
f3 = pd.DataFrame({'depth':d3v,'rot':r3v,'table':t3v})
vib1 = vm1.predict(f1); se1 = sm1.predict(f1)
vib2 = vm2.predict(f2); se2 = sm2.predict(f2)
vib3 = vm3.predict(f3); se3 = sm3.predict(f3)

# base scalarized objective (sum across samples)
vib1n = norm01(vib1, vmin1, vmax1); se1n = norm01(se1, smin1,
smax1)
vib2n = norm01(vib2, vmin2, vmax2); se2n = norm01(se2, smin2,
smax2)
vib3n = norm01(vib3, vmin3, vmax3); se3n = norm01(se3, smin3,
smax3)
base = (W_se*se1n + W_vib*vib1n) + (W_se*se2n + W_vib*vib2n) +
(W_se*se3n + W_vib*vib3n)

# high vibration penalty grows with normalized depth
d1n = norm01(d1v, d1[0], d1[1]); d2n = norm01(d2v, d2[0],
d2[1]); d3n = norm01(d3v, d3[0], d3[1])
pen_v = PENALTY_ALPHA * (
    np.maximum(0.0, vib1 - vcap1)*(0.25+0.75*d1n)
    + np.maximum(0.0, vib2 - vcap2)*(0.25+0.75*d2n)
    + np.maximum(0.0, vib3 - vcap3)*(0.25+0.75*d3n)
)

# cross-sample constraints
depth_norm = ( (d1[1]-d1[0]) + (d2[1]-d2[0]) + (d3[1]-d3[0]) )
/ 3.0 + 1e-9
pen_depth = GAMMA_DEPTH_ORDER * (
    np.maximum(0.0, (d1v + m1) - d2v) + np.maximum(0.0, (d2v +
m2) - d3v)
) / depth_norm

rot_norm = ( (rot1_hi-rot1_lo) + (rot2_hi-rot2_lo) + (rot3_hi-
rot3_lo) ) / 3.0 + 1e-9
pen_rot = GAMMA_ROT_ORDER * (
    np.maximum(0.0, r2v - r1v) + np.maximum(0.0, r3v - r1v)
) / rot_norm

min_t23 = np.minimum(t2v, t3v)
t_span = max(np.max(tab1_levels) - np.min(tab1_levels), 1e-9)
pen_table = GAMMA_TABLE_PREF * np.maximum(0.0, t1v - min_t23)
/ t_span

eps_se = 0.01 * ((smax1-smin1 + smax2-smin2 + smax3-smin3) /
3.0)
pen_se = GAMMA_SE_ORDER * (
    np.maximum(0.0, (se2 - se1) + eps_se) + np.maximum(0.0,
(se3 - se2) + eps_se)
) / (eps_se + 1e-9)

return base + pen_v + pen_depth + pen_rot + pen_table + pen_se

```

```

    return objective

# Bounds for 9-D vector
d1 = per_sample[S1]['depth_bounds']; d2 =
per_sample[S2]['depth_bounds']; d3 = per_sample[S3]['depth_bounds']
r1_lo, r1_hi = min(per_sample[S1]['rot_levels']),
max(per_sample[S1]['rot_levels'])
r2_lo, r2_hi = min(per_sample[S2]['rot_levels']),
max(per_sample[S2]['rot_levels'])
r3_lo, r3_hi = min(per_sample[S3]['rot_levels']),
max(per_sample[S3]['rot_levels'])
t1_lo, t1_hi = min(per_sample[S1]['table_levels']),
max(per_sample[S1]['table_levels'])
t2_lo, t2_hi = min(per_sample[S2]['table_levels']),
max(per_sample[S2]['table_levels'])
t3_lo, t3_hi = min(per_sample[S3]['table_levels']),
max(per_sample[S3]['table_levels'])

lower_bounds = np.array([d1[0], r1_lo, t1_lo, d2[0], r2_lo, t2_lo,
d3[0], r3_lo, t3_lo])
upper_bounds = np.array([d1[1], r1_hi, t1_hi, d2[1], r2_hi, t2_hi,
d3[1], r3_hi, t3_hi])

