A mechanical property prediction system for G-Lattices via machine learning

Article *in* Engineering Optimization · January 2024

DOI: 10.1080/0305215X.2023.2295353

5 authors, including:

Arash Armanfar
Istanbul Technical University
4 PUBLICATIONS **15** CITATIONS
SEE PROFILE

Alper Taşmektepligil
Yildiz Technical University
4 PUBLICATIONS **10** CITATIONS
SEE PROFILE

Ersan Ustundag
Istanbul Technical University
101 PUBLICATIONS **2,011** CITATIONS
SEE PROFILE

Ismail Lazoglu
Koc University
253 PUBLICATIONS **5,202** CITATIONS
SEE PROFILE

A Mechanical Property Prediction System for G-Lattices via Machine Learning

Arash Armanfar[*a], A. Alper Tasmektepligil[b,a], Ersan Ustundag[a], Ismail Lazoglu[c], Erkan Gunpinar[a]

[a]*Istanbul Technical University*
[b]*Yildiz Technical University*
[c]*Manufacturing and Automation Research Center Koc University Sariyer Istanbul 34450 Turkey*

Abstract

G-Lattices—a novel family of periodic lattice structures introduced by Armanfar and Gunpinar—demonstrate diverse mechanical properties owing to their generatively designed shapes. To effectively assess the properties of lattice structures, experimental tests and finite element analysis (FEA) are commonly used. However, the complex nature of these structures poses challenges, leading to high computation time and costs. This study proposes a machine learning (ML) approach to quickly predict the mechanical properties of G-Lattices under defined loading conditions. G-Lattice training data is generated through a sampling technique, and voxelized data is employed as ML feature vectors for predicting properties determined by FEA. To address the uneven distribution of target values, samples are clustered and utilized to train a classification model. This two-step process involves the classification of G-Lattices, followed by the application of specific regression models trained for each cluster for precise predictions. According to experiments, the obtained ML model, which predicts stiffness-over-volume ratios for G-Lattices, achieved a mean absolute percentage error of 6.5% for 1600 G-Lattices in a few seconds. Furthermore, approximately 70% of the 40000 G-Lattices exhibited errors within 5%. The ML model's rapid predictions and acceptable accuracy make it useful for quick decision-making and seamless integration into optimization processes.

Keywords: Lattice structure, G-Lattices, computer-aided design, mechanical property, machine learning.

1. Introduction

Lattice structures are popular in additive manufacturing (AM) due to their superior properties such as reduction of manufacturing time/material consumption and improvements in the acoustic, dielectric, and mechanical properties of the printed components [1]. A unit cell is recognized as the fundamental building block of a lattice structure, which reflects the properties of the overall structure. Extensive research has demonstrated that the mechanical characteristics of a unit cell are significantly influenced by struts and their connection topology [2, 3].

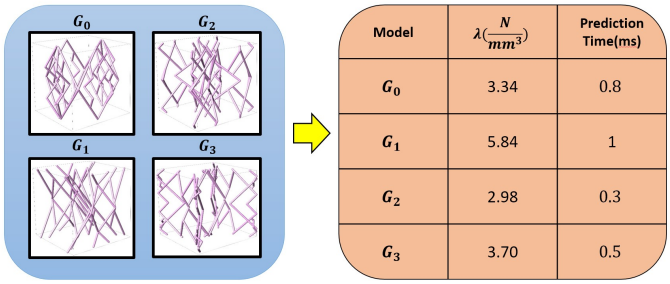
Identifying the mechanical properties of lattice structures traditionally relies on finite element analysis (FEA) and experimental tests, but the intricate geometry of lattice structures poses challenges, resulting in high computation time and costs [4]. This study proposes a method to construct a robust mathematical model capable of predicting the mechanical properties of a given lattice structure. The main challenge in the related previous studies arises from the limited availability of strut-based unit cell data [2], making it difficult to create the necessary dataset for training [5]. Consequently, our specific focus centers on G-Lattices [2], a novel family of lattice structures. The design of G-Lattices follows the generative design concept, allowing the rapid generation of numerous unique strut-based unit cells. As demonstrated in [2], altering the design parameters of G-Lattices and its connection topology significantly influenced their mechanical properties. The main aim of this research is to provide designers/engineers a system that evaluates the mechanical properties of G-Lattices. A designer can benefit from this mathematical model in two aspects:

1. **Quick determination of mechanical properties:** The model enables the rapid determination of the mechanical properties for a G-Lattice without the need for time-consuming and costly FEA tests. This efficiency is crucial in fast-paced design environments where timely decision-making is paramount.
2. **Integration into optimization processes:** The mathematical model is designed to seamlessly integrate into optimization processes, enabling designers to explore and identify optimal G-Lattices efficiently. This accelerates the overall design cycle by providing quick insights into the performance characteristics of different lattice configurations.

Fig. 1 illustrates usage scenario for the present work. Given four G-Lattices, $G_0, \ldots, G_3$, stiffness-over-volume ratios are predicted via machine learning in a few milliseconds.

For this purpose, a uniform sampling approach is first employed to evenly sample G-Lattices in the design space. Furthermore, a voxelization method is employed to characterize G-Lattices for forming required train data. Consequently, the resulting dataset includes voxel information and a numerical mechanical property for a set of G-lattices with distinct geometries. As stated in [6], values of the predicted engineering properties for the sampled G-Lattices could be agglomerated within certain ranges. This situation is also observed when G-Lattices are sampled. Therefore, the number of G-Lattices with values lying in these ranges is increased using a further sampling, which we call *balanced sampling*. In this work, the dataset is first divided into

[*]**Email:** armanfar@itu.edu.tr **Adress:** Inonu Caddesi, No:65, Gumussuyu, 34437, Istanbul, Turkey **Tel:** +90-212-2931300 **Fax:** +90-212-2450795



Model	$\lambda(\frac{N}{mm^3})$	Prediction Time(ms)
G_0	3.34	0.8
G_1	5.84	1
G_2	2.98	0.3
G_3	3.70	0.5

Figure 1: Stiffness-over-volume ratios for the given G-Lattices can be predicted via machine learning in a few milliseconds.

subsets based on these ranges. A classification model is then trained, which identifies the cluster of a given G-Lattice. Finally, separate regression machine-learning models are obtained for these clusters. Therefore, the proposed method involves a two-step prediction procedure dedicated to performance predictions of G-Lattices.

This article is organized as follows. In Section 2, we provide an overview of the relevant prior art. Section 3 includes a detailed description of G-Lattices and their generation procedure. The proposed approach for establishing machine learning models to predict the mechanical properties of a given G-Lattice is explained in Section 4. Section 5 discusses this study's experiments. Finally, in Section 6, we conclude.

2. Prior work

Due to their simplicity, versatility, and broad applicability, the design and optimization of manual or strut-based lattice structures have become popular research topics in recent years. In [7], the authors provided a comprehensive review of various lattice structures, their design principles and the additive manufacturing techniques used to fabricate them. The mechanical properties of lattice structures depends mainly on the shape parameters of the struts and their connection topology[1, 8]. Various studies have demonstrated that aligning struts in a lattice structure with its loading directions can significantly improve its stiffness [9, 10, 2]. Hence, the use of topology optimization to design lattice structures has been proposed in several works [11, 12].

Identifying the mechanical properties of lattice structures is time-consuming. One way is to physically test the structure for measuring its mechanical behavior [13, 14]. In [15], a range of mechanical properties of conventional unit cells, such as body-centered cubic (BCC) and face-centered cubic (FCC), was evaluated through compression experiments. However, these tests required significant resources and time. Finite element analysis (FEA) is another widely used method to investigate the mechanical performance of lattice structures [16]. One of the main challenges to conducting FEA for these structures stems from their complex geometry[17]. More precisely, the connection details between the struts can significantly affect the overall behavior of the structure. Additionally, performing FEA simulations for a model may take days, which also requires significant computational resources and expertise in modeling and simulation.

