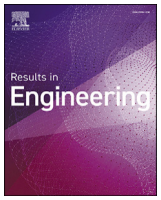
Results in Engineering 20 (2023) 101587

Contents lists available at ScienceDirect

Results in Engineering

journal homepage: www.sciencedirect.com/journal/results-in-engineering



Research paper

Application of machine learning-based algorithms to predict the stress-strain curves of additively manufactured mild steel out of its microstructural characteristics

Jorge Lizarazu [a,b,*], Ehsan Harirchian [b], Umar Arif Shaik [b], Mohammed Shareef [b], Annie Antoni-Zdziobek [c], Tom Lahmer [a,b]

[a] Material Research and Testing Institute at the Bauhaus-Universität Weimar (MFPA Weimar), Coudraystr. 9, 99423 Weimar, Germany
[b] Institute of Structural Mechanics (ISM), Bauhaus-Universität Weimar, Marienstr. 15, 99423 Weimar, Germany
[c] Univ. Grenoble Alpes, Grenoble INP, CNRS, SIMaP, 38000 Grenoble, France

ARTICLE INFO

Keywords:
Arc-direct energy deposition
Mild steel
Dual phase steel
Machine learning
Stress-strain curve

ABSTRACT

The study presents a Machine Learning (ML)-based framework designed to forecast the stress-strain relationship of arc-direct energy deposited mild steel. Based on microstructural characteristics previously extracted using microscopy and X-ray diffraction, approximately 1000 new parameter sets are generated by applying the Latin Hypercube Sampling Method (LHSM). For each parameter set, a Representative Volume Element (RVE) is synthetically created via Voronoi Tessellation. Input raw data for ML-based algorithms comprises these parameter sets or RVE-images, while output raw data includes their corresponding stress-strain relationships calculated after a Finite Element (FE) procedure. Input data undergoes preprocessing involving standardization, feature selection, and image resizing. Similarly, the stress-strain curves, initially unsuitable for training traditional ML algorithms, are preprocessed using cubic splines and occasionally Principal Component Analysis (PCA). The later part of the study focuses on employing multiple ML algorithms, utilizing two main models. The first model predicts stress-strain curves based on microstructural parameters, while the second model does so solely from RVE images. The most accurate prediction yields a Root Mean Squared Error of around 5 MPa, approximately 1% of the yield stress. This outcome suggests that ML models offer precise and efficient methods for characterizing dual-phase steels, establishing a framework for accurate results in material analysis.

1. Introduction

1.1. Additive manufacturing

Additive manufacturing (AM) has evolved as a disruptive technology not only for rapid prototyping, but also for building structures with a complex topology out of given Computer Aided Design (CAD) models. As a result, pure academic and industry-oriented research in this field have increased visibly in recent years [1].

Unlike traditional or conventional metal manufacturing methods, AM is based on a layer-by-layer welding procedure, being arc-Direct Energy Deposition (arc-DED) one technique that, due to its higher deposition rates [2], makes production of relatively large parts possible; however, at a cost of losing certain accuracy and increasing notoriously the surface roughness of the final product [3].

A first insight about the mechanical behavior of any material is given by its uni-axial relation between stresses and strains, since essential properties such as the Young's modulus and the yield stress are extracted from it; furthermore, nonlinear strain hardening parameters when the material enters into the plastic range are also determined out of the mentioned curve.

More than general, the uni-axial stress-strain relationship for a given metallic material is obtained experimentally, which involves an expensive and time consuming procedure, considering that a proper number

* Corresponding author at: Institute of Structural Mechanics (ISM), Bauhaus-Universität Weimar, Marienstr. 15, 99423 Weimar, Germany.
E-mail addresses: jorge.lizarazu@uni-weimar.de (J. Lizarazu), ehsan.harirchian@uni-weimar.de (E. Harirchian), arif.shaik@gmx.de (U.A. Shaik), sharif.ms10@gmail.com (M. Shareef), annie.antoni@phelma.grenoble-inp.fr (A. Antoni-Zdziobek), tom.lahmer@uni-weimar.de (T. Lahmer).

https://doi.org/10.1016/j.rineng.2023.101587
Received 11 September 2023; Received in revised form 1 November 2023; Accepted 10 November 2023
Available online 27 November 2023

of coupons for a tensile test should be well prepared. These disadvantages are enhanced if the effect of the phase fractions is to be investigated, resulting in several series of experiments. Therefore, the main goal of the present work is to determine the stress-strain curves of a given material when its phase fractions can be modified. This is achieved with the aid of numerical simulations and the usage of ML techniques.