# Build base objective and optimize once using default weights
objective = joint_objective_factory(W_SE, W_VIB)
optimizer = psw.single.GlobalBestPSO(
    n_particles=SWARM_SIZE, dimensions=9,
    options={'c1':1.6, 'c2':1.6, 'w':0.7},
    bounds=(lower_bounds, upper_bounds)
)
cost, pos = optimizer.optimize(objective, iters=PSO_ITERS,
verbose=False)

# Discrete rounding utilities
def nearest(val, levels):
    arr = np.array(levels, dtype=float)
    return float(arr[np.argmin(np.abs(arr - val))])

# Assemble final settings per class for this base run
opt = {}
opt[S1] = dict(depth=float(np.clip(pos[0], d1[0], d1[1])),
rot=nearest(pos[1], per_sample[S1]['rot_levels']),
table=nearest(pos[2], per_sample[S1]['table_levels']))
opt[S2] = dict(depth=float(np.clip(pos[3], d2[0], d2[1])),
rot=nearest(pos[4], per_sample[S2]['rot_levels']),
table=nearest(pos[5], per_sample[S2]['table_levels']))
opt[S3] = dict(depth=float(np.clip(pos[6], d3[0], d3[1])),
rot=nearest(pos[7], per_sample[S3]['rot_levels']),
table=nearest(pos[8], per_sample[S3]['table_levels']))

# Prediction helper
def predict_for(label, setting):
    f = pd.DataFrame({k:[v] for k,v in setting.items()})
    vm, sm = per_sample[label]['v_model'], per_sample[label]['s_model']
    return float(vm.predict(f)[0]), float(sm.predict(f)[0])

# Report per-class predictions for the base run

```

```

rows = []
for lab in [S1, S2, S3]:
    vib, se = predict_for(lab, opt[lab])
    rows.append([lab, class_map[lab], opt[lab]['depth'],
opt[lab]['rot'], opt[lab]['table'], vib, se])
res_df = pd.DataFrame(rows,
columns=['sample', 'class', 'depth', 'rot', 'table', 'pred_VibX', 'pred_SE'
]).sort_values('class')
display(res_df)
res_df.to_csv("joint_pso_optimized_settings.csv", index=False)
print("Saved: joint_pso_optimized_settings.csv")
print(f"Joint cost: {cost:.6f}")

# Sanity print of constraints
print("\nConstraint checks:")
print(f"Depth order: {S1}: {opt[S1]['depth']:.4f} < {S2}:
{opt[S2]['depth']:.4f} < {S3}: {opt[S3]['depth']:.4f}")
print(f"Rot order: {S1}: {opt[S1]['rot']:.4f} >= {S2}:
{opt[S2]['rot']:.4f}, {S1}: {opt[S1]['rot']:.4f} >= {S3}:
{opt[S3]['rot']:.4f}")
print(f"Table pref: {S1}: {opt[S1]['table']:.4f} vs min({S2},{S3}) =
{min(opt[S2]['table'], opt[S3]['table']):.4f}")

# ===== Tradeoff sweep summary table (per sample
rows) =====
def run_tradeoff_all_samples(W_vib, W_se, swarm_size=SWARM_SIZE,
iters=PSO_ITERS):
    obj = joint_objective_factory(W_se=W_se, W_vib=W_vib)
    optzr = psw.single.GlobalBestPSO(
        n_particles=swarm_size,
        dimensions=9,
        options={'c1':1.6, 'c2':1.6, 'w':0.7},
        bounds=(lower_bounds, upper_bounds)
    )
    cost, pos = optzr.optimize(obj, iters=iters, verbose=False)