Machine learning techniques can be promising in predicting the mechanical behavior of lattice structures as they can perform quick predictions without the need for computational resources and engineering expertise. However, the research community has paid relatively little attention to this topic.

ML techniques were employed by Kulagin et al. [18] to establish connections between the elasticity properties of 2D lattice archimats and their patterns. Hassanin et al.[19] conducted a study to predict the mechanical properties of additively manufactured lattice structures with three strut-defining parameters using deep learning approaches. They employed various machine learning approaches such as shallow neural networks, deep neural networks, and deep learning neural networks, which lead to mean percentage errors of 5.26%, 14.60%, and 9.39% for ultimate strength, elastic modulus, and specific strength, respectively. Authors in [5] conducted a study on the mechanical properties of lattice structures using the support vector regression (SVR) method. A total of 57 samples were utilized as test/train data, and their feature vectors were defined with voxelization principles. The study reported a mean error of 24.61% and 40.9% for elastic modulus and yield strength, respectively.

Our work is positioned in this new line of research, improving the state of the art by introducing a set of machine learning techniques (clustering and linear regression) to predict the mechanical behavior of newly introduced lattice structures—G-Lattices. Instead of randomly sampling G-Lattices in the design space, a balanced sampling technique [20] subsequent to a uniform sampling approach [6, 21] is proposed.

3. G-Lattices

G-Lattices [2] is a novel family of periodic and strut-based lattice structures. Armanfar and Gunpinar demonstrated that G-Lattices had better mechanical properties under compression compared to traditional lattice structures such as Face and Body-Centered Cubic with Z-truss (FBCCZ). A particle tracing (PT) algorithm (as detailed in [2]) can be utilized for the generation of G-Lattices. Inserted particles in a sub-cell (1/8 of a unit cell) are moved along a straight line or a curve, while the present work focuses on G-Lattices consisting of only straight struts. Sweeping a circle along their trajectories then forms a set of struts as depicted in Figure 2 (a). After mirroring the struts in a sub-cell, a G-Lattice is obtained (See Figure 2 (b)). The G-Lattice can then be duplicated in three orthogonal directions for lattice embedding. Due to the generative (randomized) nature of the PT algorithm, it is possible to synthesize distinct G-Lattice geometries. Note here that the processing time to generate a G-Lattice is less than ten milliseconds. Furthermore, the algorithm optionally takes user-defined constraints (such as the overhang constraint of additive manufacturing) into account. Therefore, the obtained models satisfy all these constraints.

4. Proposed system

Given a set of design and manufacturing constraints, a mathematical model (ML model) is developed in this work via machine learning to predict mechanical properties for a G-Lattice. Fig. 3

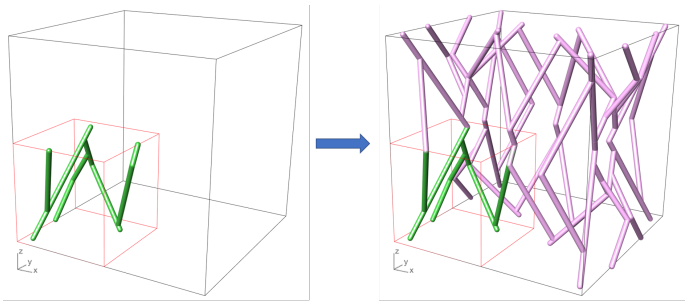


Figure 2: (a) Struts are generated in a sub-cell (1/8 of a unit cell). (b) A G-Lattice is formed by mirroring struts in a sub-cell.

outlines the pipeline of the proposed method. G-Lattices are uniformly sampled (by G-Lattice generator) to increase model distinctness. Mechanical properties of each G-Lattice are then validated using finite element analysis (FEA). G-Lattices and their mechanical performance are stored in a database and utilized as datasets for computing an ML model. G-Lattice generator samples G-Lattices until an ML model with desired quality is obtained.

This section involves three sub-sections. A method is outlined in Section 4.1 to collect uniform samples of G-Lattice. The FEA procedure used is explained in Section 4.2. Finally, the steps for establishing a mathematical model via machine learning are described in Section 4.3.

4.1. Generative sampling of G-Lattices

A computer-aided design (CAD) model is represented using *design parameters*. A design space is an n-dimensional space, where each design parameter represents a dimension in the design space. The design space is formed by design parameters and their value ranges (i.e., lower and upper bounds). A G-Lattice model is considered as a design in this work. Let $\{x_i \,|\, i = 0, 1, \ldots, N-1\}$ be design parameters for a G-Lattice. Setting values for these parameters yields a G-Lattice with unique geometry. Additionally, $\{l_i\}$ and $\{u_i\}$ denote lower and upper limits for these design parameters, resp.

4.1.1. Feature vector design for a G-Lattice

The present work aims at computing a machine learning (ML) model for estimating the mechanical properties of G-Lattices. Therefore, a feature vector, $\mathcal{F} = \{x_i \,|\, i = 0, 1, \ldots, N-1\}$, should be designed to encode a G-Lattice and can involve design parameters, $\{x_i \,|\, i = 0, 1, \ldots, N-1\}$. First of all, only struts in a sub-cell (instead of those in a unit cell) can be taken into account while designing $\mathcal{F}$. Recall that a G-Lattice is obtained by mirroring struts in a sub-cell. Particle tracing parameters (such as strut endpoints) can be utilized as design parameters representing $\mathcal{F}$. However, different design parameter sets can generate the same G-Lattice. In other words, a design parameter in $\mathcal{F}$ should represent the same feature of a G-Lattice. There must be one-to-one correspondence between the design parameters of two different G-Lattices. Fig. 4 illustrates this, in which the same G-Lattice is obtained via two different tracing patterns. Here, S_i denotes a strut generated in the i^{th} particle tracing step.

A sub-cell, $\mathcal{D} = \{v_i \,|\, i = 0, 1, \ldots, N-1\}$, is discretized using (same size) $M \times M \times M$ voxels, $\{v_i\}$, to represent $\mathcal{F}$, where $N = M^3$. Instead of utilizing struts for such discretization, center lines of struts are considered to discretize $\mathcal{D}$. Recall that G-Lattices consisting of struts with constant radii are taken into account in this work. Therefore, if center lines of struts (in a sub-cell, $\mathcal{D}$) pass through v_i, x_i is 1. Otherwise, it is set to 0. Fig. 5 illustrates voxels in $\mathcal{D}$ for a G-Lattice. Here, cubes with bold edges within the sub-cell represent solid voxels, while the remaining depict the void voxels. Finally, all $l_i = 0$ and $u_i = 1$ in the proposed $\mathcal{F}$.

4.1.2. G-Lattice sampling technique

Each execution of the PT algorithm in [2] generates a new G-Lattice model. And multiple executions of the algorithm produce multiple, potentially distinct G-Lattice models. The obtained G-Lattice models will be utilized to train the ML model. However, depending on G-Lattice models existing in the dataset, prediction accuracy of the obtained ML model may change. Therefore, instead of randomly generating G-Lattice models, a uniform sampling approach [6] can be employed.

To avoid similar G-Lattice models, a similarity metric, which is based on Audze and Eglais potential energy [21, 22] is exploited. They used the analogy of minimizing forces between charged particles. Particles are in equilibrium when the potential energy is at the minimum. Let $\{G_j, j = 0, \ldots, R-1\}$ be G-Lattices, where R is the number of the samples, that have been already generated and exist in the dataset. G denotes a G-Lattice. Potential energy, E, between G and $\{G_j\}$ is computed as follows:

$$E = \sum_{j=0}^{R-1} \frac{1}{d^2(G, G^j)}, \tag{1}$$

where

$$d(G, G^j) = \sqrt{\sum_{i=0}^{N-1} (x_i - {x_i}^j)^2}. \tag{2}$$

Here, x_i and ${x_i}^j$ denote i^{th} voxel values for G-Lattice models, G and G^j. E favors designs (G-Lattices) to be located as far as possible based on the Euclidean distance metric. The smaller the potential energy values (E), G can be considered more distinct to G-Lattices, $\{G_j\}$.

