Modelling and optimization of the grinding process

Sohan Anisetty
180030040

Abstract

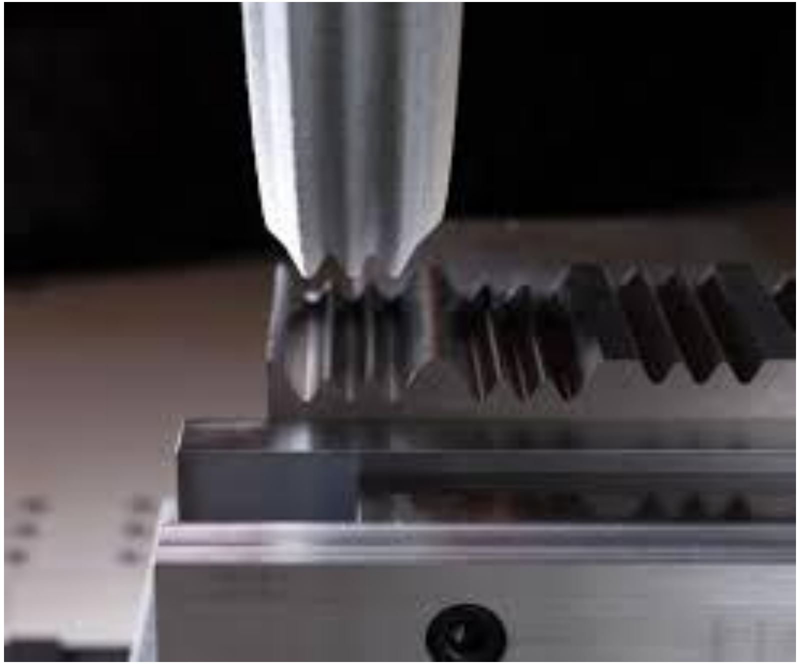
The objective of this study is to show how neural networks and decision trees can be used to model and optimize grinding processes, using creep feed grinding of alumina with diamond wheels as an example.

Importance of machine learning in modelling manufacturing processes

Process modeling and optimization are very important issues in manufacturing engineering. Machining processes are usually too complicated to develop proper analytical models. The operations of machining process thus still rely heavily on the skill of human operators.

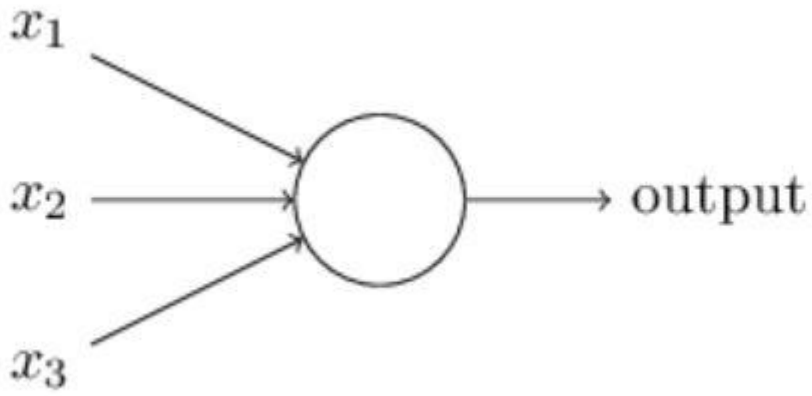
The grinding Process

Grinding is a material removal and surface generation process used to shape and finish components made of metals and other materials.
 creep-feed grinding has a high material removal rate and is used for machining as opposed to finishing.