    # build best settings per class
    best = {
        S1: dict(depth=float(np.clip(pos[0], d1[0], d1[1])),
rot=nearest(pos[1], per_sample[S1]['rot_levels']),
table=nearest(pos[2],
per_sample[S1]['table_levels'])),
        S2: dict(depth=float(np.clip(pos[3], d2[0], d2[1])),
rot=nearest(pos[4], per_sample[S2]['rot_levels']),
table=nearest(pos[5],
per_sample[S2]['table_levels'])),
        S3: dict(depth=float(np.clip(pos[6], d3[0], d3[1])),
rot=nearest(pos[7], per_sample[S3]['rot_levels']),
table=nearest(pos[8],
per_sample[S3]['table_levels'])),
    }

    # predictions per sample
    preds = {
        S1: predict_for(S1, best[S1]),
        S2: predict_for(S2, best[S2]),
        S3: predict_for(S3, best[S3]),
    }

```

```

    }

    # total objective value at the optimized point
    obj_val = float(obj(np.array([pos]))[0])

    return best, preds, obj_val

tradeoffs = [
    ("Balanced",          0.50, 0.50),
    ("Vibration-priority", 0.80, 0.20),
    ("Energy-priority",   0.20, 0.80),
]

rows = []
details = {}

for name, wv, ws in tradeoffs:
    best, preds, objv = run_tradeoff_all_samples(wv, ws)
    details[name] = {'settings': best, 'preds': preds,
'objective_total': objv}

    for lab in [S1, S2, S3]:
        dpth = float(best[lab]['depth'])
        rtt = float(best[lab]['rot'])
        tbl = float(best[lab]['table'])
        pv, pse = preds[lab] # pred_VibX, pred_SE
        rows.append([lab, name, wv, ws, dpth, rtt, tbl, float(pv),
float(pse), float(objv)])

tradeoff_df = pd.DataFrame(rows, columns=[
    'sample', 'tradeoff', 'w_vib', 'w_se', 'depth', 'rot', 'table', 'pred_Vi
bX', 'pred_SE', 'objective'
])

display(tradeoff_df)
tradeoff_df.to_csv("tradeoff_summary_per_sample.csv", index=False)
print("Saved: tradeoff_summary_per_sample.csv")

# ===== Learn Vib-X thresholds (t1, t2)
=====
comb = []
for lab in [S1, S2, S3]:
    d = per_sample[lab]['df']
    for v in d['vibration_x'].values:
        comb.append((v, class_map[lab]))
comb = pd.DataFrame(comb, columns=['vibx', 'cls']).dropna()

def miscls_for_thresholds(t1, t2, df):
    pred = np.where(df['vibx'] <= t1, 3, np.where(df['vibx'] <= t2, 2,
1))
    return (pred != df['cls']).mean()

if len(comb) >= 30:
    vs = np.sort(comb['vibx'].unique())
    cand = np.percentile(vs, np.linspace(5, 95, 15))
    best = (None, None, 1.0)
    for t1 in cand:

```

```

        for t2 in cand:
            if t2 <= t1:
                continue
            err = miscls_for_thresholds(t1, t2, comb)
            if err < best[2]:
                best = (t1, t2, err)
        t1b, t2b, _ = best
        local1 = np.linspace(max(vs.min(), t1b*0.9), min(vs.max(),
t1b*1.1), 21)
        local2 = np.linspace(max(vs.min(), t2b*0.9), min(vs.max(),
t2b*1.1), 21)
        best2 = best
        for t1 in local1:
            for t2 in local2:
                if t2 <= t1:
                    continue
                err = miscls_for_thresholds(t1, t2, comb)
                if err < best2[2]:
                    best2 = (t1, t2, err)
        t1_opt, t2_opt, err_opt = best2
    else:
        t1_opt = np.percentile(comb['vibx'], 33)
        t2_opt = np.percentile(comb['vibx'], 66)
        err_opt = miscls_for_thresholds(t1_opt, t2_opt, comb)

print(f"\nVibration-X thresholds: t1={t1_opt:.3f} (3 vs 2),
t2={t2_opt:.3f} (2 vs 1) | miscls={err_opt:.3f}")
pd.DataFrame({'threshold': ['t1_3v2', 't2_2v1'], 'value': [t1_opt, t2_opt]
}).to_csv("vibx_thresholds.csv", index=False)