Given (1) a set of design and manufacturing constraints, (2) already sampled G-Lattices, $\{G_j, j = 0, 1, \ldots, R-1\}$, (3) number, Q, of randomization for sampling a G-Lattice and (4) edge size for voxelization, Algorithm 1 generates a G-Lattice by taking the energy, E, in Equation 1 into account. Recall that the function ComputeEnergy (Line 4) takes previously sampled G-Lattices, $\{G_j\}$, and edge size, e, for voxelization as inputs. After $Q-$times randomly generating G-Lattices (Line 3), the one with minimum value for E is stored in the data base. In case there are no sampled G-Lattices (i.e., $N = 0$ or initial sampling step), a G-Lattice is randomly generated in the design space.

4.2. Finite element analysis of G-Lattices

G-Lattices sampled using Algorithm 1 are evaluated via FEA to determine their mechanical properties. It is assumed that a G-Lattice should replace the inner solid region of a box with an edge

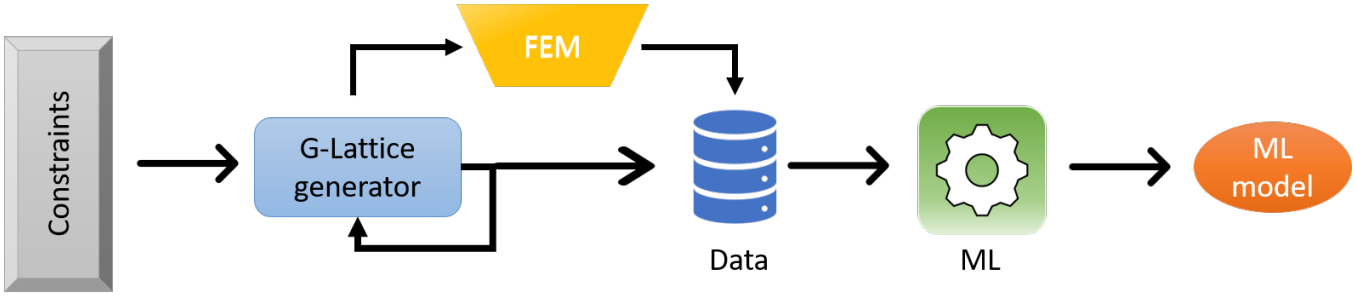


Figure 3: Workflow of the proposed method to develop a machine learning model for predicting mechanical properties of G-Lattices.

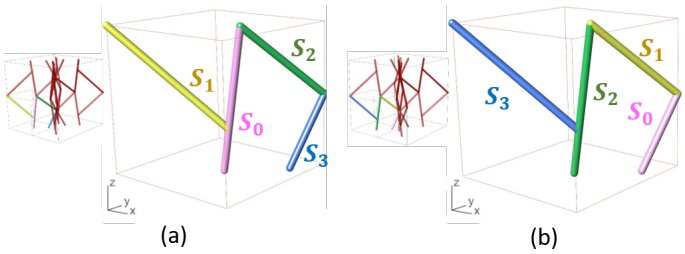


Figure 4: Two different tracing approaches ((a) and (b)) can result in the same G-Lattice model. S_i denotes a strut obtained in the i^{th} particle tracing step.

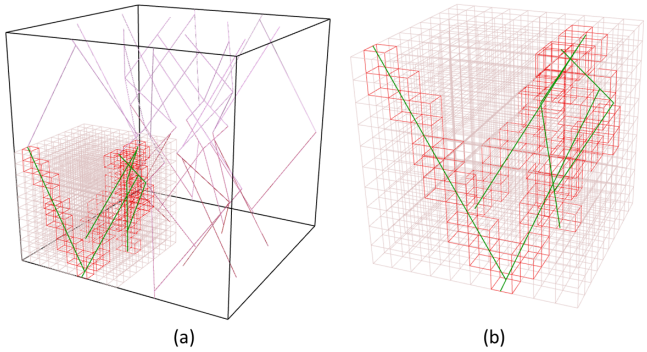


Figure 5: Discretization of a G-Lattice. (a) The center lines of struts are shown in a unit cell. Only center lines in a sub-cell are utilized to discretize (with voxels) a G-Lattice. In case center lines pass through voxels (with bold edges), voxel values are set to 1 (solid). Otherwise, they are zero. (b) A zoomed view $\mathcal{D}$ for the G-Lattice in (a).

equal to that of a unit cell. Note that the wall thickness of the box should be chosen to be small in order not to overwhelm G-Lattice properties. These walls support the struts of a G-Lattice with free ends. Images 1 and 2 in Figure 6 (a) illustrate a G-Lattice embedded box from various viewpoints. The commercial package ABAQUS is used for FEA. The mechanical properties of interest, such as the stiffness-over-volume ratio, maximum stress, energy absorption, and others, are then extracted from the FEA results.

In this study, the behavior of G-Lattices under compression loading is evaluated. The ABAQUS simulation platform employs the Galerkin method for the weak-form presentation in the FEA evaluations. The initial step involves the generation of various cell types (each representing a unit cell model) utilizing the methodology outlined in Section 3. It is noteworthy that any G-Lattice model can be adeptly utilized in the creation of periodic lattice structures owing to the geometric constraints established during the G-Lattice modeling process. Subsequently, a G-Lattice embedded box is tied to two rigid plates positioned on its upper and lower surfaces, as depicted in Figure 6 (b). The implementation of a tie constraint yields more accurate results compared to experimental tests [15]. The lower plate is immovably fixed in all directions, whereas the upper plate is restricted to motion solely in orthogonal directions and prevented from rotations. The loading direction is represented by a displacement vector, $\vec{U} = (U_x, U_y, U_z)$, illustrated in Figure6 (b), and specified for the upper plate. The cumulative reaction force, denoted as RF_t, results from the motion of the upper plate, serving as an indicator of the stiffness inherent in the loaded model. This experimental setup with defined constraints effectively emulates real-world compression tests, offering a more realistic representation of the structural behavior of G-Lattices under compression loading conditions. RF_t can be defined as the summation of reaction forces in X, Y, and Z-directions (RF_x, RF_y and RF_z,

Algorithm 1 Sampling a G-Lattice, G.

Input:

1. $\{\tau_k\}$, design and manufacturing constraints;
2. $\{G_j, j = 0, 1, \ldots, R-1\}$, G-Lattices stored in data base;
3. Q, number of randomization for sampling a G-Lattice, G;
4. e, edge size for voxelization;

Output:

1. G, a G-Lattice;

Algorithm:

$E_{min} \leftarrow 9E^9$;
for $i = 0$ to $Q - 1$ **do**
  $G' :=$ GenGLattice($\{\tau_k\}$);
  $E_j :=$ ComputeEnergy($\{G_j\}, G', e$);
  if $E_j < E_{min}$ **then**
    $E_{min} \leftarrow E_j$;
    $G \leftarrow G'$;
  end if
end for
Return G;

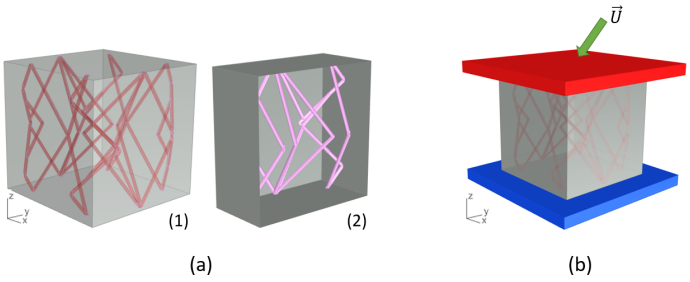


Figure 6: (a) The solid inner region of a box is replaced by a G-Lattice (1) with its cut view (2). (b) The model is tied to two rigid plates for finite element analysis: the lower one is fixed and the upper one is limited to move in the orthogonal (X, Y and Z) directions. $\vec{U}$ denotes the displacement vector.

respectively). $RF_t = \sqrt{RF_x^2 + RF_y^2 + RF_z^2}$. Hence, $\frac{RF_t}{V}$ yields stiffness-over-volume ratio for G.

