Improving Predictive Accuracy in Surrogate Modeling of Plate Deflection through Physics-Informed Feature Engineering

https://doi.org/10.31713/MCIT.2025.075

Roman Onatskyi
PhD Student: Dept. of Mathematical Modelling and Intelligent Computing in Engineering
National Technical University
"Kharkiv Polytechnic Institute
Kharkiv, Ukraine
roman.onatskyi@infiz.khpi.edu.ua

Serhii Misiura
Associate Professor of the Department of Mathematical Modeling and Intelligent Computing in Engineering,
National Technical University
"Kharkiv Polytechnic Institute"
Senior Researcher of the Department of Vibration and Thermostability Studies, Anatolii Pidhornyi Institute of Power Machines and Systems of NAS of Ukraine
Kharkiv, Ukraine
serhii.misiura@khpi.edu.ua

Abstract — **This work proposes a physics-informed feature engineering approach to improve the predictive accuracy of surrogate models for problems in solid mechanics. The prediction of the maximum deflection of a thin square plate under a uniform load is considered as a case study. A dataset of 5000 samples was generated using the finite element method in Ansys Mechanical. This was followed by a physical validation based on the theory of small deflections, and invalid samples were discarded. A random forest regressor algorithm was used for the surrogate model, and its hyperparameters were optimized using RandomizedSearchCV. Two input feature architectures were compared: a baseline architecture (using fundamental physical parameters) and a physics-informed architecture (using complex engineering features, specifically relative flexibility K1 and cylindrical rigidity D). The results showed a significant increase in accuracy when using physics-informed features compared to the baseline approach. An analysis of feature importance confirmed the dominant role of K1 and the load p, which is fully consistent with theoretical mechanics. The obtained results demonstrate that feature engineering based on physical principles improves both the accuracy and interpretability of machine learning surrogate models.**

Keywords — finite element method; surrogate models; feature engineering; random forest; machine learning; plate deflection

I. Introduction

Modern design and analysis of engineering structures, especially thin plates and shells, are closely linked to the use of numerical methods like the Finite Element Method (FEM) [1, 2]. Despite their high accuracy, these methods are computationally expensive. This makes them difficult to use for optimization or uncertainty analysis tasks that require hundreds or thousands of simulations.

An effective alternative is the use of surrogate models based on machine learning [3,4]. Such models, trained on a limited amount of data from FEM simulations, can predict the system's behavior almost instantly with high accuracy. However, the quality of a surrogate model critically depends on two factors: the strategy for generating the training dataset and the architecture of the input features fed into the model.

This work investigates the influence of input feature architecture on the accuracy of a surrogate model, using the problem of predicting the maximum deflection of a thin square plate under a uniform load. The study aims to compare the effectiveness of two input data architectures: a baseline one, consisting of fundamental physical parameters, and a physics-informed one, which uses complex engineering features.

II. Problem Formulation

A. Data Generation and Preparation

To train the model, an initial dataset of 5000 points was generated using random uniform sampling. The input parameters were varied within the following ranges: plate size, a [0.05, 1.5] m; thickness, t [0.005, 0.01] m; Young's modulus, E [$1\cdot10^{10}$, $2\cdot10^{11}$] Pa; Poisson's ratio, ν [0.20, 0.45]; and load, p [$1\cdot10^{3}$, $1\cdot10^{5}$] Pa.

The target variable was the maximum plate deflection w, m.

The deflection value was obtained as a result of an FEM simulation in the Ansys Mechanical CAE system. The modeling and calculation were performed using the APDL (Ansys Parametric Design Language) scripting language [8].

The algorithm for construction and calculation is based on the same principles that were described in detail in the paper [6].

Since a random combination of parameters can lead to results that are outside the applicability of the small deflection theory, a physical validation and data filtering were performed according to the criteria of the Kirchhoff-Love [5] theory of small deflections. That is, $0.0001 < w/t < 0.5$, where 0.0001 is the relative limit of deflection measurability.

After filtering, the final dataset consisted of 2249 physically valid samples, or 44.98% of the initial volume. These samples were used for further training and testing.