To carry out the numerical simulations for predicting the mechanical behavior of an AM part at a component scale (macro scale), it is crucial to look at the microstructural characteristics of the as built material. This reveals which numerical procedure is to be used after categorizing the component as homogeneous or heterogeneous. Additionally, it provides certain information whether the component will behave anisotropically or not.

As a matter of fact, the approach followed in this work clearly belongs to the fourth paradigm of science. In the context of Material Science, every paradigm represents a key step in understanding and developing better materials [4]; for instance, the pursuit of knowledge was solely based on empirical methods in the first paradigm; while in the second one, the formulation of mathematical models based on the laws of physics was feasible, although analytical solutions were found only for very simple practical problems. Later came the era of the third paradigm, where it was possible to solve approximately the set of equations of the second paradigm with the aid of computers by implementing numerical techniques. Nowadays, the fourth paradigm is being established and characterized by big data driven science, which in the present case is given by the utilization of hundreds of results obtained after the first three paradigms to make accurate prognosis.

1.2. Machine learning

ML, which is one of the most common types of Artificial Intelligence (AI), can figure out any relation between input and output in order to make predictions. It has been successfully applied to establish such relations in many engineering applications such as in [5–8] to name a few. In Solid Mechanics, it has been applied in material property prediction [9–11], crack recognition [12,13], damage classifications [14–16], and additive manufacturing [17–20] among others.

There have been several studies that demonstrate the efficacy of ML in predicting the mechanical response of AM parts. For instance, Artificial Neural Networks (ANNs) have been shown to achieve high accuracy in predicting the tensile strength of hybrid composites [21]. Support vector regression has been used to predict the behavior of geomaterials [22]. Random Forest (RF) has been used to predict the compressive strength of basalt fiber reinforced concrete [23]. In a study by K. Yang et al. [24], a framework to predict the Young's modulus of silicate glasses by combining molecular simulations with ML has been developed. In their work, RF and feed-forward neural networks were adopted to predict the Young's modulus, achieving an accuracy of 99.1% and 98%, respectively. Moreover, X. Liu et al. [25] analyzed fracture toughness measurements by using ML models such as regression trees and ANNs. It was demonstrated that, for such engineering problems, ML solutions constitute a major improvement against their empirical counter parts in terms of reliable functionality and rapid deployment. Additionally, ML algorithms were used to link the macro and micro scales in order to carry out localization on high-contrast composite materials [26], where two-phase RVEs were represented as two-color (black and white) domains. Furthermore, Convolutional Neural Networks (CNNs) have been used to determine stress and strain fields [27]. For example, Yang et al. [28] generated the elastic strain field, including local strain concentrations, for composite materials.

Further applications of ML in Solid Mechanics include the realization of inverse analysis for the design of Metamaterials; for this, Deep Learning (DL), a branch of ML that employs ANNs, is used [29]. The goal here is to find out for example topological characteristics in order to obtain desired material properties [30], such as higher stiffness [31], superior buckling resistance [32] and optimized energy-absorbing capabilities [33]. Once the geometry is determined, AM is employed to build the final parts.

In the present work, the input for training ML-based algorithms is given by specific microstructural characteristics provided either as a set of numerical parameters or as an image, while the output for training consists of a one-dimensional array holding the stresses at given strains, which are to be obtained deterministically i.e. after the Principles of Classical Mechanics.

2. Methodology

Basically, the present work is carried out in two stages. First, the generation of training and testing data; then, the application of several AI techniques on it in order to learn and predict the stress-strain curves of the given metallic material.

2.1. Generation of training and testing data

The process of data generation, either for training or testing, is the one proposed by Lizarazu et al. [3], where each RVE is created out of a Voronoi Tessellation. Here, this process is briefly summarized and further complemented.

2.1.1. Microstructural investigation

A thoroughly study of the material's microstructure is performed once with the aid of Light Microscope, Scanning Electron Microscope (SEM) and X-ray diffraction, in order to determine heat affected regions, grain characteristics and phase composition. Lizarazu et al. [3] reported that the material's phase volume fractions are 99% Ferrite and 1% Cementite. This information was further confirmed after simulating the solidification process of the material with the aid of Thermo-Calc software (version 2021) and TCFE9 Steels/Fe-alloys database [34], whose result, showing the evolution of the phase fraction composition in relative mass, is depicted in Fig. 1. In this figure, the abscissa represents the cooling down temperature during the manufacturing process and the ordinate represents the relative amount of mass for a given phase and temperature; then, it uses a scale between 0 and 1, where a value of 0 indicates the total absence of that phase and 1, which is equivalent to 100%, shows that the whole alloy is composed exclusively of that given phase. Fig. 1 shows how four phases evolve when the material cools down. It can be seen in this figure that for temperatures lower than 600 °C, the material is composed only by two phases: Ferrite and Cementite; reaching the former a relative mass value of almost 1 and the latter a value close to zero.