4.3. Mechanical property estimation of G-Lattices

Sampled G-Lattices should be provided in a format that can be interpreted by machine learning algorithms. The dataset, Γ, consists of a set of feature vectors, $\{\mathcal{F}_j\}$ for G-Lattices, $\{G_j\}$, and their output parameters, $\{\overline{Y}_j\}$. $\Gamma = \{(\mathcal{F}_j, \overline{Y}_j), j = 0, 1, \ldots, R-1\}$. Recall that $\mathcal{F} = \{x_i \,|\, i = 0, 1, \ldots, N-1\}$ includes the voxel information, and R denotes the number of sampled G-Lattices. $\mathcal{F}$ includes binary values representing whether the material exists in cells or not, which are used to train the ML model. As the nature of the binary form of the data is already normalized, any further normalization or standardization processes have become unnecessary in the dataset preparation step.

The number of variables ($N = M^3$) in the feature vector directly affects the required number of G-Lattices in Γ to have a ML model with high prediction accuracy. Therefore, various datasets with different M values should be formed and investigated to optimize ML model performance.

G-Lattice sampling process in Algorithm 1 generates G-Lattices that are (as much as possible) evenly distributed in the design space (based on the feature vector). However, we may still have a challenge in that there can exist an uneven distribution of $\overline{Y}$ values among the sampled G-Lattices. This issue has been already addressed in [6], which can pose challenges for machine learning models. More precisely, $\overline{Y}$ values of the G-Lattices can potentially be agglomerated within a certain range. If required, the number of G-Lattices in the poor ranges, where a few G-Lattices are sampled, can be increased. Given a dataset $\Gamma = \{(\mathcal{F}_j, \overline{Y}_j), j = 0, 1, \ldots, R-1\}$, a supervised clustering algorithm is then applied to group the sampled G-Lattices based on $\overline{Y}_j$ into n clusters. Let C_j denote the cluster label for G_j. We then compute a (supervised classification model) $\mathcal{M}^C$, which takes $\mathcal{F}_j$ of a given G-Lattice as input and predicts its cluster label. For each cluster, a linear regression model, $\mathcal{M}_k^R$, is obtained ($k = 0, \ldots, n-1$) to compute Y value for given a G-Lattice, G, that is in the k^{th} cluster. Fig. 7 illustrates the steps to computing Y of G.

To prevent any overfitting occurring within the model, all preliminary and following ML models were trained using test, train, and validation sets. K-fold cross-validation was utilized for the validation sets.

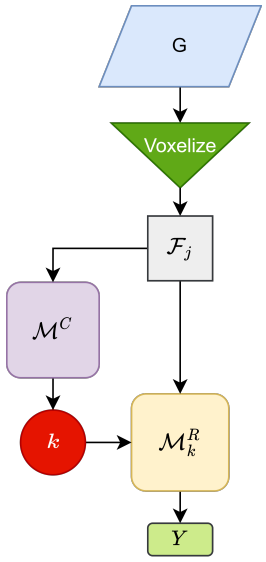


Figure 7: Workflow to calculate Y for a given a G-Lattice, G, with a feature vector, $\mathcal{F}_j$. Cluster label, k, for G is found using the classification model, $\mathcal{M}^C$. k is then utilized in the regression model, $\mathcal{M}_k^R$, to compute Y.

To investigate the accuracy of the classification model, $\mathcal{M}^C$, the model's accuracy was assessed using accuracy, precision, recall, F1-score, and Roc-AUC metrics. Furthermore, to validate the performance of regression models, mean absolute (relative) percentage error (Ψ), mean absolute error (MAE) and root mean squared error (RMSE) were also utilized. Note that using various metrics for evaluating ML models can give a better insight into their performance. Equations 3 to 10 contain formulations for these metrics. Here, TP, TN, FP, FN stand for the number of true positive, true negative, false positive, and false negative predictions of the classification model. For regression models, Y_j and $\overline{Y}_j$ denote the predicted and actual values for the corresponding mechanical properties, respectively.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5}$$

$$\text{F1 - score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{6}$$

$$\text{AUC} = \int_{-\infty}^{\infty} Y(t)dx(t) \tag{7}$$

$$\Psi = \frac{100}{R} \sum_{j=0}^{R-1} \left| \frac{Y_j - \overline{Y}_j}{\overline{Y}_j} \right| \tag{8}$$

$$\text{MAE} = \frac{1}{n} \sum_{j=1}^{n} \left| Y_j - \overline{Y}_j \right| \tag{9}$$

$$\text{RMSE} = \sqrt{\sum_{i=1}^{n} \frac{\left(Y_j - \overline{Y}_j\right)^2}{n}} \tag{10}$$

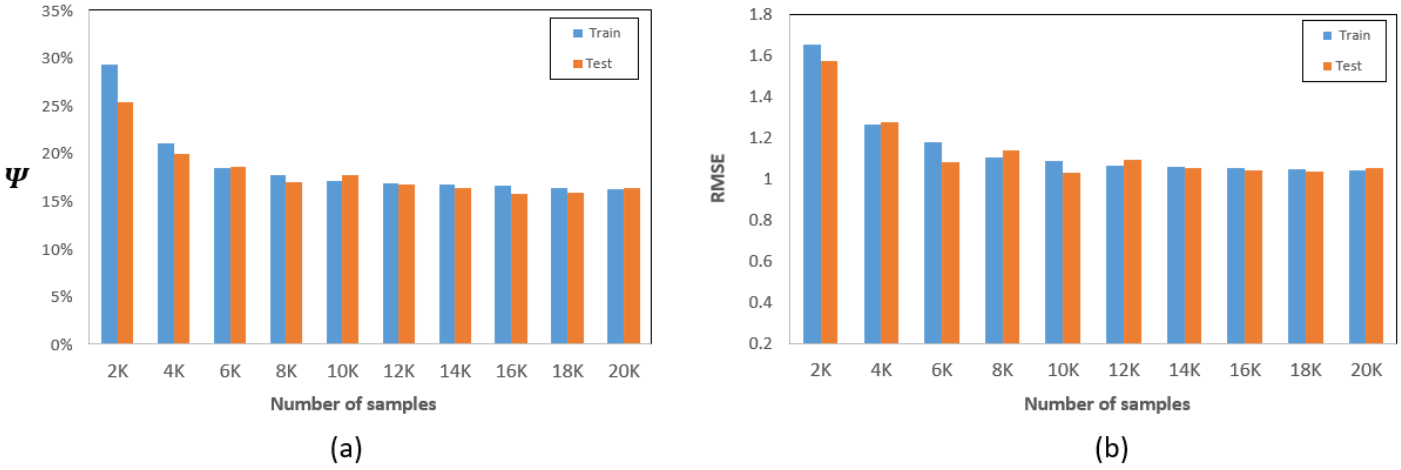


Figure 8: Effect of the number of samples on the (a) mean absolute percentage error (Ψ) and (b) root main square error (RMSE) of the resulting models obtained using 2000-20000 G-Lattices. 2K represents 2000.

5. Results and discussion

Experimental values are first described here before demonstrating the efficiency of the proposed technique. The parameter, Q, in G-Lattice sampling (Section 4.1.2) was set to a value of 100. The edge size of a sub-cell and strut diameters were set to 1 and 0.03 mm, respectively. To address the issue of merging struts in a G-Lattice, spheres with diameters of 0.035 mm were placed at the strut connection points. Furthermore, a 45° overhang constraint was included in the G-Lattice generation method. Note that all the G-Lattices used in the dataset were generated using the mentioned parameters through the algorithm proposed by authors in [2].