B. Surrogate Model

The random forest regressor, an ensemble algorithm from the Python package sklearn.ensemble [7], was chosen as the surrogate model. This choice was made due to its high efficiency on tabular data, resistance to overfitting, and ability to capture complex non-linear dependencies without the need for data scaling.

For better visualization and interpretation of the results, the dataset was converted to the following units before machine learning: linear parameters (a, t, w) to mm, and Young's modulus and load values to MPa.

The regressor model was configured with the following hyperparameters to ensure an optimal balance between prediction accuracy and computational efficiency: the number of trees in the forest was set to 98; the maximum depth of each tree was limited to 11 levels; all available features were used for each split; the minimum number of samples for an internal node split was set to 5; and the minimum number of samples in a leaf node was set to 3.

To select the optimal hyperparameters for the regressor model, the RandomizedSearchCV method from the scikit-learn library [7] was used, which performs stochastic optimization in the hyperparameter space combined with cross-validation.

The optimization process included defining parameter ranges for key hyperparameters, applying 5-fold cross-validation, optimizing for the negative mean squared error metric, and limiting the process to 50 iterations. This approach allowed for a systematic exploration of the hyperparameter space to select the optimal settings for the model.

C. Input Feature Architectures

To study the influence of the input feature architecture on the quality of the surrogate model, two different approaches to forming the input data vector were considered.

Each architecture was designed considering the physical principles of the plate deflection prediction problem and provided a different level of integration of engineering knowledge into the machine learning process.

1) Baseline Architecture.

The baseline architecture used a set of fundamental physical parameters that directly characterize the plate's geometry, material properties, and loading. The input vector consisted of five parameters: [a (mm), t (mm), E (MPa), ν, p (MPa)]. This approach allowed the model to independently learn the complex non-linear relationships between the input parameters and the target variable w (mm) based solely on the patterns in the training data.

2) Physics-informed Architecture.

This architecture was based on an engineering analysis of physical dependencies and the construction of corresponding features. Based on the fundamental parameters, complex engineering features were created that integrate several basic quantities into single, physically meaningful characteristics:

- **K1**: the relative flexibility of the plate, defined as $K1 = a/t$, which characterizes the geometric flexibility of the structure.
- **D**: the cylindrical bending rigidity, defined as (1), which integrates the influence of material properties (E, ν) and geometry (t) on bending resistance.
- **p**: the intensity of the uniform load (MPa).

$$D = E \cdot t^3 / 12 \cdot (1 - \nu^2) \qquad (1)$$

This approach explicitly provided the model with information about the physical mechanisms of deformation, particularly the non-linear dependence of deflection on thickness, which contributed to more effective learning.

D. Quality Assessment

For each architecture, the model was trained on 80% of the data and tested on the remaining 20%. The quality was evaluated using the coefficient of determination (R^2) and the mean squared error (MSE).

III. Results

Training and testing the models on the filtered, physically correct data using the two feature architectures showed a significant difference in prediction quality.

The summary table of results for the two architectures is shown in Table 1. The model accuracy comparison for the "Baseline" architecture is shown in Fig. 1, and for the "Physics-informed" architecture in Fig. 2.

TABLE I. Comparison of Prediction Quality for the Two Proposed Architectures

Feature Architecture	R^2	MSE
Baseline	0.646	3.223
Physics-informed	0.896	0.946

A. Baseline Architecture

The model trained on fundamental parameters showed satisfactory but limited accuracy: $R^2 \approx 0.646$, MSE ≈ 3.223, see Fig. 1 for more details. The analysis revealed that the model captured general trends but had a significant scatter of predictions, especially for large deflection values. This indicates that independently learning the complex non-linear dependence of deflection on thickness is a difficult task for the model.

B. Physics-informed Architecture

After switching to engineering features, the model's quality significantly increased: $R^2 \approx 0.896$, MSE ≈

0.946, see Fig. 2 for more details. The feature importance analysis for the improved model showed that the largest contributions to the prediction were made by physically significant parameters: K1 (relative flexibility, ~59%) and p (load, ~24%), which fully aligns with theoretical mechanics. The introduction of the D parameter allowed the model to effectively account for the combined influence of Young's modulus and thickness.