2.1.2. Generation of the representative volume elements

After processing the information extracted from the SEM with the Java-based image processing program ImageJ [35], the type of grain size distribution and its statistical parameters were determined for images of different sizes. This information was further used by Lizarazu et al. to generate corresponding synthetic representative volume elements with the aid of the Linux-based computer program Neper v. 4.2.0 [36] and perform numerical simulations in order to find the dimensions of the RVE for the material under study, which resulted to be 36 μm x 36 μm. In this work, this RVE is named the *mean* RVE, since *some* of its properties (see Table 1) represent the *mean* or *initial* values for the generation of 1000 analogous sets by using LHSM.

Provided the type of distribution (Log-normal), the statistical parameters (number of grains, mean grain size and standard deviation for each phase) and the phase fraction for 1000 sets, Neper was employed again in this work to generate synthetically 1000 unique RVEs using the Voronoi Tessellation. All these synthetic RVEs are basically 1000 images in black and white (see Fig. 2), just to differentiate the Ferrite and Cementite phases, respectively.

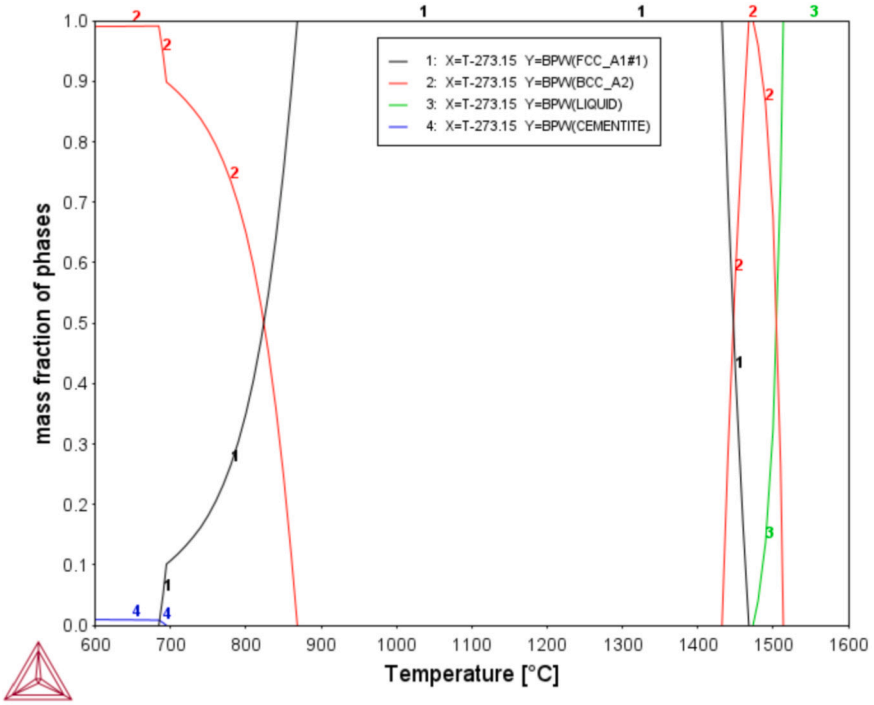


Fig. 1. Evolution of phase fractions.

Fig. 2. Two samples of synthetically generated RVEs.

Table 1
Initial input set of values for Neper.

Parameter	Value
Mean grain size of Ferrite	3.61 μm
Mean grain size of Cementite	1.33 μm
Std. deviation for Ferrite	3.46 μm
Std. deviation for Cementite	1.16 μm
Volume fraction of Cementite	1%

In principle, 200 grains were considered for each RVE, requiring this about 8 minutes for the corresponding tessellation in Neper. In order to speed up the process the number of grains was reduced to 100 and a Virtual Machine from Google Cloud Platform (e2-medium 4 CPUs, each one with 12 GB RAM memory and Operating System Ubuntu 20.04 LTS) was employed to perform the same task. With this machine capabilities, the generation of 1000 tessellations and their corresponding Finite Element discretizations demanded 4 hours of calculation.