The mechanical property of a G-Lattice to be predicted ($\overline{Y}$) was chosen as the stiffness-over-volume ratio. As mentioned before, commercial package ABAQUS was used to extract this property for modeled G-Lattices. The G-Lattices were assumed to be under a compression loading with $\vec{U} = (0, 0, -10^{-4})$ mm. A small displacement for $\vec{U}$ was considered to ensure the G-Lattices remaining in the elastic region. For the FEA simulations, the wall thickness of the outer box was set to a small value 0.0005 mm, to better investigate the mechanical properties of the G-Lattices. The Inconel 625 alloy, widely used in the fabrication of lattice structures, was chosen as the material (assumed to be isotropic). The following values for density, Young's modulus, and Poisson's ratio were, respectively, chosen in the simulations: $\rho = 8440$ kg/m^3, $E = 207.5$ GPa, and $\nu = 0.304$ [23]. Additionally, the geometric models were meshed using quadratic tetrahedral elements with a maximum size of 0.04 mm [2].

All subsequent training procedures were carried out using the following parameters:

1. The split ratio for the test and train dataset was set to 0.1 (10% and 90%, respectively).
2. The K-fold cross-validation number of k was set to 10. The train set was used to create a cross-validation dataset.

Split ratios typically range from 10% to 30%. Since 10-fold cross-validation was implemented in the train set, the minimum value was chosen for the split ratio. This prevents any performance drop in the model due to the low number of samples in the training procedure.

The number of voxels, M, was initially configured to 10, and the linear regression method was chosen. Size of the dataset directly affects the trained model's quality [6, 20]. However, in order to determine the minimum necessary samples, the number of G-Lattices was systematically incremented by 2000. Fig. 8 depicts the impact of sample size on the model's performance. The convergence of Ψ (a) and RMSE (b) became apparent as the sample size progressively increased, reaching a noticeable stabilization point at 20000 samples.

The number of voxels, M. M was also tested to investigate its effect on the machine learning results. Increasing M results in a cubic increase for the feature vector elements as $N = M^3$. Enlarging N yields to the higher model resolutions. However, this not only deepens the problem of processing large amounts of data but also necessitates larger memory. In the experiments, M was adjusted to 5, 8, 10, and 16. Using 8000, 12000, 16000, and 20000 G-Lattices, linear regression models were obtained. Mean absolute (relative) percentage error, Ψ, values obtained for training and test data are shown in Figure 9. $M = 10$ was chosen as Ψ values were similar to those of $M = 5$ and $M = 8$, and $M = 10$ also produced finer representation in contrast. Compared to $M = 16$, Ψ value of $M = 10$ was lower. Furthermore, according to Figure 8 (a), Ψ decreased when the number of G-Lattices, R, increased in the set. Convergence of Ψ was observed at $R = 20000$, and therefore, G-Lattice sampling was terminated. The difference between Ψ values for the train and test data was only about 4 to 5 percent, which was negligible. Therefore, no overfitting has been observed in the regression models.

Alternative ML algorithms. In the initial stage of the analysis, six machine learning (ML) algorithms were utilized, linear regression, Lasso and Ridge methods, the boosting methods, XGBoost, Gradient Boosting trees, and LightGBM methods. The rationale behind the selection of these methods is rooted in their applicability to different data scenarios. The inclusion of the linear regression method and its derivatives (Lasso and Ridge) is

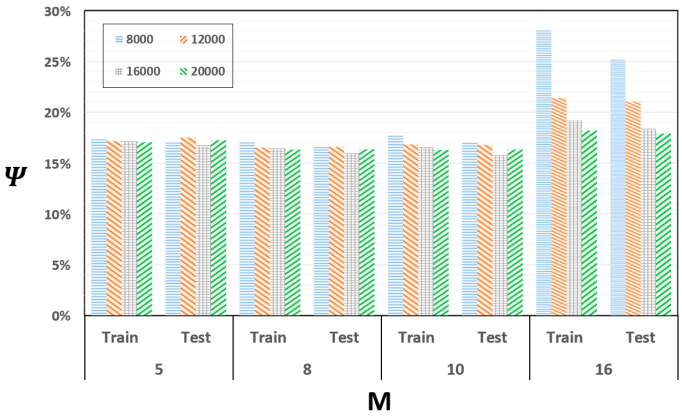


Figure 9: Ψ versus M for training and test data used in linear regression models obtained using 8000, 12000, 16000, and 20000 G-Lattices.

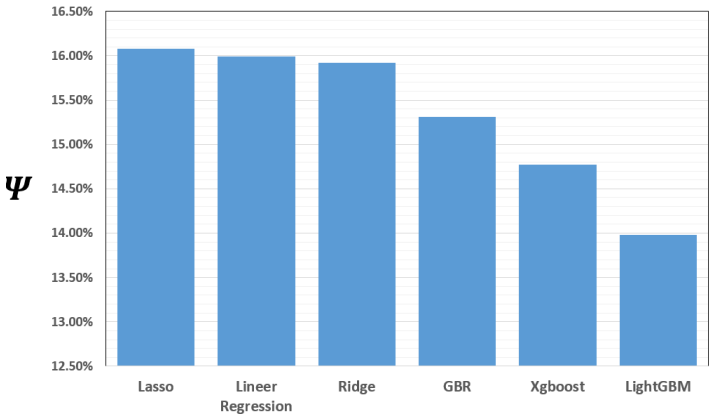


Figure 10: Comparison of various machine learning algorithms for 20000 samples.

motivated by their simplicity and effectiveness in capturing linear relationships within the data. These methods are particularly efficient at yielding reliable results in cases characterized by linearity. Conversely, the incorporation of tree-based models (Gradient Boosting, XGBoost, and LightGBM) was driven by their capacity to address more intricate scenarios involving non-linear relationships. However, the tree-based models are susceptible to overfitting due to the intricacy of their hyper-parameter tuning. Among these tree-based methods, XGBoost and LightGBM stand out as popular choices owing to their exceptional performance metrics [20]. Specifically, LightGBM is favored for its efficiency and rapid training speed, coupled with reduced memory usage—attributes that render it particularly advantageous when handling extensive datasets.

The performance results of each algorithm are illustrated in Figure 10. Table 1 provides the hyper-parameters utilized for these methods. To prevent models from getting stuck in local optima and to mitigate overfitting, it is essential to optimize hyper-parameters. There are several approaches available for achieving this goal [24, 25]. In this work, hyper-parameters were optimized using a grid Search function of the Python Scikit-learn library [1] and FLAML hyper-parameter optimization tool [2]. The model's performance can vary based on the subset of data used for training. To ensure consistent results across different datasets, K-fold cross-validation was commonly utilized [26, 27]. Thus, a 10-fold cross-validation was employed in all hyper-parameter optimization steps. Based on these results, XGBoost and LightGBM exhibited good and similar performances compared to others, and therefore, these two methods were chosen as the main machine learning methods and utilized in the experiments.

Clustering and multiple regression models. When the regression model obtained using 20000 G-Lattices was thoroughly analyzed, prediction errors for stiffness-over-volume ratios were high when $\overline{Y} >= 5$. This was probably due to the uneven distribution of $\overline{Y}$ values in various ranges as shown in Figure 11. Therefore, G-Lattices were clustered into $k = 2$ groups (G-Lattices with $\overline{Y} < 5$ and $\overline{Y} >= 5$) as described in Section 4.3. As the latter group consisted of a smaller number of G-Lattices, further G-Lattice sampling (Section 4.1.2) was carried out to sample 20000 G-Lattices. Among them, 10000 was obtained using Algorithm 1. While generating the next 10000 G-Lattices, only vertically aligned G-Lattices were generated using Algorithm 1. These G-Lattices had higher stiffness-over-volume ratios under compression loading and can be obtained using the G-Lattice generation method described by Armanfar and Gunpinar [2]. The distribution of $\overline{Y}$ values for 40000 G-Lattices is shown in Figure 11.