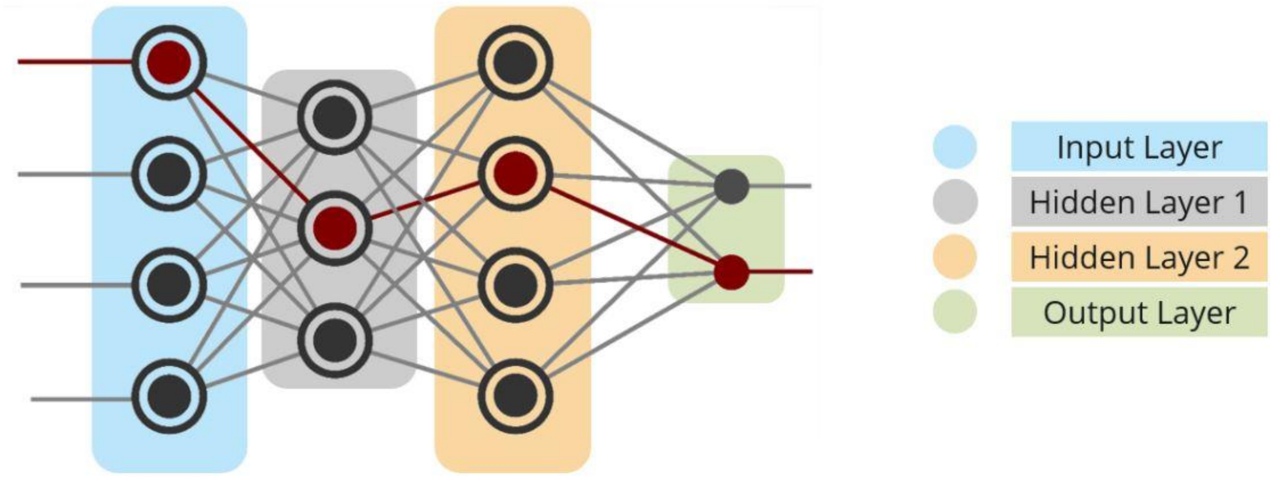
Neural Networks

Neural networks mimic the way our brains solve the problem. The core component of a neural network is a perceptron. A perceptron mimics the neurons in our brain.



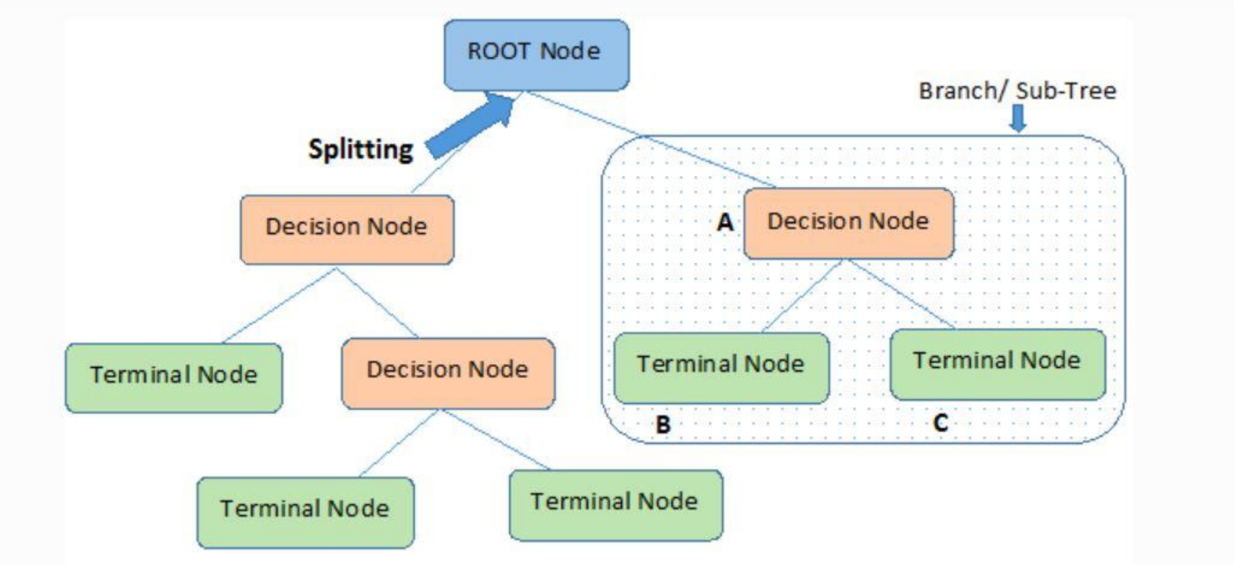
Neural Networks algorithm

Neural Networks are really just a composition of Perceptrons, connected in different ways and operating on different activation functions. Backpropagation and gradient descent are the key concepts used.



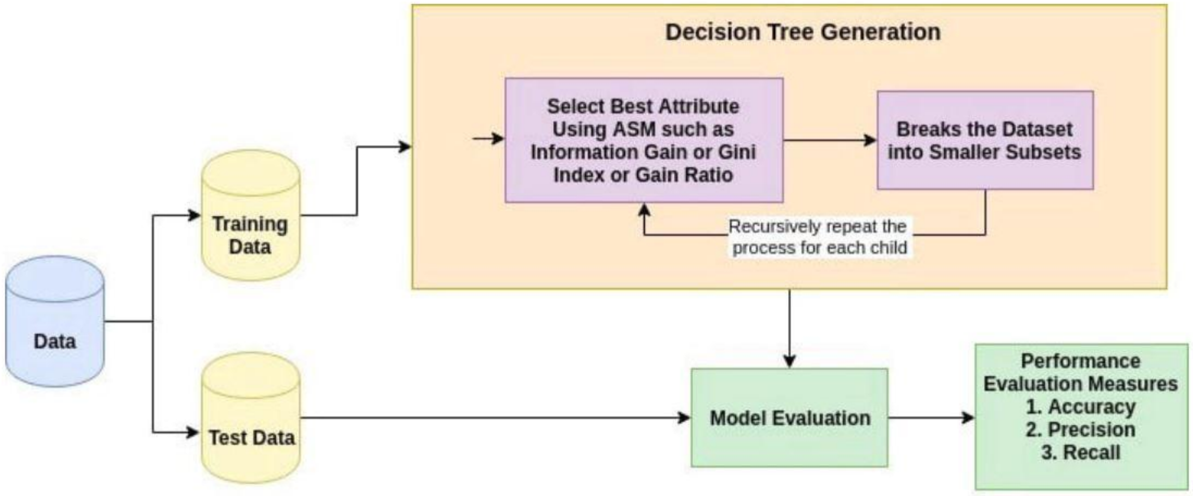
Decision trees

A decision tree is one of the supervised machine learning algorithms. A decision tree follows a set of if-else conditions to visualize the data and classify it according to the conditions. Decision trees work well with categorical Data.