2.1.3. Constitutive model

The hardening law of the Ferrite phase for the uniaxial case is given by Eq. (1), which was proposed by Rodriguez and Gutiérrez [37]. The first term (σ_0) on the right hand side of this equation includes the effect of the lattice friction and the elements in the solid solution, whose accurate determination for the alloy under study is reported in [3] after a chemical analysis. The second term on the right hand side of Eq. (1) describes the hardening behavior of the phase and includes several parameters, whose identification for the current material was also reported by Lizarazu et al. [3] and their meaning is explained in Table 2. In Eq. (1), the recovery rate is defined in terms of the Ferrite mean grain size d_α as $k_r = 10^{-5}/d_\alpha$ [38] and the displacement mean free path L is actually the mean grain size d_α.

$$\sigma = \sigma_0 + \alpha M \mu \sqrt{b} \sqrt{\frac{1 - e^{-M k_r \epsilon_p}}{k_r L}} \tag{1}$$

The model can be simplified by reducing the number of parameters; then, the bulk number $B = \alpha \mu \sqrt{b} = 0.19$ MPa is defined, as a result Eq. (1) can be written as

$$\sigma = \sigma_0 + M B \sqrt{1 - e^{-\frac{M d_\alpha}{10^{-5}} \epsilon_p}}. \tag{2}$$

