

Prediction of Material Removal Rate and Surface Roughness in CNC Turning of Delrin Using Various Regression Techniques and Neural Networks and Optimization of Parameters Using Genetic Algorithm



Susheem Kanwar, Ranganath M. Singari, and Vipin

Abstract This paper aims to compare the different regression techniques in conventional turning of a cylindrical workpiece of Acetal Homopolymer Delrin and to determine the most accurate one among them for determining the MRR and SR. Three different cutting parameters, namely feed (mm/rev), depth of cut (mm) as well as speed (RPM), are varied, and the corresponding MRR and surface roughness are represented by a Taguchi L27 orthogonal array. The orthogonal array is divided into training as well as testing datasets by making use of the `train_test_split` functionality in python. The testing data is one-third of the entire dataset with the remaining data forming the training data. In each case, the mean square error (MSE) is determined by contrasting obtained values with the output data present in testing dataset. Out of all the functions formulated, the neural network (NN) gives the least mean square error of 0.108. Genetic algorithm (GA) is then applied to optimize the input parameter values. It took 139 generations to achieve the optimum value of 1.735 μm for SR and 827.473 mm^3/min for MRR. This combination of resulting values was obtained at 299.887 rpm, 0.59 mm/rev feed and 1.49 mm depth of cut.

Keywords Regression · Taguchi · Mean square error · Neural network · Genetic algorithm

1 Introduction

Delrin is a thermoplastic polymer trademarked by DuPont and commonly known as polyoxymethylene (POM). It has properties similar to some metals and therefore is a potential alternative. Regression is applied to the data to determine a continuous output. It is a supervised learning operation in which the algorithm is trained on data, where it figures out the correlation among input and output variables and then

S. Kanwar (✉) · R. M. Singari · Vipin
Delhi Technological University, New Delhi, Delhi 110042, India
e-mail: sushkanwar@gmail.com

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2021
R. M. Singari et al. (eds.), *Advances in Manufacturing and Industrial Engineering*,
Lecture Notes in Mechanical Engineering,
https://doi.org/10.1007/978-981-15-8542-5_4

makes predictions for new output values given new input values. In this paper, the algorithms are fed the rake angle, speed and feed as input and the corresponding MRR and surface roughness values as output. Higher rake angle usually gives better surface finish [1]. When graphite and polymer composites are cut orthogonally, it is found that surface finish is extremely poor when the rake angle is between 0° and 5° . On increasing the rake angle, the concavities formed on the machine surface decrease resulting in improved surface finish [2]. The rake angle generally used varies from about 6° to 20° , and it can even reach up to 30° under certain conditions [3]. We found the surface roughness to be minimized in this region.

One of the problems with machining is that built-up edge (BUE) is created. If machining parameters are not selected carefully, then there is a build-up of layer at the tool edge called BUE and at the tool-rake interface known as built-up layer (BUL) [4–7]. Especially, when the aluminium undergoes dry machining, the effect of BUE and BUL is more pronounced. However, the environmental concerns accompanying the utilization of coolants and their disposal make dry machining an attractive alternative. To offset the disadvantage of BUE and BUL, dry machining of aluminium is accompanied by the use of high rake angles (about 30°) and ultra-hard tool materials (like diamond and CBN) [8].

Apart from surface finish, MRR is also of tremendous interest to us. Rake angle does not have as major an effect on MRR as depth of cut (DOC). DOC followed by speed is more influential in determining the MRR in a machining operation [9]. However, on increasing the rake angle and keeping the other parameters constant, there is slight decrease in the MRR [10].

Another important machining parameter is the feed rate. Feed has a greater effect on SR than other machining parameters [11, 12]. Feed is related to surface roughness by the formula $R_t = f^2/8r$ [13], where f = feed rate and R_t = peak-to-valley surface roughness. Thus, feed rate and surface finish share an inverse relationship. On the other hand, the MRR is directly proportional to feed rate [14].

Speed plays a major role in determining output parameters like MRR and surface finish. As speed increases, it leads to an increase in MRR as well while reducing the surface finish [15]. Speed and feed dominate the MRR of the workpiece [15].