Decision trees algorithm

1. The root node feature is selected based on the results from the Attribute Selection Measure(ASM).
2. The ASM is repeated until a leaf node, or a terminal node cannot be split into sub-nodes.



Experimental setup

Creep feed grinding experiments were performed in the down grinding mode to grind Aluminium specimens with resinoid and vitrified wheels. The diameter and thickness of the wheel were 175 mm and 6.25 mm, respectively. Every wheel has an abrasive layer of 3.125 mm thickness. Wheel speed was kept constant at 22 m/s throughout the experiments

Data

The five input variables used were bond type (B), mesh size (m), concentration (c), work speed (Vw), and depth of cut (d); these variables were varied at two levels. Surface finish (Ra), normal grinding force pe unit width (F,), and grinding power per unit width (Pw) were the three output variables.

No.	B	m	c	v_w, in/min (mm/s)	d, 10^{-3} in (mm)	R_a, μin (μm)	F_n, lb/in (N/mm)	P_w, Hp/in (kW/m)
1	Resin	80	50	6.8 (2.9)	58 (1.47)	34.8 (0.88)	248 (43.4)	7.13 (209)
2	Resin	80	50	6.8 (2.9)	58 (1.47)	33.76 (0.86)	259 (45.3)	7.64 (224)
3	Resin	180	50	6.8 (2.9)	102 (2.59)	32.64 (0.83)	378 (66.2)	11.21 (329)
4	Resin	180	50	6.8 (2.9)	102 (2.59)	33.2 (0.84)	379 (66.3)	11.72 (344)
5	Resin	80	100	6.8 (2.9)	102 (2.59)	33.84 (0.86)	367 (64.2)	10.7 (314)
6	Resin	80	100	6.8 (2.9)	102 (2.59)	34.8 (0.88)	373 (65.3)	10.7 (314)
7	Resin	180	100	6.8 (2.9)	58 (1.47)	26.52 (0.67)	317 (55.5)	7.64 (224)
8	Resin	180	100	6.8 (2.9)	58 (1.47)	26.88 (0.68)	323 (56.5)	8.15 (239)
9	Resin	80	50	16.2 (6.9)	102 (2.59)	35.76 (0.91)	546 (95.6)	20.38 (599)
10	Resin	80	50	16.2 (6.9)	102 (2.59)	35.44 (0.90)	573 (100.3)	19.87 (584)
11	Resin	180	50	16.2 (6.9)	58 (1.47)	33.36 (0.85)	428 (74.9)	12.73 (374)
12	Resin	180	50	16.2 (6.9)	58 (1.47)	33.44 (0.85)	454 (79.5)	13.24 (389)
13	Resin	80	100	16.2 (6.9)	58 (1.47)	35.12 (0.89)	392 (68.6)	11.21 (329)
14	Resin	80	100	16.2 (6.9)	58 (1.47)	35.2 (0.89)	416 (72.8)	11.72 (344)
15	Resin	180	100	16.2 (6.9)	102 (2.59)	27.76 (0.71)	626 (109.6)	22.41 (658)
16	Resin	180	100	16.2 (6.9)	102 (2.59)	27.72 (0.70)	654 (114.5)	21.9 (643)
17	Vitrified	80	50	6.8 (2.9)	102 (2.59)	34.25 (0.87)	210 (36.8)	5.09 (150)
18	Vitrified	80	50	6.8 (2.9)	102 (2.59)	34.12 (0.87)	214 (37.5)	5.6 (165)
19	Vitrified	180	50	6.8 (2.9)	58 (1.47)	22.38 (0.57)	174 (30.5)	3.68 (108)
20	Vitrified	180	50	6.8 (2.9)	58 (1.47)	22.51 (0.57)	185 (32.4)	3.66 (108)
21	Vitrified	80	100	6.8 (2.9)	58 (1.47)	32.02 (0.81)	156 (27.3)	3.56 (105)
22	Vitrified	80	100	6.8 (2.9)	58 (1.47)	31.23 (0.79)	158 (27.7)	3.57 (105)
23	Vitrified	180	100	6.8 (2.9)	102 (2.59)	33.2 (0.84)	156 (27.3)	4.89 (144)
24	Vitrified	180	100	6.8 (2.9)	102 (2.59)	32.74 (0.83)	162 (28.4)	5.09 (149)
25	Vitrified	80	50	16.2 (6.9)	58 (1.47)	36.22 (0.92)	237 (41.5)	6.62 (194)
26	Vitrified	80	50	16.2 (6.9)	58 (1.47)	35.04 (0.89)	220 (38.5)	6.11 (179)
27	Vitrified	180	50	16.2 (6.9)	102 (2.59)	24.41 (0.62)	459 (80.3)	12.73 (374)
28	Vitrified	180	50	16.2 (6.9)	102 (2.59)	23.75 (0.60)	409 (71.6)	12.23 (359)
29	Vitrified	80	100	16.2 (6.9)	102 (2.59)	34.45 (0.88)	347 (60.7)	9.17 (269)
30	Vitrified	80	100	16.2 (6.9)	102 (2.59)	34.91 (0.89)	378 (66.2)	9.37 (275)
31	Vitrified	180	100	16.2 (6.9)	58 (1.47)	33.53 (0.85)	194 (34.0)	6.11 (179)
32	Vitrified	180	100	16.2 (6.9)	58 (1.47)	33.92 (0.86)	174 (30.5)	5.71 (168)