Table 1: Hyper-parameter values for the ML models.

ML method	Hyper-parameters and ranges
Linear regression	-
Lasso	alpha:[0.001 1.0] max_iter:[1000]
Ridge	alpha:[0.001 1.0] max_iter:[1000]
Gradient boosting trees	learning_rate:[0.01 0.3] n_eastimators:[50 500] max_depth:[4 10] subsamples:[0.5 1.0] max_features:[0.5 1.0]
LightGMB	learning_rate:[0.01 0.3] max_depth:[4 10] colsample_bytree:[0.5 1.0] num_leaves:[20 1000]
XGBoost	learning_rate:[0.01 0.3] n_estimators:[50 500] max_depth:[2 10] subsamples:[0.5 1.0] colsample_bytree:[0.5 1.0]

The classification model, $\mathcal{M}^C$, was then computed using LightGBM [3] and XGBoost methods for k-labeled G-Lattices. The classification results were promising as the values for accuracy are given in Table 2. According to the results, XGBoost performs slightly better performance thus it was selected as the classification algorithm.

Two regression models, $\mathcal{M}_1^R$ and $\mathcal{M}_2^R$, were obtained for each cluster. Table 3 depicts the results in different regression metrics for these models. The best Ψ values of the test data for $\mathcal{M}_1^R$ and

[1] https://scikit-learn.org/stable/
[2] https://scikit-learn.org/stable/
[3] https://lightgbm.readthedocs.io/en/v3.3.2/

Table 2: Evaluation of LightGBM and XGBoost classification models using different metrics.

Metric		ROC-AUC	Accuracy	Recall	Precision	F1-Score
LightGBM	**Train**	0.909	0.840	0.755	0.866	0.806
	Test	0.919	0.844	0.766	0.872	0.816
	10 Fold	0.911	0.837	0.759	0.864	0.808
XGBoost	**Train**	0.903	0.840	0.754	0.856	0.814
	Test	0.952	0.846	0.869	0.774	0.819
	10 Fold	0.916	0.841	0.767	0.866	0.813

Table 3: Prediction quality of two clusters for LightGBM and XGBoost.

ML models		$\mathcal{M}_1^R$			$\mathcal{M}_2^R$		
Metrics		ψ	**MAE**	**RMSE**	ψ	**MAE**	**RMSE**
LightGBM	**Train**	0.078	0.323	0.414	0.132	1.051	1.336
	Test	0.085	0.335	0.415	0.146	1.074	1.371
	10 Fold	0.086	0.342	0.421	0.149	1.092	1.388
XGBoost	**Train**	0.071	0.281	0.343	0.138	1.051	1.323
	Test	0.093	0.358	0.435	0.145	1.671	1.351
	10 Fold	0.094	0.371	0.454	0.141	1.212	1.314

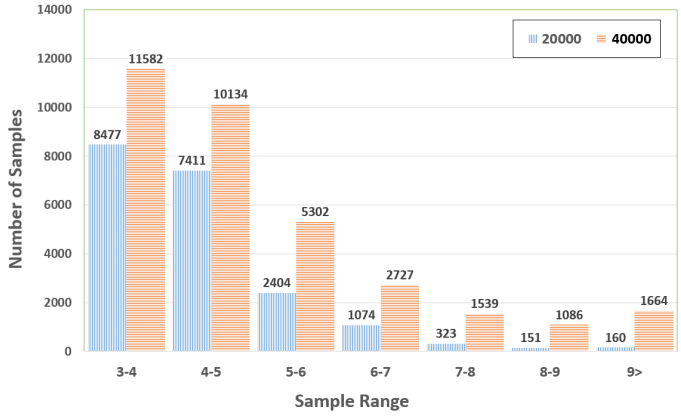


Figure 11: Distribution of $\overline{Y}$ values for 20000 and 40000 G-Lattices.

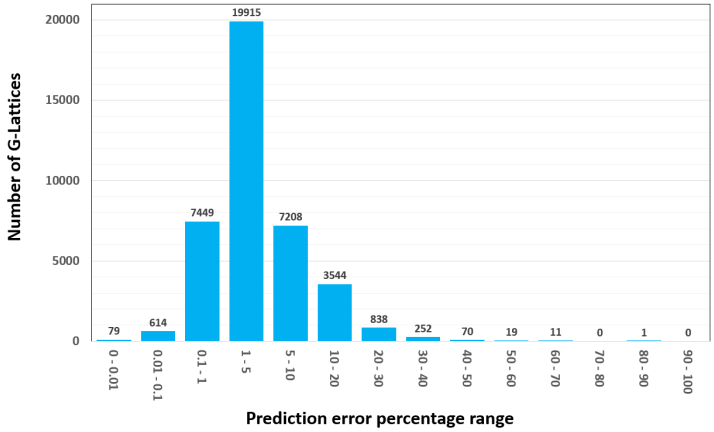


Figure 12: Number of G-Lattices versus mean absolute relative error percentage ranges for 40000 G-Lattices.

$\mathcal{M}_2^R$ were, respectively, 8.5% and 14.5%. Therefore, LightGBM and XGBoost algorithms were selected for training $\mathcal{M}_1^R$ and $\mathcal{M}_2^R$ respectively. The optimized values for selected hyper-parameters for the selected algorithms are given in Table 4. Note that, by increasing the number of data in the second group, it will potentially be possible to further decrease the Ψ value for the second regression model.

Error distribution (number of G-Lattices versus prediction error percentage) for 40000 G-Lattices can be seen in Figure 12. Approximately 70% of G-Lattices had errors less than/equal to 5%. Furthermore, about 97% of the G-Lattices had prediction errors smaller than 30%. Finally, given a G-Lattice, the prediction time for Y was less than one millisecond. Based on these results, it can be concluded that the proposed method can be utilized for accurate and fast prediction of $\overline{Y}$ values for G-Lattices. Fig. 13 illustrates actual ($\overline{Y}$) and predicted (Y) stiffness-over-volume ratios (with $\overline{Y} \geq 5$ and $\overline{Y} < 5$) for several G-Lattices.

Ψ was 16.3% when a single regression model for 20000 G-Lattices was employed. Therefore, the proposed approach (clustering and multiple regression) increased the prediction accuracy of the regression models. On the other hand, MAE and RMSE were 0.469 and 0.641 respectively. This can be explained by the excessive number of G-Lattices located in the first cluster. After the ML models (for both classification and regression) and their corresponding hyper-parameters were determined by the extensive tests, the final model was generated by using the whole 40000 of data.

For validating the prediction performance of the ML models, 1600 new G-Lattice models were further generated by randomization. Ψ was 6.5%, which means that the ML models make well predictions for such unseen data.