# optional plot
if not comb.empty:
    plt.figure(figsize=(6, 3.2))
    for c, name in [(1, 'Class 1 (highest SE)'), (2, 'Class 2'), (3,
'Class 3 (lowest SE)')]:
        xs = comb[comb['cls'] == c]['vibx'].values
        if len(xs):
            plt.hist(xs, bins=20, alpha=0.5, label=name)
    plt.axvline(t1_opt, linestyle='--', label=f"t1={t1_opt:.1f}")
    plt.axvline(t2_opt, linestyle='-.', label=f"t2={t2_opt:.1f}")
    plt.xlabel("Vibration-X (mm/s)")
    plt.ylabel("Count")
    plt.title("Vib-X distributions with learned thresholds")
    plt.legend()
    plt.tight_layout()
    plt.show()

# ===== Online control helper
=====
class_to_label = {class_map[lab]: lab for lab in [S1, S2, S3]}

def categorize_by_vibx(vibx_value: float) -> int:
    return 3 if vibx_value <= t1_opt else (2 if vibx_value <= t2_opt
else 1)

def recommend_settings_from_vibx(vibx_value: float):
    cls = categorize_by_vibx(vibx_value)

```

```
lab = class_to_label[cls]
st = opt[lab]
return {
    'detected_class': cls,
    'sample_label': lab,
    'recommended_depth': st['depth'],
    'recommended_rot': st['rot'],
    'recommended_table': st['table'],
    't1': float(t1_opt),
    't2': float(t2_opt)
}

# Example
# rec = recommend_settings_from_vibx(120.0)
# print(rec)
```



RESUME

Abdolsattar Roudini holds a PhD in Mining Engineering from Karadeniz Technical University (Trabzon, Türkiye). His doctoral research focused on instrumented laboratory cutting, investigating vibration behavior and specific energy demand of a rotary cutting head. He has further experience in numerical analysis, laboratory test platform development, and sensor–data integration for experimental and applied rock mechanics.

He completed an MSc in Engineering Geology at Islamic Azad University, Zahedan (ranked second in his cohort), and a BSc in Mining Engineering at Shahid Bahonar University of Kerman. As part of his doctoral training, he contributed as a guest researcher at Montan University of Leoben within the Erasmus Program, where he carried out research on haulage systems under the H2020 Robominers project and undertook coursework in Safe Deep Mining, including rock engineering in deep mines, management of rock-pressure risks, and backfilling. He also completed a micro-certificate in Economic Leadership in Mining (University of British Columbia) and a certificate program in Augmented and Virtual Reality (Future Skills Academy).

Mr. Roudini has 18 years of combined academic and industry experience. From 2009 to 2018, he served as a Rock Mechanics Laboratory Instructor at the University of Sistan and Baluchestan, providing laboratory instruction, supporting field training, and assisting with the supervision of graduate students. His prior professional experience includes service as technical manager for several mines and participation in mine plan evaluation within the Sistan and Baluchestan Mining Engineering Organization. His involvement in research projects includes development of a laboratory-scale cutting head for performance evaluation, and his recent contribution to a TÜBİTAK 1001 project on AI-supported autonomous excavation optimization for boom-type roadheaders.

He is a member of Working Groups 1–5 in COST Action CA23118, Futures-Oriented Governance of Outer Space, where he contributes to themes related to governance, technology, sustainability, and space-resource futures. His research interests include mechanical rock excavation, data-driven modelling, digitalisation and automation in mining, AI and machine learning in rock mechanics, AR/VR applications in mining engineering education, and emerging directions in space-resource extraction.