There are several regression techniques which were used in predicting the SR and MRR like:

1.1 Linear Regression

Linear establishes a relationship between independent variables and dependent variables in an equation [16]. It is one of the most widely used regression analysis techniques because of its simplicity and the fact that a dataset which varies linearly is easier to fit than a nonlinear dataset [17, 18]. For n points present in a dataset, linear regression models assume a linear correlation between x and y . No relation is perfect, and often there is some error/noise which gets incorporated into the equation as shown below:

$$y_i = \beta_0 + \beta_1 x_1 + \dots + \beta_n x_n + \varepsilon_i = x^T \beta + \varepsilon_i \text{ where } i = 1, \dots, n$$

1.2 KNN Regression

It is among the simplest algorithmic techniques in machine learning. In KNN algorithm, the output value is the average of K nearest values. KNN regression technique is a non-parametric type of regression analysis [19].

1.3 Support Vector Regression (SVR)

SVRs come under the category of supervised learning models that predict a continuous output value for a given input. They come under the category of support vector machine (SVM) or support vector network [20]. Our goal in SVR regression is to determine a function that maximizes deviations for all the data points [21]. Errors less than the threshold value of ε are considered negligible, but those greater than that are unacceptable [22]. Linear support vector regression has a general equation:

$$y = \sum_i = 1 (a_i - a) \langle x_i, x \rangle + b$$

1.4 Bayesian Ridge

Target value is a linear combination of input values. Bayesian regression includes a regularization parameter. Bayesian ridge regression estimates β using L2-constrained least squares [23]. Bayesian ridge regression has a greater than quadratic fit time as the number of samples makes scaling hard. In contrast to the ordinary least squares (OLS) estimator, the weights are shifted toward zeros, which lends stability. By maximizing the marginal log-likelihood over a number of iterations, estimation is achieved. There are several implementation strategies for Bayesian ridge. The implementation mentioned in this paper is taken from [24]. Moreover, better values of the regularization parameters inspired from the recommendation in [25].

1.5 Decision Tree Regression

The core algorithm called ID3 involves breaking the dataset into increasingly smaller subsets represented by a combination of decision as well as leaf nodes [26]. Numerous connections branch forth from a decision node, each of which represents an attribute

while leaf nodes represent a decision. Node corresponding to the best predictor is referred to as the root node.

1.6 Gradient Boosting Regression

It is used to make predictions by using an ensemble of ML models. The various ML models are generalized by allowing optimizing on an appropriate cost function [27]. Gradient boosting model helps in the optimization of any arbitrary differentiable loss functions. Training set of the form $\{(x_1, y_1), \dots, (x_n, y_n)\}$ used to determine approximation $F(x)$ to minimize the loss function

$$L(y, F(x)) : F = \arg \min E_x, y [L(y, F(x))].$$

1.7 Neural Networks

Simulate the neurons in the human brain. Artificial neural networks consist of a minimum of three layers, viz. the input, output and hidden layer. The connection between is assigned weights which can be positive, negative or zero [28]. The NN learns the correlation present in the data through repetition. More the number of repetitions, better the NN learns. Once it has learned the relationship, it can generalize to previously unseen data and can thus predict output values for new input data.

1.8 Genetic Algorithm (GA)

GA is an evolutionary algorithm which we have utilized for optimization. In GA, we start off with an initial population which may be randomly generated. We then select the fittest members from this initial population based on various criteria and make them pass their “genes” to the next generation. This step is known selection. “Offspring” are created by mating the parent population selected in the previous step. This process keeps on repeating until the point where the offspring produced are not much different from the parents. This point is known as convergence [29–31].

2 Experimental Investigations

In this paper, three different cutting parameters, namely feed (mm/rev), depth of cut (mm) as well as speed (RPM), vary, and the corresponding MRR (mm³/min) and

surface roughness (micrometres) are represented by a Taguchi L27 orthogonal array. The `train_test_split` function of `sklearn` splits the dataset further into two categories: training and testing. Testing data represents one-third of the entire dataset, with the rest being training data.