Research paper results

The authors used a four-layer 5x5x4x3 BP network and a mean square error loss function for modelling the grinding process. For optimization they used back propagation and Boltzmann factor. They used
$C1 = 1.0$, $C2 = 3.0$, $C3 = 6.0$, and $C4 = 6.0$ for the objective function.

Modelling results (1335 epochs)

No.	B	m	c	v_w, in/min (mm/sec)	d, 10^{-3} in (mm)	R_a, μmin (μm)	F_n, lb/in (N/mm)	P_w, Hp/in (kw/m)
1	Resin	80	50	6.8 (2.9)	58 (1.47)	34.35 (0.87)	256 (44.8)	6.98 (205)
2	Resin	180	50	6.8 (2.9)	102 (2.59)	32.92 (0.84)	379 (66.3)	11.53 (339)
3	Resin	80	100	6.8 (2.9)	102 (2.59)	34.34 (0.87)	369 (64.6)	9.89 (291)
4	Resin	180	100	6.8 (2.9)	58 (1.47)	26.51 (0.67)	313 (54.8)	8.83 (259)
5	Resin	80	50	16.2 (6.9)	102 (2.59)	34.32 (0.87)	567 (99.2)	19.50 (573)
6	Resin	180	50	16.2 (6.9)	58 (1.47)	33.28 (0.85)	438 (76.7)	13.43 (394)
7	Resin	80	100	16.2 (6.9)	58 (1.47)	34.36 (0.87)	405 (70.9)	10.98 (322)
8	Resin	180	100	16.2 (6.9)	102 (2.59)	27.13 (0.69)	636 (111.3)	21.86 (642)
9	Vitrified	80	50	6.8 (2.9)	102 (2.59)	34.28 (0.87)	203 (35.5)	5.75 (169)
10	Vitrified	180	50	6.8 (2.9)	58 (1.47)	22.73 (0.58)	170 (29.8)	4.19 (123)
11	Vitrified	80	100	6.8 (2.9)	58 (1.47)	32.08 (0.81)	160 (28.0)	4.12 (121)
12	Vitrified	180	100	6.8 (2.9)	102 (2.59)	33.04 (0.84)	169 (29.6)	4.81 (141)
13	Vitrified	80	50	16.2 (6.9)	58 (1.47)	34.33 (0.87)	226 (39.6)	6.11 (179)
14	Vitrified	180	50	16.2 (6.9)	102 (2.59)	24.20 (0.61)	437 (76.5)	12.41 (365)
15	Vitrified	80	100	16.2 (6.9)	102 (2.59)	34.36 (0.87)	354 (62.0)	9.45 (278)
16	Vitrified	180	100	16.2 (6.9)	58 (1.47)	33.40 (0.85)	189 (33.1)	5.72 (168)