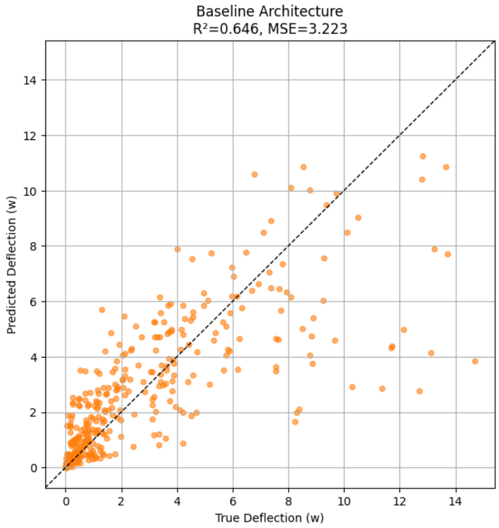

Figure 1. Comparison of model accuracy for the "Baseline"

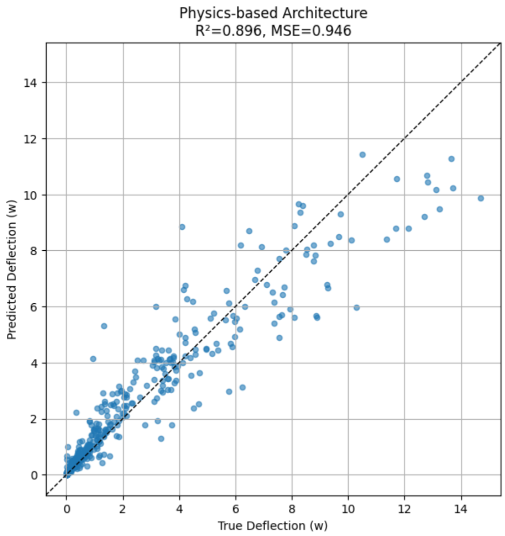

Figure 2. Comparison of model accuracy for the "Physics-informed"

This confirms that the model learned to make decisions based on the same complex criteria used by a design engineer.

IV. Conclusions

The conducted research demonstrates that the architecture of input features is a critically important factor when building surrogate models for problems in solid mechanics. The transition from a set of basic physical parameters to physics-informed features, such as relative flexibility (K1) and cylindrical rigidity (D), significantly increases the prediction accuracy of a model based on the random forest regressor algorithm. Feature engineering not only improves quality metrics but also enhances the model's interpretability, as it makes decisions based on the same physical principles used in classical theoretical mechanics, and generally improves its ability to generalize.

References

[1] K. J. Bathe, Finite element procedures, 2nd ed. Watertown, MA: Klaus-Jurgen Bathe, 2014, 1037 p.

[2] O. C. Zienkiewicz, R. L. Taylor, and S. Govindjee, The finite element method: its basis and fundamentals, 8th ed. Oxford: Butterworth-Heinemann, 2024, 650 p.

[3] R. A. Babudzhan, O. O. Vodka, and M. I. Shapovalova, "Application of computational intelligence methods for the heterogeneous material stress state evaluation," Herald of Advanced Information Technology, vol. 5, no. 3, pp. 198–209, 2022, doi: 10.15276/hait.05.2022.15.

[4] W. Zhang, Z. Zhang, C. Wu, and A. Goh, "Assessment of basal heave stability for braced excavations in anisotropic clay using extreme gradient boosting and random forest regression," Underground Space, vol. 1, pp. 1–12, 2020, doi: 10.1016/j.undsp.2020.03.001.

[5] S. P. Timoshenko and S. Woinowsky-Krieger, Theory of plates and shells, 2nd ed. New York: McGraw-Hill, 1989, 580 p. (McGraw-Hill International Series).

[6] R. Onatskiy and S. Misiura, "Development of a methodology for rapid strength assessment of dented vessels using AI on the example of dataset formation for plates under pressure," Bulletin of the National Technical University "KhPI". Series: Dynamics and Strength of Machines, no. 1, pp. 60–68, 2025. doi: 10.20998/2078-9130.2025.1.331577.

[7] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.

[8] M. K. Thompson and J. M. Thompson, ANSYS Mechanical APDL for Finite Element Analysis, 1st ed. Oxford: Butterworth-Heinemann, 2017. doi: 10.1016/B978-0-12-812981-4.00001-0.