A homopolymer Delrin rod of diameter 34 mm was selected, and CNC turning operation was performed on it. Before performing final CNC turning operation, a roughing operation was performed on it. At the end of the turning operation, the diameter was reduced from 34 to 33 mm. In the CNC turning operation, three different depths of cuts were provided: 0.5, 1.0 and 1.5 mm. The entire rod was broken into three pieces of equal length, and all the operations for a particular depth of cut were performed on each rod in succession. Various regression techniques were then applied on the dataset, and the mean square error was calculated to determine the accuracy of the ML regression models. The `scikit-learn` library of python was used for implementing the various regression techniques [32]. In addition to regression techniques, a neural network has also been implemented to predict the MRR and surface roughness using the `Keras` library [33]. `Matplotlib` library has been utilized to plot the experimental MRR and SR values along with the predicted MRR and SR values [33]. `MATLAB` optimization toolbox is used to implement multi-objective genetic algorithm to find the optimized values of the input variables [34]. CNC turning experimental data obtained was arranged in a L27 orthogonal array (Table 1).

3 Results and Discussion

It was found that the least mean square error while predicting both MRR and SR was obtained by NN. The mean square error obtained, for each regression technique, is represented in Table 2. As can be seen from the table, neural net gives the best overall results, as it has an extremely small MSE.

The above results can be confirmed by visualizing the values obtained using regression techniques with the true values of both MRR and SR on two different graphs (the graph for neural network has not been included as the values for NN have been reshaped and will have to be represented by another scale than shown in Figs. 1 and 2).

Out of the three regressions, viz. linear, SVR and Bayesian ridge regression, linear regression has the least error for both MRR and SR. For linear regression, the equation found is:

$$y = [[0.0092381 \quad -1.31666667 \quad -1.15333333] \\ [-6.41442733 \quad -886.39633333 \quad -65.8162]] * x \\ + [1.4800000000000002 \quad 3380.573266666667].$$

Table 1 L27 orthogonal array

Exp. no.	Control factors			Speed (<i>s</i>) (rpm)	Feed (<i>f</i>) (mm/rev)	Depth of cut (<i>d</i>) (mm)	Surface roughness (<i>Ra</i>) (μm)	MRR (mm^3/min)
	A	B	C					
1	1	1	1	150	0.2	0.5	1.04	1083.77
2	1	1	2	150	0.2	1.0	2.44	1784.15
3	1	1	3	150	0.2	1.5	0.56	1120.99
4	1	2	1	150	0.4	0.5	0.90	1665.85
5	1	2	2	150	0.4	1.0	1.42	1568.58
6	1	2	3	150	0.4	1.5	0.96	1898.23
7	1	3	1	150	0.6	0.5	0.86	1702.32
8	1	3	2	150	0.6	1.0	1.14	2505.56
9	1	3	3	150	0.6	1.5	1.16	1933.68
10	2	1	1	250	0.2	0.5	2.62	2140.88
11	2	1	2	250	0.2	1.0	5.38	2216.46
12	2	1	3	250	0.2	1.5	0.52	2538.14
13	2	2	1	250	0.4	0.5	2.32	1811.81
14	2	2	2	250	0.4	1.0	3.58	2469.70
15	2	2	3	250	0.4	1.5	0.64	2040.10
16	2	3	1	250	0.6	0.5	1.92	1410.84
17	2	3	2	250	0.6	1.0	3.44	1532.72
18	2	3	3	250	0.6	1.5	0.94	1579.62
19	3	1	1	300	0.2	0.5	2.04	1047.48
20	3	1	2	300	0.2	1.0	4.80	960.00
21	3	1	3	300	0.2	1.5	1.00	1085.79
22	3	2	1	300	0.4	0.5	2.24	251.998
23	3	2	2	300	0.4	1.0	3.84	461.219
24	3	2	3	300	0.4	1.5	0.80	980.251
25	3	3	1	300	0.6	0.5	2.70	143.928
26	3	3	2	300	0.6	1.0	3.76	0.0035
27	3	3	3	300	0.6	1.5	0.74	0.1408

The equation obtained on application of linear regression was taken, and genetic algorithm (GA) was applied to it for optimizing the input variables. In order to implement GA, the optimization toolbox in MATLAB was used. In this experiment, the MATLAB genetic algorithm was selected in the optimization toolbox. The following parameters are used during the optimization: An initial population of 50 with feasible population as the function and tournament type with a crossover of 0.8. We also selected a single-point crossover and mutation which is adaptive feasible.