Optimization results

Table 4. Optimization Results with BP and the Boltzmann Factor

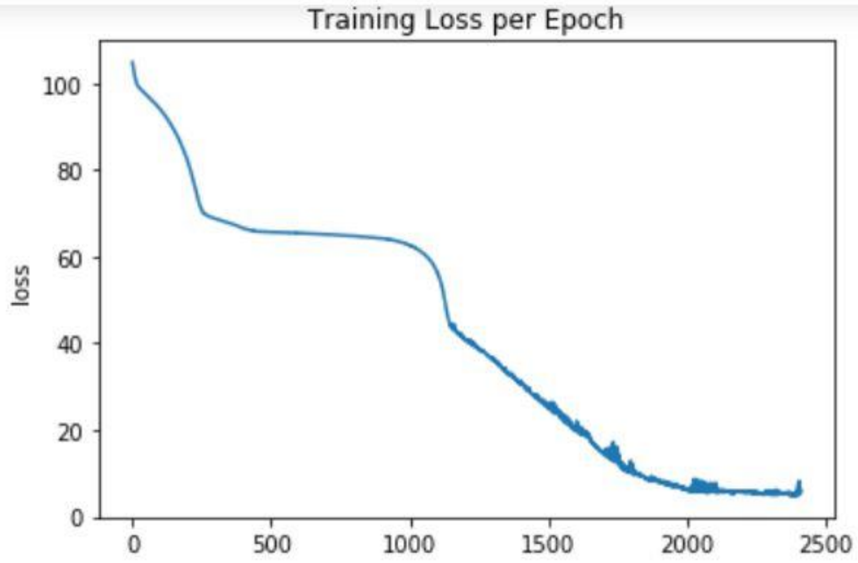
Search no.	Number of iterations	Optimal inputs					Optimal outputs			Optimal solution F
		B	m	c	v_w, in/min (mm/s)	d, 10^{-3} in (mm)	R_a, μin (μm)	F_n, lb/in (N/mm)	P_w, Hp/in (kW/m)	
1	134	Vitrified	180	50	16.2 (6.9)	63.17 (1.6)	31.72 (0.81)	182 (31.9)	5.65 (166)	197.73
2	291	Vitrified	180	50	16.2 (6.9)	63.17 (1.6)	31.72 (0.81)	182 (31.9)	5.65 (166)	197.73
3	161	Vitrified	180	50	16.2 (6.9)	63.17 (1.6)	31.72 (0.81)	182 (31.9)	5.65 (166)	197.73
4	188	Vitrified	180	50	16.2 (6.9)	63.17 (1.6)	31.72 (0.81)	182 (31.9)	5.65 (166)	197.73
5	145	Vitrified	180	50	16.2 (6.9)	63.17 (1.6)	31.72 (0.81)	182 (31.9)	5.65 (166)	197.73

Modelling using neural networks

The neural network has to predict 3 variables; Surface finish (Ra), normal grinding force per unit width (F,), and grinding power per unit width (Pw) simultaneously.
The difference in the order of magnitude of the 3 outputs is substantial. I.e. power is in the order of hundreds, force is in the order of tens and finish is less than 1. Thus the MSE loss function prioritized the loss in power over the others and hence surface finish was not able to be modelled correctly. Hence I used mean absolute percentage loss as the loss function instead of MSE which was used in the paper. The architecture of my neural network is 5 -15-10-10-5-5- 3 with adam optimizer and over 3000 epochs.

Neural networks was modelled and trained in tensorflow using keras.

Loss of the network




```
]: training_score = model.evaluate(X_train,y_train,verbose=0)
   val_score = model.evaluate(X_val,y_val,verbose=0)

]: val_score

]: 6.666107177734375

]: training_score

]: 4.930661201477051
```

Predictions using my neural network model

	Ra	Fn	Pw	Predicted Ra	Predicted Fn	Predicted Pw	Error
0	0.86	45.3	224.0	0.885605	44.684059	208.940201	3.686732
1	0.84	66.3	344.0	0.817917	70.173286	328.931610	4.283767
2	0.88	65.3	314.0	0.860412	63.754124	299.029724	3.120304
3	0.68	56.5	239.0	0.661269	48.090286	223.973923	7.975333
4	0.90	100.3	584.0	0.898322	96.648186	454.584290	8.662503
5	0.85	79.5	389.0	0.851513	79.693810	374.118744	1.415778
6	0.89	72.8	344.0	0.886053	73.544640	345.399353	0.624391
7	0.70	114.5	643.0	0.701642	98.733368	463.489471	13.974084
8	0.87	37.5	165.0	0.866918	34.966625	163.063034	2.761283
9	0.57	32.4	108.0	0.571043	23.586077	107.987572	9.132657
10	0.79	27.7	105.0	0.807620	22.720039	104.969040	6.746025
11	0.83	28.4	149.0	0.835271	30.948351	143.898041	4.344094
12	0.89	38.5	179.0	0.908373	41.457371	193.944016	6.031499
13	0.60	71.6	359.0	0.619985	67.590164	316.087921	6.961445
14	0.89	66.2	275.0	0.880609	57.444946	269.454529	5.432290
15	0.86	30.5	168.0	0.850906	37.634304	175.611145	9.659668

Research paper results vs mine

R_a, μmin (μm)	F_n, lb/in (N/mm)	P_w, Hp/in (kw/m)
34.35 (0.87)	256 (44.8)	6.98 (205)
32.92 (0.84)	379 (66.3)	11.53 (339)
34.34 (0.87)	369 (64.6)	9.89 (291)
26.51 (0.67)	313 (54.8)	8.83 (259)
34.32 (0.87)	567 (99.2)	19.50 (573)
33.28 (0.85)	438 (76.7)	13.43 (394)
34.36 (0.87)	405 (70.9)	10.98 (322)
27.13 (0.69)	636 (111.3)	21.86 (642)
34.28 (0.87)	203 (35.5)	5.75 (169)
22.73 (0.58)	170 (29.8)	4.19 (123)
32.08 (0.81)	160 (28.0)	4.12 (121)
33.04 (0.84)	169 (29.6)	4.81 (141)
34.33 (0.87)	226 (39.6)	6.11 (179)
24.20 (0.61)	437 (76.5)	12.41 (365)
34.36 (0.87)	354 (62.0)	9.45 (278)
33.40 (0.85)	189 (33.1)	5.72 (168)

