Developement of Different Machine Learning Models for Local Contact Search

Ankit *
Department of Mechanical Engineering
Indian Institute of Technology Guwahati
Guwahati, Assam, India.
ankit.meiitg@iitg.ac.in

Dipjyoti Nath
Department of Mechanical Engineering
Indian Institute of Technology Guwahati
Guwahati, Assam, India.
dnath@iitg.ac.in

Roger A. Sauer
Ruhr University Bochum
Bochum, Germany.
roger.sauer@rub.de

Sachin Singh Gautam
Department of Mechanical Engineering
Indian Institute of Technology Guwahati
Guwahati, Assam, India.
ssg@iitg.ac.in

Abstract

Recently, improvements in computational capabilities have triggered a surge in the use of machine learning (ML) models for various applications. One notable application of these advancements is the optimization of computational processes in contact mechanics. This involves a computationally demanding step known as contact search, which comprises global and local contact search. Global contact search identifies potential contact points between bodies, while local search pinpoints the exact pairs of points in contact [3].Oishi and Yoshimura [2] developed an ANN model that accelerated the local contact search.

In the present work, five distinct ML models—multiple linear regression, polynomial regression, deep neural network, decision tree, random forest, and extreme gradient boosting (XGBoost) are chosen based on their unique capabilities. These models are employed to predict local parametric coordinates (ξ, η) . The models were provided with input data consisting of variations in nodal coordinates for a 4 noded quadrilateral, along with corresponding slave points. Our analysis revolves around assessing these models in terms of computational efforts and efficiency as they are employed to predict the local parametric coordinates. This comparative evaluation sheds light on the respective strengths and weaknesses of each model, offering valuable insights for optimizing computational efficiency in contact mechanics applications.

1 Machine Learning Models

In the realm of ML, various models and algorithms offer a wide array of tools to tackle diverse data-driven challenges. For our present study, suitable models are selected based on their range in complexity and ability to exhibit unique abilities to fit the dataset. Among the models under consideration, multiple linear regression is considered for understanding the relationships between multiple predictors and a target variable, polynomial regression to capture nonlinear patterns, and DNN for their capacity to learn intricate patterns [1]. Decision Trees is taken for their interpretability, random forests for ensemble learning, and XGBoost for gradient-boosted predictions [1]. Each model is carefully selected to address specific characteristics of the dataset and contribute to the comprehensive understanding of our study's domain.

*Corresponding author: ankit.meiitg@iitg.ac.in

1st Conference on Applied AI and Scientific Machine Learning (CASML 2024).

2 Data Generation

In this approach, a 4-noded quadrilateral element with initially 15 degrees of freedom (12 from the element and 3 from the slave point) is simplified to 7 degrees of freedom. This is done as follows. Two nodes are fixed at $(0, 0, 0)$ and $(1, 0, 0)$, reducing dimensionality and complexity by concentrating on variations in two dimensions. Third node is varied within the region bounded by $x = 0$, $x = 2$, $y = 0$, and $y = 1$, while fourth node varies within $x = -1$, $x = 1$, $y = 0$, and $y = 1$. Geometric checks ensure segments are between 0.1 and 1 in length and angles between 10° and 170°. Five thousand valid quadrilateral elements are generated for machine learning. Then, for each quadrilateral, ten thousand slave points are created within a 3D box bounded by $x = \pm 1.5$, $y = \pm 1.5$, and $z = \pm 3$. Thus, the problem involves 7 degrees of freedom consisting of nodal coordinates and slave point positions.

3 Results and Discussions

3.1 Hyperparameter Tuning

Hyperparameter tuning [4] is a crucial aspect of optimizing ML models, involving the systematic search for the most effective configuration of parameters that are not learned during training. These parameters, known as *hyperparameters*, play a pivotal role in determining a model's performance and generalization ability [4]. In the present study, Random search is employed as the primary hyperparameter tuning technique across various models. Table 1 discusses the various hyperparameters used in different models, their importance, and the best selected parameters for the models.

Table 1: Hyperparameter selection for different ML models

ML model	**Hyperparameters**	**Optimum hyperparamter**
Polynomial regression	Polynomial degree	2
DNN	Hidden layers	3
	Neurons per layer	24
	Activation function	Sigmoid
	Learning rate	0.01
	Batch size	32
Decision tree	Maximum depth	None
	Minimum samples split	5
	Minimum samples leaf	2
	Maximum feature	Automatic
Random forest	Number of estimators	135
XGBoost	Number of estimators	170
	Learning rate	0.1
	Subsample	0.8
	Column Sample by Tree	0.7
	RT* for minimum child weight (γ)	0.2
	L1 weight regularization (α)	0.2
	L2 weight regularization (λ)	0.2

* : Regularization term

3.2 Error Assessment Across Diverse ML Models

A total of 2185256 data points were generated. A split ratio of 80 to 20 was used to divide it into training and test set. For the subsequent error analysis of the diverse machine learning models,

two key metrics, namely Mean Squared Error (MSE) and Coefficient of determination (R^2), are considered. The MSE, calculated as the average of the squared difference between the predicted and actual values, serves as a robust measure of a models accuracy

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{1}$$

where n is the number of data points, Y_i is the actual value, and $\hat{Y}_i$ is the predicted value. In the present study, the are ξ and η coordinates of the closest point projection in the target segment.