The bulk number B in Eq. (2) is the last parameter to be varied using the LHSM.

```
***tess
 **format
   3.4
 **general
   2 standard
 **cell
  100
  *id
   1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
  20 21 22 23 24 25 26
   27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49
   50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72
   73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95
   96 97 98 99 100
  *mode
   1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 1 1 1
   1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
   1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

(a)

(b)

Fig. 3. Example of section assignment to a given RVE.

Table 2
Set of parameters for the uniaxial hardening law of the Ferrite phase.

Symbol	Name	Units
M	Taylor's factor	-
α	Proportionality factor	-
μ	Shear modulus	MPa
b	Burger's vector	-
L	Displacement mean free path	m
k_r	Recovery rate	-
ϵ_p	Equivalent plastic strain	-

Cementite is a brittle material with high strength (up to 3000 MPa), i.e. it practically does not exhibit plastic behavior; then, its constitutive formulation is solely based on Hook's Law with Young's modulus of 210 GPa and Poisson's ratio of 0.3 [38].

2.1.4. Finite element modeling

Once the 1000 RVEs are generated after employing LHSM and the Voronoi Tessellation, it follows their discretization into linear triangular finite elements using Neper again. All further steps concerning the FE modeling are done with the aid of ABAQUS CAE, whose final purpose is to simulate a tensile test that will result in a set of 1000 stress-strain unique curves. Here, the structure is to be stretched until it achieves a maximum total strain of 30%, as it was the maximum strain value achieved by the metallic specimens before the material brakes.

The procedure in ABAQUS CAE consists of assigning the previously presented material properties to each phase and finite element in every RVE. After that, periodic boundary conditions are applied to each RVE [39] in order to later simulate a tensile test. This process is almost fully automatized by implementing several Python subroutines as described in Fig. 4.

Fig. 3a shows a snapshot of a .tess file, where mode identifiers separate the phases into Modes 1 and 2 (Ferrite and Cementite, respectively) and Fig. 3b shows the visualization corresponding to the same *.tess file. Though, these visualization images (.png) are not considered in this stage of work, they are used later as input for training the ML-algorithm.

2.2. Machine learning for stress-strain prediction

In this paper, two types of supervised ML models are implemented based on the nature of the input data (see Fig. 5):

A. An algorithm to predict the response in terms of stress-strain curves of AM mild steel out of microstructural parameters.
B. An algorithm to predict the response in terms of stress-strain curves of AM mild steel out of images of synthetically generated RVEs.

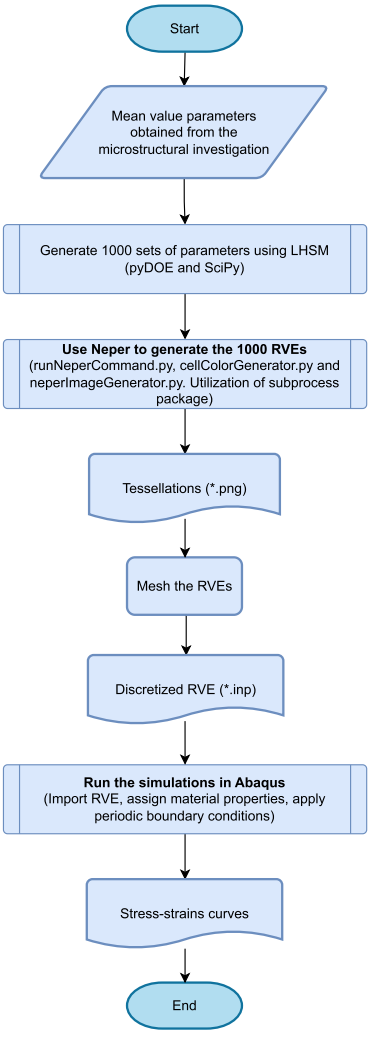


Fig. 4. Semi-automatic workflow of data generation and calculation processes.

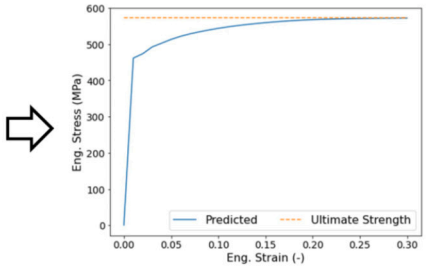
The development of both above mentioned algorithms was achieved by following several stages of the ML life cycle, as shown in Fig. 6.

The predictions can be done in terms of Engineering Stress/Strain measurements, as well as in terms of True Stress/Logarithmic Strain measurements, everything depending on the data used for training the ML models.