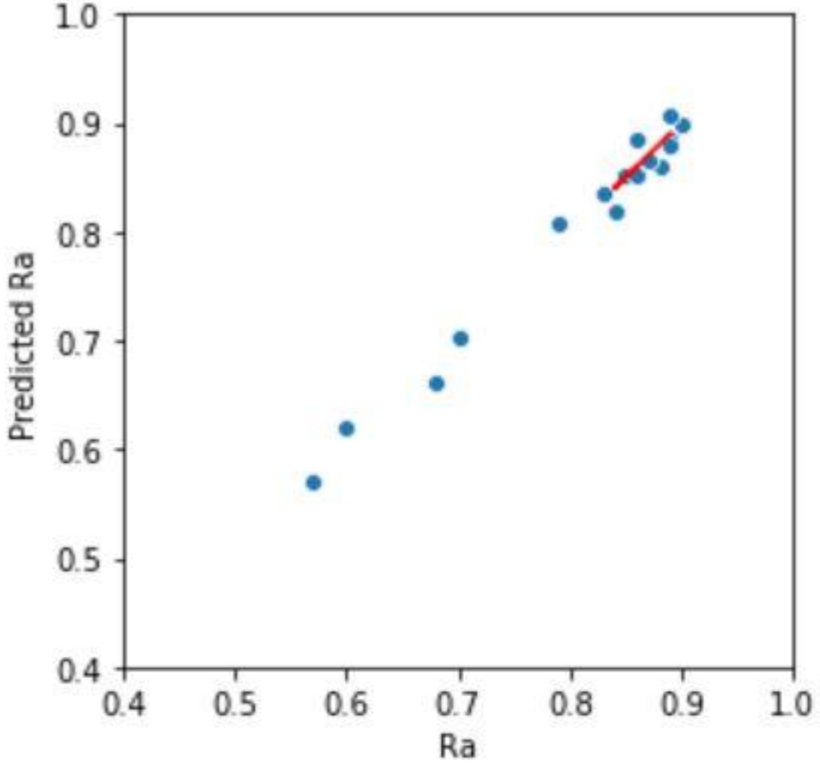
	Ra	Fn	Pw	Predicted Ra	Predicted Fn	Predicted Pw	Error
0	0.86	45.3	224.0	0.885605	44.684059	208.940201	3.686732
1	0.84	66.3	344.0	0.817917	70.173286	328.931610	4.283767
2	0.88	65.3	314.0	0.860412	63.754124	299.029724	3.120304
3	0.68	56.5	239.0	0.661269	48.090286	223.973923	7.975333
4	0.90	100.3	584.0	0.898322	96.648186	454.584290	8.662503
5	0.85	79.5	389.0	0.851513	79.693810	374.118744	1.415778
6	0.89	72.8	344.0	0.886053	73.544640	345.399353	0.624391
7	0.70	114.5	643.0	0.701642	98.733368	463.489471	13.974084
8	0.87	37.5	165.0	0.866918	34.966625	163.063034	2.761283
9	0.57	32.4	108.0	0.571043	23.586077	107.987572	9.132657
10	0.79	27.7	105.0	0.807620	22.720039	104.969040	6.746025
11	0.83	28.4	149.0	0.835271	30.948351	143.898041	4.344094
12	0.89	38.5	179.0	0.908373	41.457371	193.944016	6.031499
13	0.60	71.6	359.0	0.619985	67.590164	316.087921	6.961445
14	0.89	66.2	275.0	0.880609	57.444946	269.454529	5.432290
15	0.86	30.5	168.0	0.850906	37.634304	175.611145	9.659668