On the other hand, R^2, is employed to gauge how well the independent variables explain the variance in the dependent variable. The values of R^2 ranges from 0 to 1, where higher values indicate a better fit. The expression for R^2 is given by

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}{\sum_{i=1}^{n}(Y_i - \bar{Y})^2} \tag{2}$$

where n is the number of data points, Y_i is the actual value, $\hat{Y}_i$ is the predicted value, and $\bar{Y}$ is the mean of the actual values.

In Table 2, a comprehensive comparison of the training error for various ML models under consideration is presented. The training error analysis of the Ml models in the provided table reveals their performance in fitting the training data. Multiple linear regression shows a moderate fit with a MSE of 0.046972 and a R^2 of 0.853513, indicating it explains approximately 85.35% of the variance in the training data. Polynomial regression demonstrates a very good fit, with an MSE of 0.001184 and an R^2 of 0.996317, explaining about 99.63% of the variance. DNN exhibits an excellent fit, with an MSE of 0.000280 and an R^2 of 0.999124, explaining approximately 99.91% of the variance. Decision tree performs very well, with an MSE of 0.000527 and a R^2 of 0.998357, explaining about 99.84% of the variance. Random forest achieves exceptional performance with an MSE of 0.000121 and an R^2 of 0.999621, explaining approximately 99.96% of the variance. XGBoost, while still performing well, has a higher training error compared to other models, with an MSE of 0.003454 and an R^2 of 0.989229, explaining about 98.92% of the variance. These results indicate that Random forest and DNN are particularly adept at capturing the underlying patterns in the training data, suggesting their potential for accurate predictions on unseen data.

Similarly, Table 3 outlines the comparison of testing error for the same set of ML models. The analysis of testing errors for various ML models in this study provides valuable insights into their predictive performance. Among the models evaluated, Random forest emerges as the top performer, achieving the lowest MSE of 0.000837 and the highest R^2 value of 0.997391. This indicates Random forest's exceptional ability to capture complex patterns in the data and generalize well to unseen data. Following closely behind is the DNN model, which demonstrates strong performance with an MSE of 0.000321 and a R^2 of 0.998997, suggesting its effectiveness in capturing intricate data patterns and generalizing well to new data. Polynomial regression also performs well, with an MSE of 0.001765 and a R^2 of 0.994512, highlighting the importance of considering non-linear relationships in the data. The Decision Tree model, while not as accurate as Random forest or DNN, still performs reasonably well with an MSE of 0.003247 and anR^2 of 0.989880. XGBoost, while performing adequately, falls slightly short compared to the other models, with an MSE of 0.003559 and a R^2 of 0.988903. These results underscore the significance of selecting the most suitable model for a given dataset and research question, as different models excel in capturing different aspects of the data's complexity.

Table 2: Training error for different ML models

ML model	**MSE**	$\mathbf{R}^2$
MLR	0.046972	0.853513
Polynomial regression	0.001184	0.996317
DNN	0.000280	0.999124
Decision tree	0.000527	0.998357
Random forest	0.000121	0.999621
XGBoost	0.003454	0.989229

MSE : Means Squared Error; MLR: Multiple Linear Regression

Table 3: Testing error for different ML models

ML model	**MSE**	$\mathbf{R}^2$
MLR	0.046973	0.853511
Polynomial regression	0.001765	0.994512
DNN	0.000321	0.998997
Decision tree	0.003247	0.989880
Random forest	0.000837	0.997391
XGBoost	0.003559	0.988903

MSE : Means Squared Error; MLR: Multiple Linear Regression

4 Conclusion

The assessment of six regression models focuses on their accuracy as determined by error metrics, offering insight into their respective trade-offs. The choice of a regression model should be based on the specific accuracy requirements of a task. Models such as DNN may offer high accuracy but could be complex in nature. Decision tree, random forest, and XGBoost provide a balanced approach, delivering accurate results without sacrificing too much simplicity. Multiple Linear Regression, while generally simpler, may not achieve the same level of accuracy as more complex models. Understanding these trade-offs in accuracy is essential for selecting the most appropriate model for any given application.

Acknowledgement

The authors gratefully acknowledge the support from SERB, DST under the projects IMP/2019/000276, CRG/2022/002218 and VSSC, ISRO through MoU No.: ISRO:2020:MOU: NO:480.

References

[1] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, 1 edition, 2006.

[2] A. Oishi and S. Yoshimura. A new local contact search method using a multi-layer neural network. *CMES - Computer Modeling in Engineering and Sciences*, 21:93–103, 2007.

[3] P. Wriggers. *Computational Contact Mechanics*. Springer Berlin, Heidelberg, 2 edition, 2006.

[4] T. Yu and H. Zhu. Hyper-parameter optimization: A review of algorithms and applications, 2020.