Table 2 Mean square error obtained for MRR and SR, which is calculated using various regressors

	Mean square error
Linear regression	0.46
Support vector regression	2.15
KNN regression	0.275
Bayesian ridge	1.981
Decision tree regression	0.647
Gradient boosting regression	0.86
Neural networks	0.108

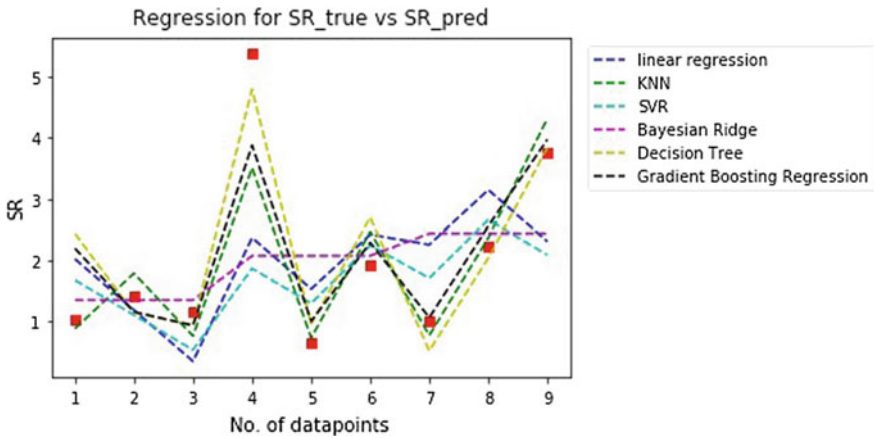


Fig. 1 Obtained and true values of surface roughness (plotted using matplotlib library in python)

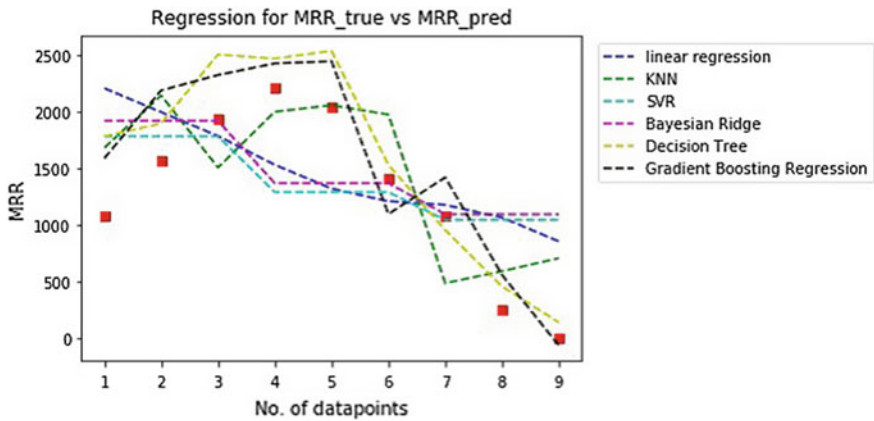


Fig. 2 Obtained and true values of material removal rate (MRR) (plotted using matplotlib library in python)

3.1 Optimization Using Genetic Algorithm (GA)

As can be seen from Fig. 3, the multi-objective GA iterates for obtaining the best solution and finds it on the 139th generation. The graph in Fig. 5 plots Objective 2 on the y-axis against Objective 1 on the x-axis giving the Pareto front. Average speed for each generation is determined in Fig. 7. The score diversity is represented in a histogram in Fig. 4 while Fig. 6 plots each individual's rank. These graphs help us determine the optimized solutions.