Results of modelling using decision trees

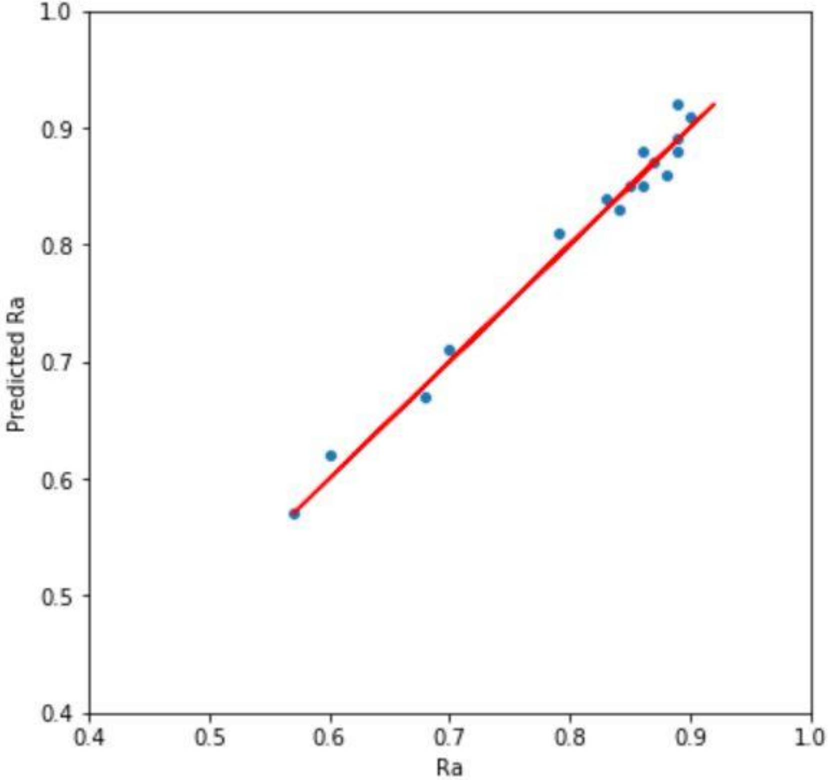
	Ra	Fn	Pw	Predicted Ra	Predicted Fn	Predicted Pw	% Error	mean Error
0	0.86	45.3	224.0	0.88	43.4	209.0	4.405423	5.640000
1	0.84	66.3	344.0	0.83	66.2	329.0	1.900590	5.036667
2	0.88	65.3	314.0	0.86	64.2	314.0	1.319087	0.373333
3	0.68	56.5	239.0	0.67	55.5	224.0	3.172217	5.336667
4	0.90	100.3	584.0	0.91	95.6	599.0	2.788515	6.570000
5	0.85	79.5	389.0	0.85	74.9	374.0	3.214068	6.533333
6	0.89	72.8	344.0	0.89	68.6	329.0	3.376565	6.400000
7	0.70	114.5	643.0	0.71	109.6	658.0	2.680287	6.636667
8	0.87	37.5	165.0	0.87	36.8	150.0	3.652525	5.233333
9	0.57	32.4	108.0	0.57	30.5	108.0	1.954733	0.633333
10	0.79	27.7	105.0	0.81	27.3	105.0	1.325230	0.140000
11	0.83	28.4	149.0	0.84	27.3	144.0	2.811254	2.036667
12	0.89	38.5	179.0	0.92	41.5	194.0	6.514294	6.010000
13	0.60	71.6	359.0	0.62	80.3	374.0	6.554148	7.906667
14	0.89	66.2	275.0	0.88	60.7	269.0	3.871190	3.836667
15	0.86	30.5	168.0	0.85	34.0	179.0	6.395273	4.836667