ML algorithms can not fit solely on raw data; instead, the data should be pre-processed to meet the requirements of individual algorithms; as

(a)

Stat. measure	cementiteMean (µm)	ferriteMean (µm)	cementitePerc (%)	Taylor's factor (-)	bulkConstant (MPa)
mean	1,32	3,57	4,59	2,97	0,19
std	0,03	0,08	1,04	0,07	0,00
min	1,22	3,32	2,15	2,76	0,18
25%	1,30	3,52	3,89	2,92	0,19
75%	1,34	3,62	5,28	3,01	0,19
max	1,42	3,84	9,32	3,19	0,20



(b)

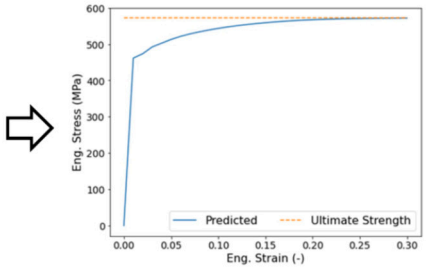


Fig. 5. Input data type on left: (a) Microstructural parameters [3] and (b) Synthetic RVE-image. On right, output data as corresponding stress-strain curves.

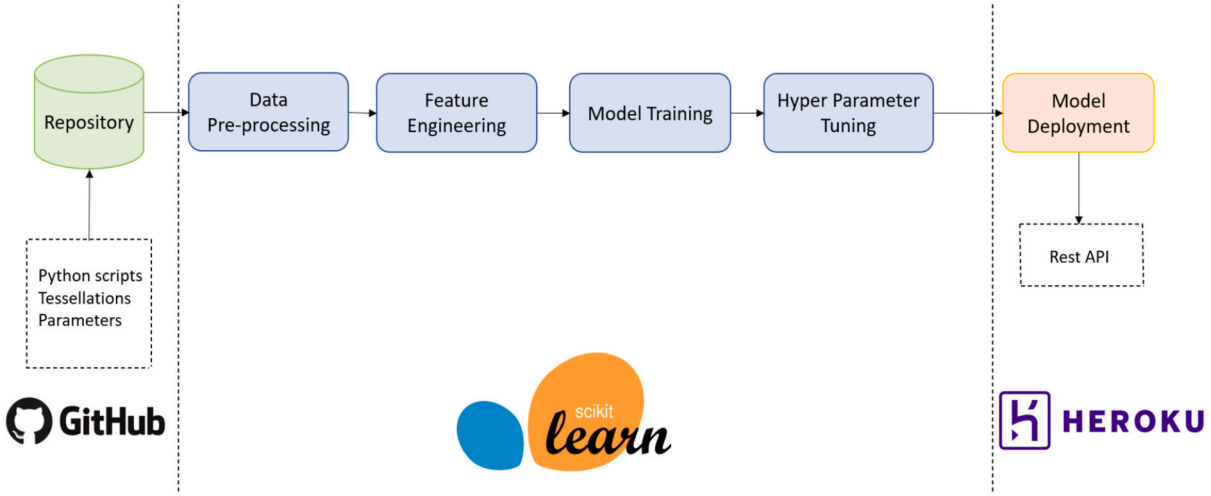


Fig. 6. Stages of the ML implementation.

such, data is transformed into useful information at every stage until the deployment, where predictions can be made from the model artifacts. In this study, two end-to-end ML pipelines were developed, which automates the stages shown in Fig. 6.

2.2.1. Data collection

In this work, the synthetic generation of data was carried out programmatically and it involved more than half of the study period. The final data consisted of:

- Microstructural parameters: Generated by applying LHSM.
- RVE: As tesselletations generated in Neper.
- Corresponding stress-strain curves obtained after simulating tensile tests with ABAQUS CAE.

2.2.2. Exploratory data analysis

Exploratory Data Analysis (EDA) is concerned with analyzing and understanding the trends of the data set. In this study, EDA was carried out in Python using packages 'pandas', 'matplotlib' and 'sweetViz'. Table 3 shows the descriptive statistics of input parameters.

2.2.3. Data pre-processing

Data pre-processing is vast and can consist of numerous steps depending upon the data type, features, and model requirements. The following two subsections describe the pre-processing steps implemented in the raw dataset.

1. **Pre-processing of input data:** The generation of input data was explained beforehand, where 1000 sets of parameters with corresponding RVE-Images were provided, as presented in Fig. 7. As part of a ML life cycle, pre-processing was carried out to recognize and fix errors in the dataset that may negatively influence a predictive model. The steps carried out to clean and prepare the input data are:
   - **Standardization:** Data for each feature drawn from various sources may differ from each other in terms of scale and distribution. The process of standardizing a dataset entails rescaling the distribution of values so that the mean of observed values is 0 and the standard deviation is 1. It can be considered as centering the data.
   - **Feature Selection:** While developing a predictive model, feature selection restricts the number of independent variables in an input

Table 3
Descriptive statistics of input parameters for 1000 samples for a log-normal Distribution.

Stat. measure.	cementiteMean (μm)	ferriteMean (μm)	cementitePerc (%)	Taylor's factor (-)	bulkConstant (MPa)
mean	1.32	3.57	4.59	2.97	0.19
std	0.03	0.08	1.04	0.07	0.00
min	1.22	3.32	2.15	2.76	0.18
25%	1.30	3.52	3.89	2.92	0.19
75%	1.34	3.62	5,28	3.01	0.19
max	1.42	3.84	9.32	3,19	0.20

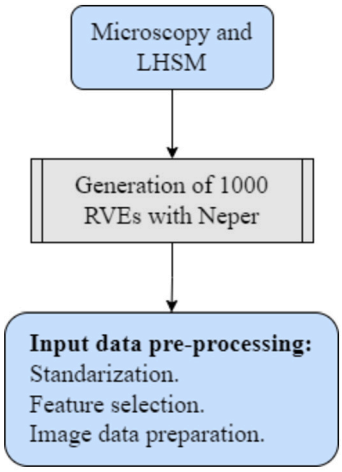


Fig. 7. Data pre-processing stages.

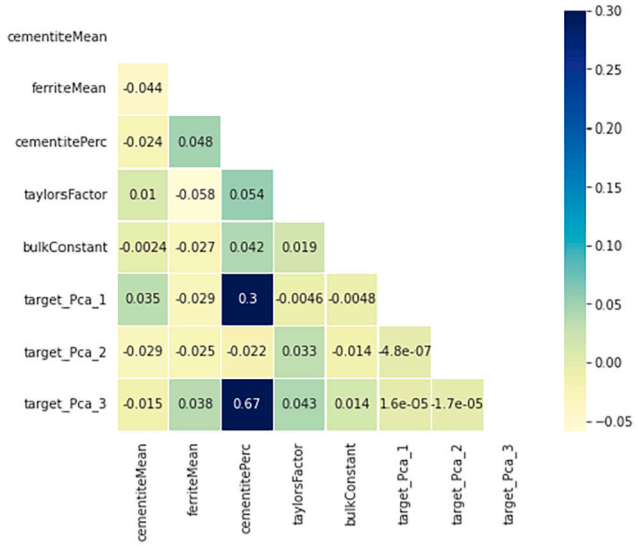


Fig. 8. Heat map showing Pearson correlation values.