Case study. In order to address the applicability of the generated ML model, a case study was conducted. Lattice structures, distinguished by their constituent unit cells, have demonstrated numerous real-world applications, particularly in components subjected to compression [4]. Assume that a region in a part experiences pure compression, and the objective is to fill this region with strong G-Lattices (i.e., G-Lattices with higher Y-values). Assume the volume occupied by a single unit cell

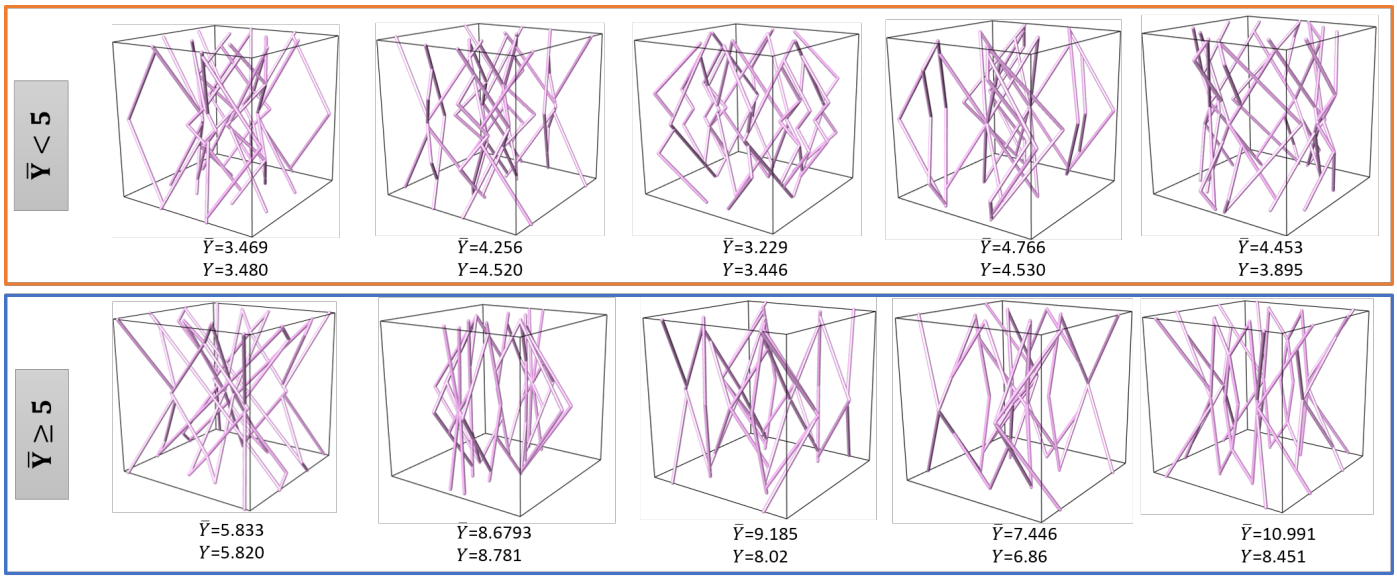


Figure 13: Actual ($\overline{Y}$) and predicted (Y) stiffness-over-volume ratios (with $\overline{Y} \geq 5$ and $\overline{Y} < 5$) for several G-Lattices.

Table 4: Optimized values for selected hyper-parameters for the chosen algorithms.

XGBoost ($\mathcal{M}_c$)	learning_rate	0.032
	n_estimators	480
	max_depth	3
	subsamples	0.881
	colsample_bytree	0.571
LighGMB ($\mathcal{M}_1^R$)	learning_rate	0.012
	num_leaves	253
	max_depth	7
	colsample_bytree	0.646
XGBoost ($\mathcal{M}_1^R$)	learning_rate	0.134
	n_estimators	139
	max_depth	5
	subsamples	0.774
	colsample_bytree	1

should be at maximum $0.015mm^3$. Additionally, the selected G-Lattice unit cell should demonstrate a minimum stiffness of $0.1N$ when subjected to a displacement of $0.0001mm$. The G-Lattice generator proposed in [2] was integrated with the ML (prediction) model developed in this work. The G-Lattice generator produced 1000 G-Lattice models with volumes below the mentioned request, and their mechanical performances (Y) were evaluated using the ML model. Among them, 10 models with the highest Y values considering stiffness constraint were selected. Table 5 presents the details of these selected G-Lattices, denoted as $G_1, G_2, \ldots, G_{10}$, including their occupied volumes, predicted (Y), and actual ($\overline{Y}$) values obtained via FEA tests. Fig. 14 shows three G-Lattices, G_2, G_7, and G_{10}, with the highest stiffness values $0.12N$, $0.14N$ and $0.13N$, respectively.

Generation of 1000 G-lattices took 95 seconds and predicting their Y values was completed in less than 5 seconds. On the other hand, conducting a FEA test for a G-Lattice was approximately 3 minutes. Therefore, the computational efficiency of the proposed methodology was considerably better than the conventional FEA tests. The evaluation metrics, ψ, MAE, and RMSE were, respectively, 12% , 1.1, and 1.2 for 10 G-Lattices. Our methodology offers significant advantages such as reducing computation time and costs. Instead of 1000 time-consuming FEA tests, we strategically identify and validate the top 10 models less than two minutes. The selected G-Lattice can then be used for further evaluations.

Implementation environment. The proposed algorithm was implemented using a PC with an Intel Core i5 6500 3.20 GHz processor and 16 GB memory, and the implementation was single-threaded. For generating the training data employed in ML, a self-operating system was developed using C# and Rhino -Grasshopper[4] libraries, which also utilized ABAQUS for mechanical property evaluations of G-Lattices. Furthermore, the machine learning algorithms were implemented in Python.

6. Conclusion and future works

This article proposes a set of machine learning algorithms composed of classification and linear regression ML methods to predict the mechanical properties of G-Lattices [2]. A two-step procedure was first used to sample various G-lattices: a uniform [21] and balanced [20] sampling. FEA was then conducted to investigate the mechanical performance of G-Lattices. Multiple regression models were then obtained for each of the G-Lattice clusters, which were determined based on the values of the predicted mechanical property. The ML model obtained performed predictions in milliseconds, and therefore, could be useful to explore/investigate many G-Lattices in a shorter time.

The accuracy of the obtained ML models can further be enhanced by increasing the number of voxels simplifying the G-Lattices. Furthermore, the number of G-Lattices with $\overline{Y} \geq 5$ can be increased to reduce the Ψ to lower values.

As future work, it is planned to generate ML models for G-Lattices under various loading conditions. Additionally, the ML

[4] https://www.rhino3d.com/6/new/grasshopper/

Table 5: Volume, actual and predicted stiffness-over-volume rations of the 10 G-Lattices obtained in the case study.

Selected G-Lattices	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8	G_9	G_{10}
Volume	9.5	10.5	11.5	9.8	10.2	10.1	15	11.5	9.2	13
Y	10.2	11.4	10.4	10.1	10.3	10.7	10.1	11.2	10.1	11.6
$\overline{Y}$	7.8	12.1	9.1	10.4	8.2	11.7	9.9	10.3	10.8	10.2

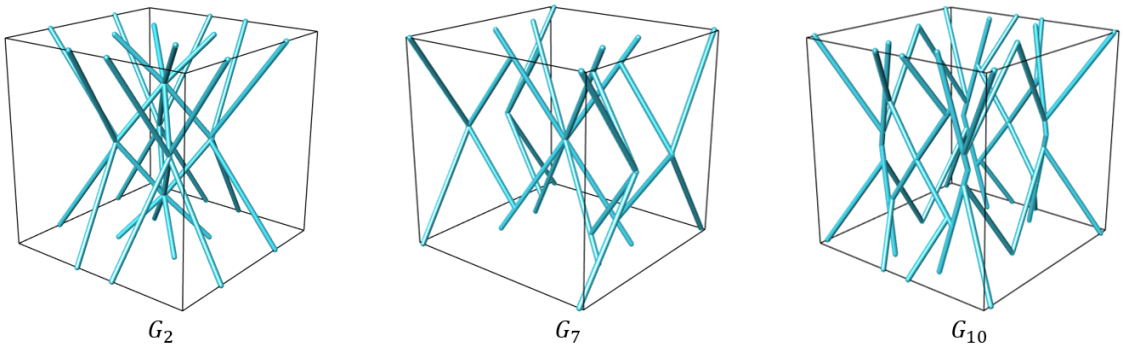


Figure 14: Three G-Lattices with the highest performance obtained in the case study.

models obtained will be integrated into G-Lattice optimization step in order to generate superior G-Lattices. Finally, deep neural networks [28, 29, 30] will also be utilized and compared with the model obtained in this work. [31] [32]

Data Availability Statement

The data that support the findings of this study are available upon request. Restrictions apply to the availability of these data, which were used under license for this study. Data are available from the corresponding author upon reasonable request and with permission from the owner.