Fig. 3 Average distance

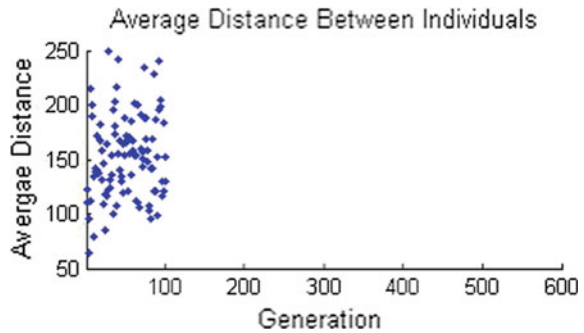


Fig. 4 Score diversity

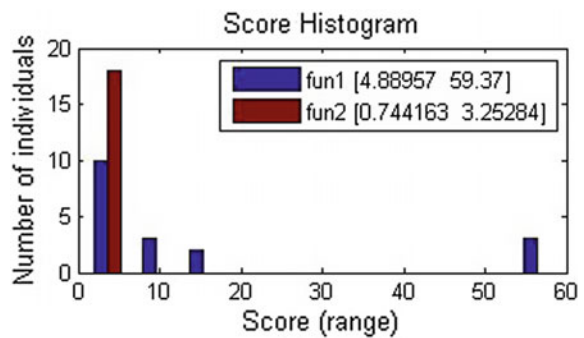


Fig. 5 Pareto front graph

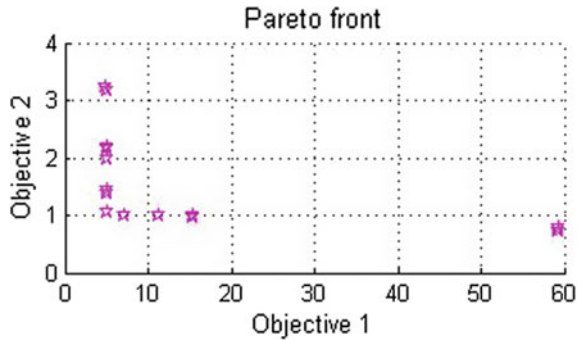


Fig. 6 Rank histogram

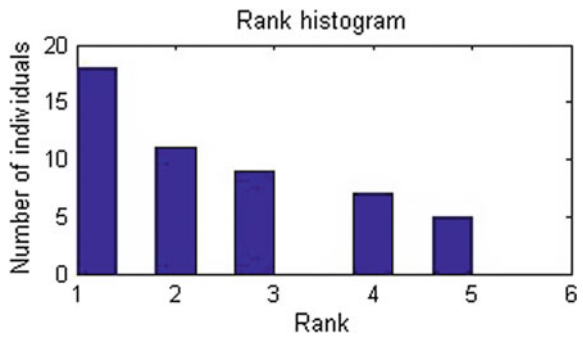
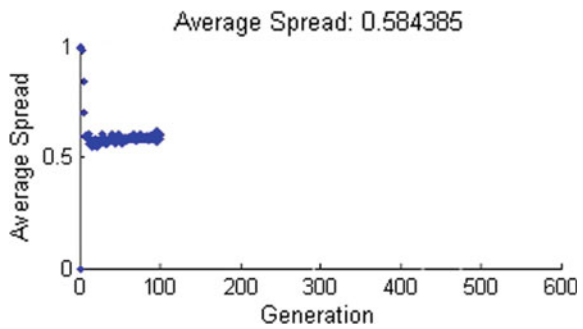


Fig. 7 Average Pareto spread



4 Conclusion

Prediction of SR and the material removal rate through regression allows us to conclude the following:

1. Neural networks give the least mean square error of 0.108 and are thus an improvement over regression.

2. In the graph showing comparison of regression techniques, it shows that KNN regression has the best fit.
3. On applying genetic algorithm, we find that optimization takes 139 generations which is quite fast.
4. The optimum cutting parameters are 150 rpm 0.6 mm/rev feed and 1.49 mm depth of cut. At this combination, SR is 0.351 μm and MRR is 1788.91 mm^3/min .
5. Increasing feed while keeping other factors constant resulted in a decrease in the surface finish of the workpiece.