dataset, which aids in both lowering the computational cost and improving model performance. Feature selection methods analyse the relationship between every independent variable to the dependent variable (target), eliminating variables that contain or hold less importance with regard to target attributes. There are several statistical methods to perform feature selection in the form of hypothesis tests. Such methods differ based on the type of dataset available. For numerical data, methods such as Pearson or Spearman's correlation and ANOVA or chi-square test for categorical data can be used. In this work, Pearson correlation was employed to rank the independent features based on their relationship with target stress-strain curves.

Fig. 8 illustrates the input feature correlations with the dependent variable class, i.e., stress-strain curves. It can be observed that none of the input parameters have much influence on target variables except that of cementite percentage. Here, the target variables are of a two-dimensional continuous value, which can be labeled as a multi-output regression problem. Dimensionality reduction methods were applied to reduce the two-dimensional stress-strain value into a lower-dimensional plane. As shown in Fig. 8, the last three variables, target_pca_1, target_pca_2 and target_pca_3, are the lower-dimensional representation of the higher dimensional stress-strain values, also known as Principal Component (PC). A detailed description of the dimensionality reduction approach used in the study is discussed in the following sections.

- **Image Data preparation:** The second input entity generated to study the material behavior is of unstructured data type, i.e. images of an RVE tessellation. ML algorithms do not understand the unstructured data and cannot be used directly for training unless converted into vectors. Several image processing steps were carried out to convert images into machine-readable vectors. Each image generated from Neper software was of (1000×1000×3) dimensions, where 1000×1000 is a pixel dimension and 3 corresponds to the RGB channel of a particular image. Resizing an image was performed to bring down the pixel size to 420×420×3, which saves computational time and improves the model's accuracy. Fig. 9 shows a few images obtained after pre-processing.

2. **Pre-processing of simulation output data:** The stress-strain values obtained from ABAQUS CAE were treated as a set of target attributes which the ML model should predict. However, the output, the two-dimensional vector of each RVE, is of different size, i.e. the number of points obtained to draw each stress-strain curve is different from one RVE to another, as ABAQUS CAE does not provide the flexibility to control this. However, it is essential that the target variables be of the same dimension and shape. To achieve this, an interpolation approach by using cubic splines was first employed; so that, a new (interpolated) set of points for each stress-strain curve is generated. As a result, every stress-strain curve can be now described with the same number of points and storage of an specific strain vector for every curve is no longer needed. Further, dimensionality reduction techniques are to be used, which transform the stress-strain curves into an effective latent space of lower dimension.