Disclosure Statement

There are no conflicts of interest.

References

[1] A. Seharing, A. H. Azman, S. Abdullah, A review on integration of lightweight gradient lattice structures in additive manufacturing parts, Advances in Mechanical Engineering 12 (2020) 168781402091695.

[2] A. Armanfar, E. Gunpinar, G-Lattices: A Novel Lattice Structure and Its Generative Synthesis Under Additive Manufacturing Constraints, Journal of Mechanical Design 145 (1) (2022) 011702.

[3] T. Maconachie, M. Leary, B. Lozanovski, X. Zhang, M. Qian, O. Faruque, M. Brandt, SLM lattice structures: Properties, performance, applications and challenges, Materials Design 183 (2019) 108137.

[4] M. Smith, Z. Guan, W. Cantwell, Finite element modelling of the compressive response of lattice structures manufactured using the selective laser melting technique, International Journal of Mechanical Sciences 67 (2013) 28–41.

[5] S. Ma, Q. Tang, Y. Liu, Q. Feng, Prediction of Mechanical Properties of Three-Dimensional Printed Lattice Structures Through Machine Learning, Journal of Computing and Information Science in Engineering 22 (3) (12 2021).

[6] E. Gunpinar, U. C. Coskun, M. Ozsipahi, S. Gunpinar, A generative design and drag coefficient prediction system for sedan car side silhouettes based on computational fluid dynamics, Computer-Aided Design 111 (2019) 65–79.

[7] A. Nazir, K. Mekonen, A. Kumar, J.-Y. Jeng, A state-of-the-art review on types, design, optimization, and additive manufacturing of cellular structures, The International Journal of Advanced Manufacturing Technology 104 (10 2019).

[8] Z. Izri, A. Bijanzad, S. Torabnia, I. Lazoglu, In silico evaluation of lattice designs for additively manufactured total hip implants, Computers in Biology and Medicine 144 (2022) 105353.

[9] A. Armanfar, E. Ustundag, E. Gunpinar, G-Puzzle: Infilling 3D Models With Reinforced G-Lattices, Journal of Mechanical Design 145 (10) (2023) 101702.

[10] H. Abdulhadi, A. Mian, Effect of strut length and orientation on elastic mechanical response of modified body-centered cubic lattice structures, Proceedings of the Institution of Mechanical Engineers, Part L: Journal of Materials: Design and Applications 233 (2019) 146442071984108.

[11] Y. Liu, S. Zhuo, Y. Xiao, G. Zheng, G. Dong, Y. F. Zhao, Rapid modeling and design optimization of multi-topology lattice structure based on unit-cell library, Journal of Mechanical Design 142 (9) (2020).

[12] C. Sharpe, C. C. Seepersad, Lattice structure optimization with orientation-dependent material properties, Journal of Mechanical Design 143 (9) (2021).

[13] S. Y. Choy, C.-N. Sun, K. F. Leong, J. Wei, Compressive properties of ti-6al-4v lattice structures fabricated by selective laser melting: Design, orientation and density, Additive Manufacturing 16 (2017) 213–224.

[14] M. Leary, M. Mazur, H. Williams, E. Yang, A. Alghamdi, B. Lozanovski, X. Zhang, D. Shidid, L. Farahbod-Sternahl, G. Witt, I. Kelbassa, P. Choong, M. Qian, M. Brandt, Inconel 625 lattice structures manufactured by selective laser melting (slm): Mechanical properties, deformation and failure modes, Materials Design 157 (2018) 179–199.

[15] M. Mazur, M. Leary, M. McMillan, S. Sun, D. Shidid, M. Brandt, 5 - mechanical properties of ti6al4v and alsi12mg lattice structures manufactured by selective laser melting (slm), in: M. Brandt (Ed.), Laser Additive Manufacturing, Woodhead Publishing Series in Electronic and Optical Materials, Woodhead Publishing, 2017, pp. 119–161.

[16] H. Lei, C. Li, J. Meng, H. Zhou, Y. Liu, X. Zhang, P. Wang, D. Fang, Evaluation of compressive properties of slm-fabricated multi-layer lattice structures by experimental test and -ct-based finite element analysis, Materials Design 169 (2019) 107685.

[17] H. Xia, J. Meng, J. Liu, X. Ao, S. Lin, Y. Yang, Evaluation of the equivalent mechanical properties of lattice structures based on the finite element method, Materials 15 (9) (2022).

[18] R. Kulagin, Y. Beygelzimer, Y. Estrin, A. Schumilin, P. Gumbsch, Architectured lattice materials with tunable anisotropy: Design and analysis of the material property space with the aid of machine learning, Advanced Engineering Materials 22 (12) (2020) 2001069.

[19] H. Hassanin, Y. Alkendi, M. Elsayed, K. Essa, Y. Zweiri, Controlling the properties of additively manufactured cellular structures using machine learning approaches, Advanced Engineering Materials 22 (3) (2020)

1901338.

[20] A. A. Tasmektepligil, E. Gunpinar, Splinelearner: Generative learning system of design constraints for models represented using b-spline surfaces, Advanced Engineering Informatics 51 (2022) 101478.

[21] E. Gunpinar, S. Gunpinar, A shape sampling technique via particle tracing for cad models, Graphical Models 96 (2018) 11–29.

[22] P. Audze, V. Eglais, New approach for planning out of experiments, Problems Dynamics and Strengths 35 (1977) 104–107.

[23] M. Leary, M. Mazur, H. Williams, E. Yang, A. Alghamdi, B. Lozanovski, X. Zhang, D. Shidid, L. Farahbod-Sternahl, G. Witt, I. Kelbassa, P. Choong, M. Qian, M. Brandt, Inconel 625 lattice structures manufactured by selective laser melting (SLM): Mechanical properties, deformation and failure modes, Materials Design 157 (2018) 179–199.

[24] V. Vakharia, M. Shah, P. Nair, H. Borade, P. Sahlot, V. Wankhede, Estimation of lithium-ion battery discharge capacity by integrating optimized explainable-ai and stacked lstm model, Batteries 9 (2) (2023).

[25] D. M. Belete, M. D. Huchaiah, Grid search in hyperparameter optimization of machine learning models for prediction of hiv/aids test results, International Journal of Computers and Applications 44 (9) (2022) 875–886.

[26] V. Suthar, V. Vakharia, V. K. Patel, M. Shah, Detection of compound faults in ball bearings using multiscale-singan, heat transfer search optimization, and extreme learning machine, Machines 11 (1) (2023).

[27] J. Wieczorek, C. Guerin, T. McMahon, K-fold cross-validation for complex sample surveys, Stat 11 (1) (2022) e454.

[28] E. Samaniego, C. Anitescu, S. Goswami, V. Nguyen-Thanh, H. Guo, K. Hamdia, X. Zhuang, T. Rabczuk, An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications, Computer Methods in Applied Mechanics and Engineering 362 (2020) 112790.

[29] V. M. Nguyen-Thanh, C. Anitescu, N. Alajlan, T. Rabczuk, X. Zhuang, Parametric deep energy approach for elasticity accounting for strain gradient effects, Computer Methods in Applied Mechanics and Engineering 386 (2021) 114096.

[30] D. Maturana, S. Scherer, Voxnet: A 3d convolutional neural network for real-time object recognition, in: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015, pp. 922–928. doi:10.1109/IROS.2015.7353481.

[31] E. Gunpinar, A. Armanfar, Helical5am: Five-axis parametrized helical additive manufacturing, Journal of Materials Processing Technology 304 (2022) 117565.

[32] A. Armanfar, A. A. Taşmektepligil, E. Ustundag, I. Lazoglu, E. Gunpinar, A mechanical property prediction system for g-lattices via machine learning, Engineering Optimization (2024) 1–19.