Comparing Machine Learning and Interpolation Methods for Loop-Level Calculations

Kerem Kezer

INTRODUCTION

- Computation of the scattering cross-section of a process requires the evaluation of computationally intensive integrals
- Machine learning and interpolation methods can reduce the calculation time of these integrals
- In this research I will compare the performance of 3 interpolation methods which are Nearest-Neighbors(NN), Inverse Distance Weighting(IDW), Radial Basis Function(RBF) and 1 machine learning method that is Light Gradient Boosting Machine(LGBM)

The Passarino-Veltman D_0 function

$$D_0\left(s_1,s_2,s_3,s_4,s_{12},s_{23},m_0,m_1,m_2,m_3\right)= C_0\int \mathrm{d}^dq\,\frac{1}{\left(q^2-m_0^2\right)\left[(q+p_2)^2-m_1^2\right]\left[(q+p_2+p_4)^2-m_2^2\right]\left[(q+p_2+p_3+p_4)^2-m_3^2\right]}$$

$$D_0^{(3)}=\mathrm{Re}\left[D_0\left(0.01,0.04,0.16,\frac{x_4}{4},1,u\left(x_6\right),\frac{x_7}{2},\frac{x_7}{2},\frac{x_7}{2},0.2\right)\right]$$

$$D_0^{(6)}=\mathrm{Re}\left[D_0\left(0.01,0.04,0.16,\frac{x_4}{4},1,u\left(x_6\right),x_7,x_8,x_9,x_{10}\right)\right]$$

$$D_0^{(9)}=\mathrm{Re}\left[D_0\left(\frac{x_1}{4},\frac{x_2}{4},\frac{x_3}{4},\frac{x_4}{4},1,u\left(x_6\right),\frac{x_7}{2},\frac{x_8}{2},\frac{x_9}{2},\frac{x_{10}}{2}\right)\right]$$

METHOD

- All calculations are done by Python
- For the data set, I uniformly sample the unit hypercube $[0,1]^d$
- Methods are applied on 4 functions which are Polynomial, Periodic, Camel and The Passarino-Veltman D_0 function.
- For each run, the number of unknown variables in the functions was increased (3,6,9).
- Methods are compared by symmetric absolute percent errors (sAPE)

$$f_{\mathrm{poly}}\left(\mathbf{x}\right)=\sum_{i=1}^{d}-x_i^2+x_i$$

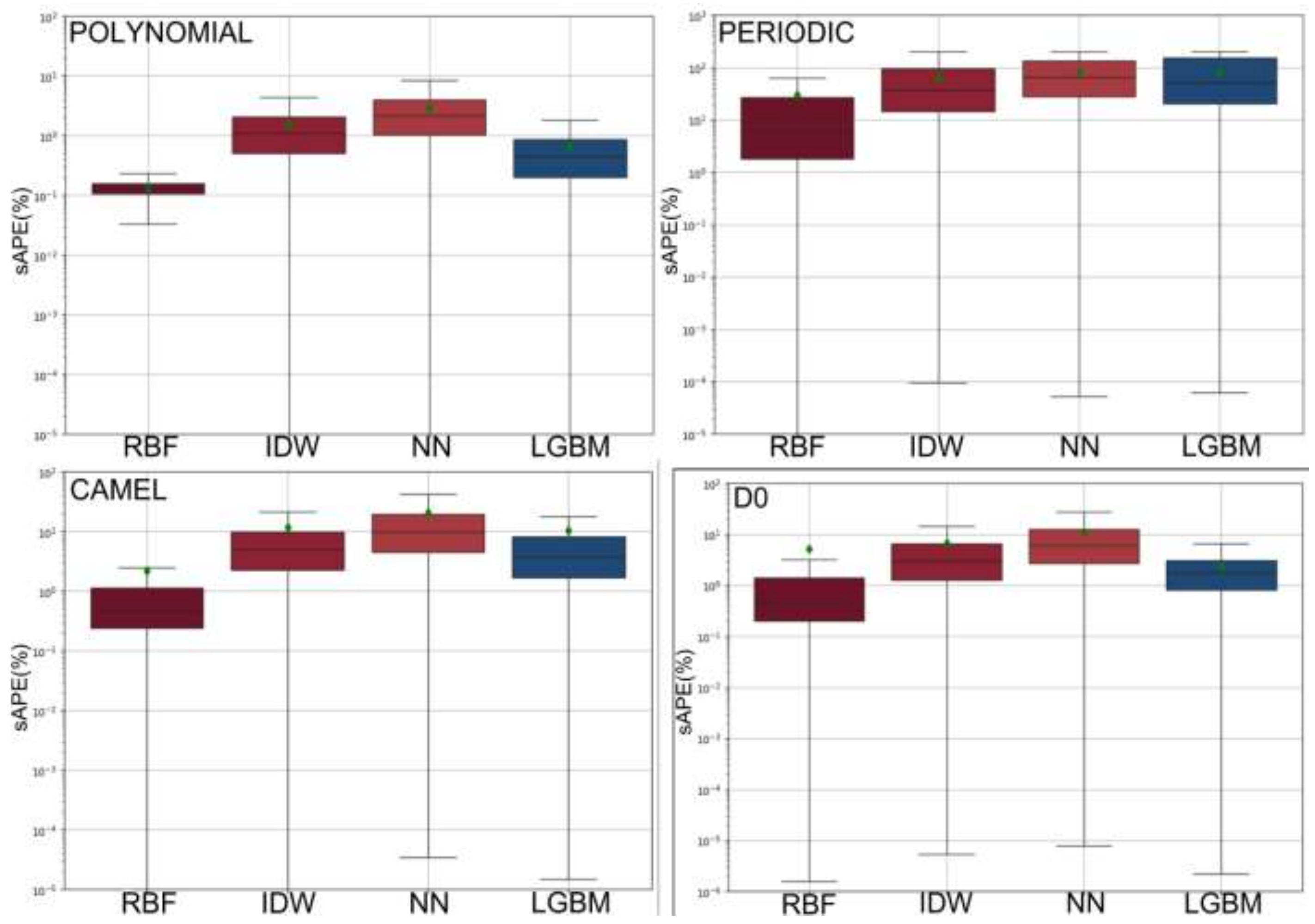
$$f_{\mathrm{periodic}}(\mathbf{x})=\bar{x}\prod_{i=1}^{d}\sin 2\pi x_i$$