Predicted outputs vs real output

neural networks

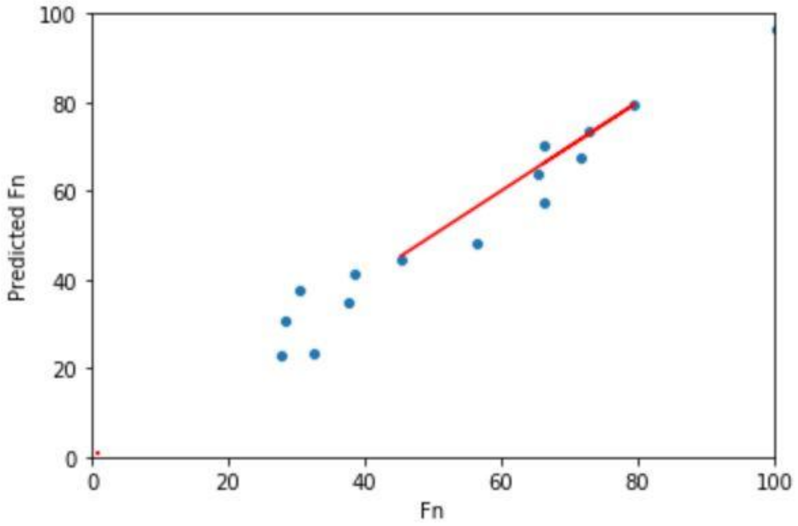


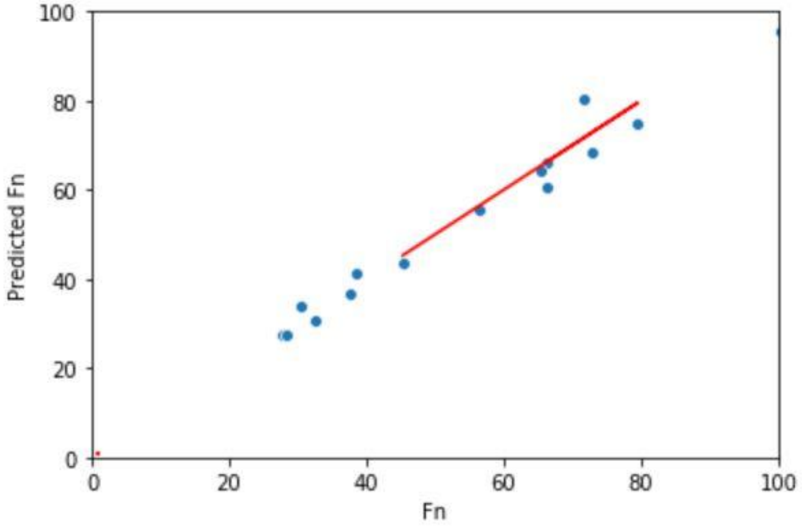
decision trees



neural networks

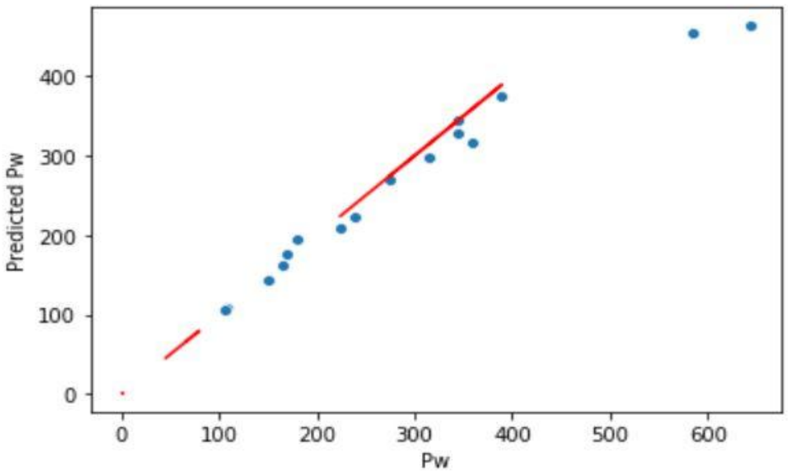
decision trees

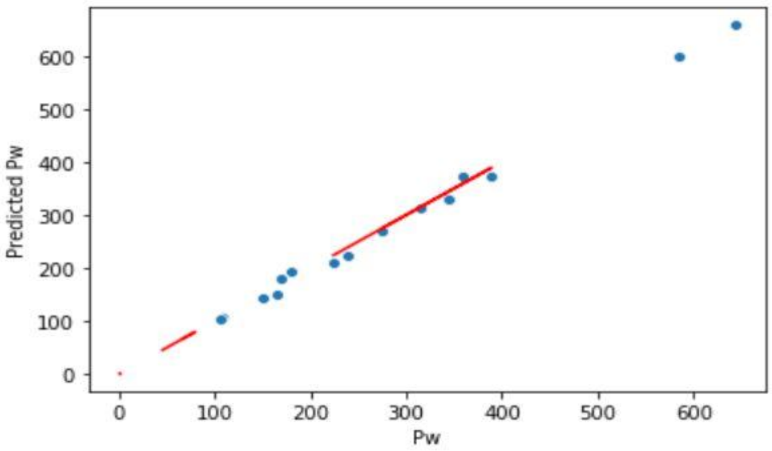




neural networks

decision trees





Optimization

There are in total four objectives considered: maximizing MRR' (= Vw*d), minimizing Pw, minimizing Fn and minimizing Ra.

The constraints of the input and output data are as follows :

1. $6.8 \leq Vw \geq 16.2$
2. $58 \leq d \geq 102$
3. $Ra \leq 33$
4. $Pw \leq 53.6$
5. bond type (B) is either resinoid (0) or vitrified (1)
6. mesh size (m) is either 80, 100, 120, 140, 160 or 180
7. concentration (c) is either 50, 75, or 100

The objective function is : F = -CI*MRR' + C2*Ra + C3*Fn + C4*P
The aim is to minimise this objective function.

Optimization technique

The optimization was done by generating 20,000 combinations of input variables and used the machine learning model to predict output variables. The optimization function was calculated for every data point and the minimum is chosen.

Neural networks

	My result	Research paper results
B	1.00	1.00
m	100.00	180.00
c	50.00	50.00
Vw	2.90	6.90
d	1.47	1.60
Ra	0.69	0.81
Fn	18.88	31.90
Pw	86.31	166.00
MRR	4.26	11.04
F	628.98	1172.76

Decision trees

	My result	Research paper results
B	1.00	1.00
m	80.00	180.00
c	50.00	50.00
Vw	4.77	6.90
d	2.00	1.60
Ra	0.81	0.81
Fn	27.30	31.90
Pw	105.00	166.00
MRR	9.52	11.04
F	786.71	1172.76

Conclusion

- The grinding process was successfully modelled using neural networks and decision trees.
- The decision trees algorithm(<1sec) was faster than neural networks(~1 min) while being more accurate.
- Optimal input parameters were obtained to minimise Power, Force and maximize material removal rate.

References

- A neural network approach for grinding processes: modelling and optimization, T. Warren Lio't and L. J. Chent (1993)