References

1. Pradeesh AR, Mubeer MP, Nandakishore B, Muhammed Ansar K, Mohammed Manzoor TK, Muhammed Raees MU (2016) Effect of rake angles on cutting forces for a single point cutting tool. *Int Res J Eng Technol (IRJET)* 3(05)
2. Yang D, Wan Z, Xu P, Lu L (2018) Rake angle effect on a machined surface in orthogonal cutting of graphite/polymer composites. *Adv Mater Sci Eng*
3. Baldoukas AK, Soukatzidis F, Lontos A, Demosthenous G (2008) Experimental investigation of the effect of cutting depth, tool rake angle and workpiece material type on the main cutting force during a turning process. In: *Proceedings of the 3rd international conference on manufacturing engineering (ICMEN)*, Ed. Ziti, Greece
4. Sanchez-Carrilero M, Marcos M (2011) SEM and EDS characterisation of layering TiO_x growth onto the cutting tool surface in hard drilling processes of Ti-Al-V alloys. *Adv Mater Sci Eng*
5. Trent ME, Wright PK (2000) *Metal cutting*, 4th edn. Butterworth-Heinemann, UK
6. Carrilero MS, Bienvenido R, Sanchez JM, Alvarez M, Gonzalez A, Marcos M (2002) A SEM and EDS insight into the BUL and BUE differences in the turning processes of AA2024 Al-Cu alloy. *Int J Mach Tools Manuf* 42(2):215–220
7. Sánchez-Sola JM, Sebastian M (2005) Characterization of the built-up edge and the built-up layer in the machining process of AA 7050 alloy. *Revista de Metalurgia* 365–368
8. Sreejith PS, Ngoi BKA (2000) Dry machining: machining of the future. *J Mater Process Technol* 101(1–3):287–291
9. Kumar S, Gupta D (2016) To determine the effect of machining parameters on material removal rate of aluminium 6063 using turning on lathe machine. *Int J Multidiscip Curr Res* 4
10. Pant G, Kaushik S, Rao DK, Negi K, Pal A, Pandey DC (2017) Study and analysis of material removal rate on lathe operation with varying parameters from CNC Lathe machine. *Int J Emerg Technol* 8(1):683–689
11. Singhvi S, Khidiya MS, Jindal S, Saloda MA (2016) Investigation of material removal rate in turning operation. *Int J Innov Res Sci Eng Technol (IJIRSET)* 5(3)
12. Abdullah AB, Chia LY, Samad Z (2008) The effect of feed rate and cutting speed to surface roughness. *Asian J Sci Res* 1(1):12–21
13. Qu J, Shih A (2003) Analytical surface roughness parameters of a theoretical profile consisting of elliptical arcs. *Mach Sci Technol* 7(2):281–294
14. Gupta R, Diwedi A (2014) Optimization of surface finish and material removal rate with different insert nose radius for turning operation on CNC turning center. *Int J Innov Res Sci Eng Technol (IJIRSET)* 3(6)
15. Bhavani TK, Satyanarayana K, Kumar GSV, Kumar IA (2017) Optimization of material removal rate and surface roughness in turning of aluminum, copper and gunmetal materials using RSM. *Int J Eng Res Technol (IJERT)* 6(02)
16. Freedman D (2009) *Statistical models: theory and practice*, 2nd edn. Cambridge University Press, Cambridge

17. Xin Y (2009) Linear regression analysis: theory and computing, 1st edn. World Scientific, Singapore
18. Montgomery D, Peck E, Vining G (2012) Introduction to linear regression analysis, 5th edn. Wiley Publishers, New York
19. Altman NS (1992) An introduction to kernel and nearest-neighbor nonparametric regression. *Am Stat* 46(3):175–185
20. Corinna C, Vladimir NV (1995) Support-vector networks. *Mach Learn* 20(3):273–297
21. Vapnik VN (2000) The nature of statistical learning theory, 2nd edn. Springer, New York
22. Smola A, Schölkopf B (2004) A tutorial on support vector regression. *Stat Comput* 14:199–222
23. Pasanen L, Holmström L, Sillanpää MJ (2015) Supporting information for Bayesian LASSO, scale space and decision making in association genetics. *PLoS ONE* 10(4)
24. Tipping ME (2001) Sparse Bayesian learning and the relevance vector machine. *J Mach Learn Res* 1
25. MacKay DJC (1992) Bayesian interpolation. *Comput Neural Syst* 4(3)
26. Quinlan JR (1986) Induction of decision trees. *Mach Learn* 1:81–106
27. Brelman L (1997) Arcing the edge. Technical Report486. Statistics Department University of California, Berkeley
28. Hopfield JJ (1982) Neural networks and physical systems with emergent collective computational abilities. *Proc Natl Acad Sci U S A* 79(8):2554–2558
29. Goldberg DE (1989) Genetic algorithms in search optimization and machine learning, 1st edn. Addison-Wesley Publishing Company Inc., Reading, MA
30. Carroll DL (1996) Chemical laser modeling with genetic algorithms. *AIAA J* 34(2):338–346
31. Winter G, Cuesta P, Periaux J, Galan M, Cuesta P (1996) Genetic algorithm in engineering and computer science. Wiley, New York
32. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: machine learning in python. *J Mach Learn Res* 12:2825–2830
33. Chollet F (2015) Keras. GitHub
34. Hunter JD (2007) Matplotlib: a 2D graphics environment. *Comput Sci Eng* 9(3):90–95