$$f_{\mathrm{Camel}}(\mathbf{x})=\frac{1}{2(\sigma\sqrt{\pi})^n}\left(\exp\left(-\frac{\sum_i\left(x_i-\frac{1}{3}\right)^2}{\sigma^2}\right)+\exp\left(-\frac{\sum_i\left(x_i-\frac{2}{3}\right)^2}{\sigma^2}\right)\right)$$

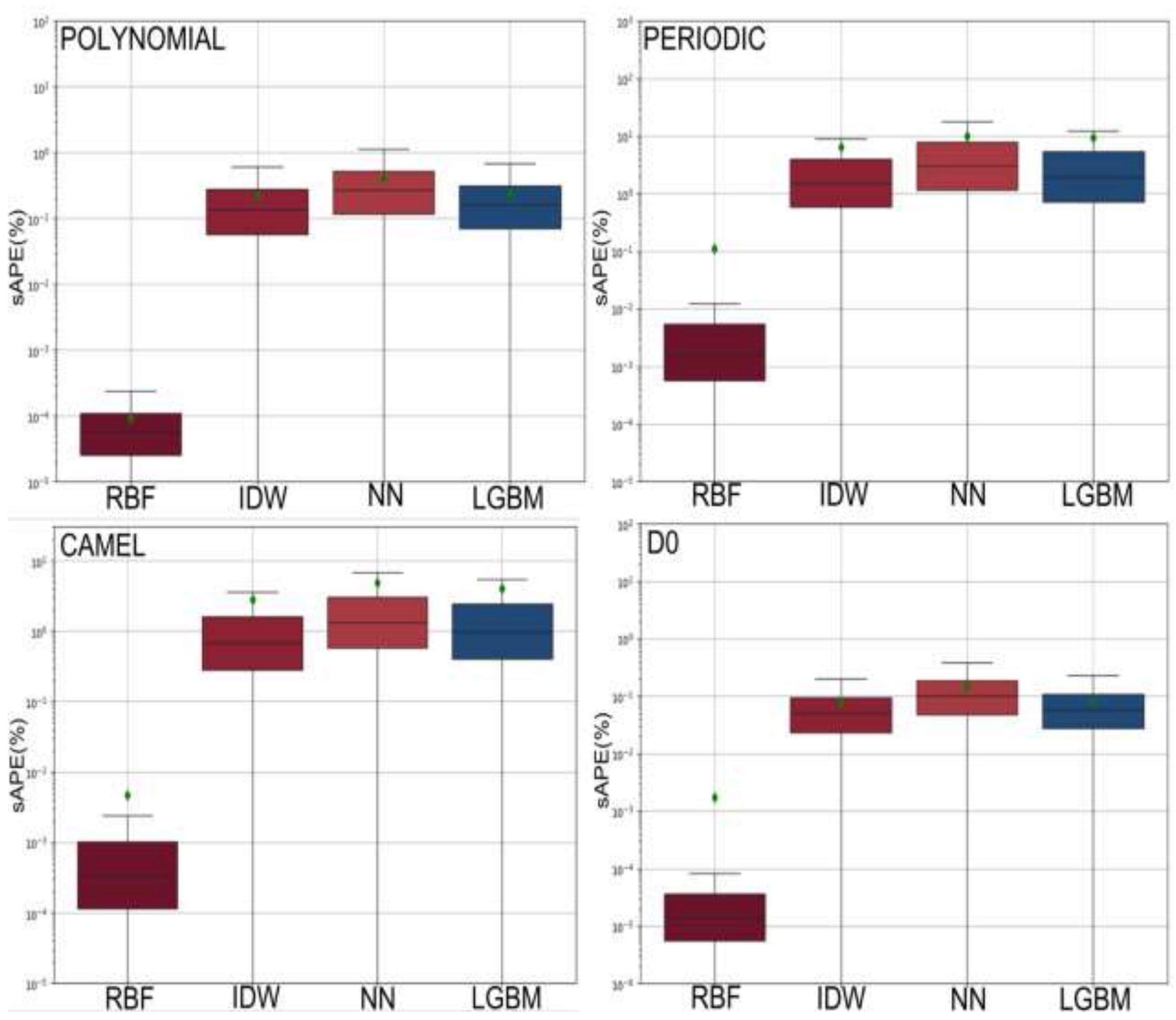
RESULTS AND DISCUSSION

- For 3 unknown variables, RBF achieves the lowest median and Q3 sAPE values. The other three methods have almost the same rate of estimating results
- For 6 unknown variables, RBF performs the best and NN performs the worst
- For 9 unknown variables, RBF shows the best performance, after which LGBM shows the second best performance for Polynomial and Camel Functions. For the periodic function, all four methods have almost the same mean value. For the D_0 function, LGBM showed the best performance in terms of mean value of sAPE
- The prediction rate of all methods decreases when we increase the number of unknown variables to 3, 6, and 9 respectively. In other words, as the number of unknown variables increased, the methods had worse sPAE values

6 Unknown Variables



3 Unknown Variables



9 Unknown Variables