   - **Cubic spline interpolation:** The process of estimating (finding) unknown values or data points that fall between known values is known as interpolation. Cubic spline interpolation is an exact fitting method, in which a piecewise polynomial of degree 3 is used for every couple of points. By using such 3^{rd}-order polynomials, it is possible to assure the continuity of the curve, the slope and the curvature between each picewise function. Furthermore, as the functions are actually low order polynomials, curve fitting using cubic splines does not show oscillation problems.

   In this study, the raw stress-strain curves (without pre-processing) consisted of 1000 sets with a minimum of 11 and a maximum of 66 points defining the elasto-plastic response until a total strain value of 30%, see Fig. 10. All of them were redefined based on

Fig. 9. Set of RVE images after pre-processing.

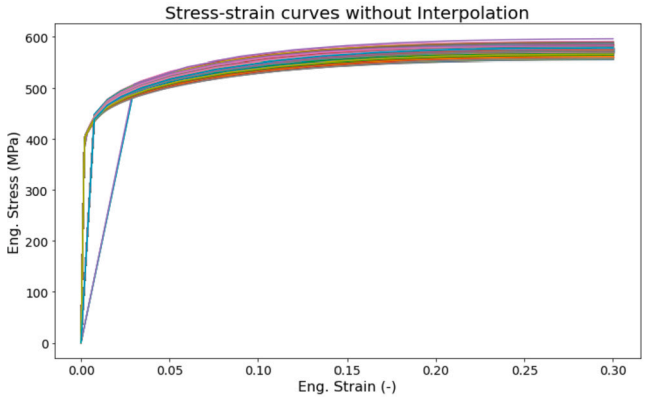


Fig. 10. 1000 raw stress-strain values obtained after simulating a tensile test in ABAQUS CAE.

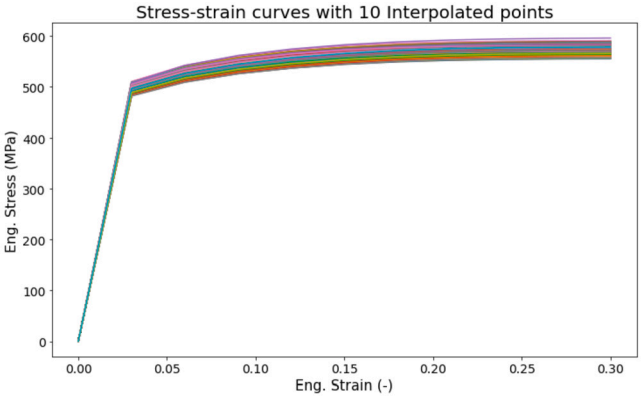


Fig. 12. A 2D Plot of 1000 stress-strain values at 10 interpolated strain points.

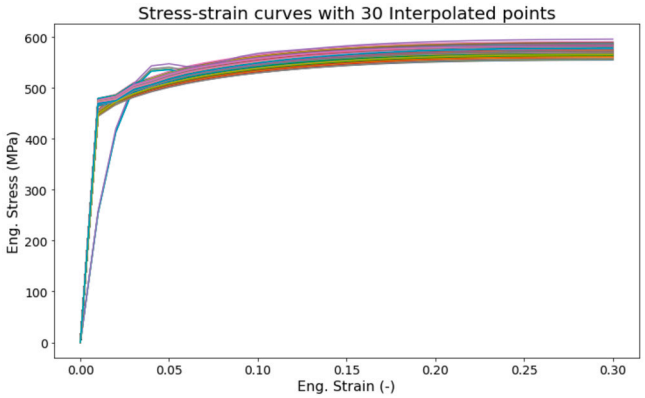


Fig. 11. A 2D Plot of 1000 stress-strain values at 30 interpolated strain points.

30 and 10 points for the corresponding ML modeling, as depicted in Figs. 11 and 12 respectively. As expected, the redefined curves with 30 points are smoother.

- **Principal component analysis:** It is a statistical technique used, for example, for dimensionality reduction of data. It consists of determining an orthonormal basis after an optimization procedure, on which the data sets are projected/referred to. Then, it is possible to lower the dimensional space by verifying that a given principal component (basis vector) accounts for a relatively much smaller variation of data. Dimensionality reduction, as it involves learning in a lower-dimensional environment, allows ML models to learn more effectively, resulting in improved accuracy with less training data.

In principle, the output data is split into training and testing data sets in a 80:20 proportion, in order to evaluate and validate the model performance. Afterwards, the dimensionality of the interpolated training data sets is reduced by using PCA for both cases, i.e. when using 30 and 10 stress-strain pairs for each simulation.

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{T}. \tag{3}$$

The $\boldsymbol{X}$ matrix in Eq. (3) depicts 1000×10 or 1000×30 components, with each row representing a stress vector whose elements correspond to the 10 or 30 total strain values. Despite the fact that stress-strain curves are made up of both a stress and a strain vector, just the stress vector is fitted/passed to PCA because the strain vectors for all (1000) cases have the same lengths and values, i.e. fitting them in PCA would have not provided more information to the model. At first, each column of $\boldsymbol{X}$ is standardized, such that, it has a mean value of 0 and a standard deviation of 1. Then, $\boldsymbol{X}$ is factorized by using singular value decomposition, as indicated in Eq. (3), where the matrix $\boldsymbol{\Sigma}$ contains the eigenvalues of each PC that measure the corresponding variance. The PCs are listed in the $\boldsymbol{V}^{T}$ matrix in decreasing order of relevance.

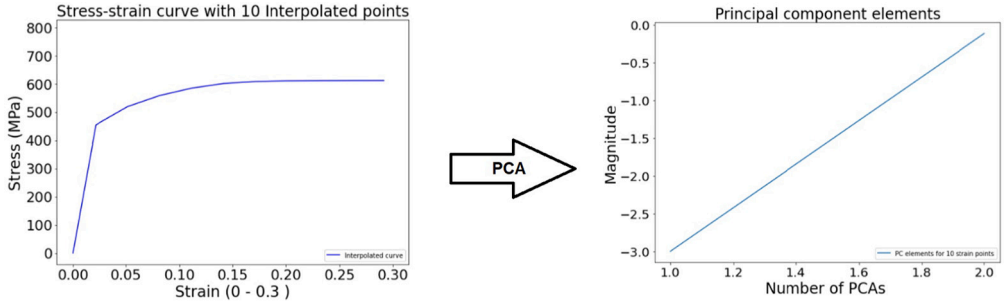


Fig. 13. Transition of stress vector from higher dimension (1000x10) to lower dimension (1000x2) with two PCs.

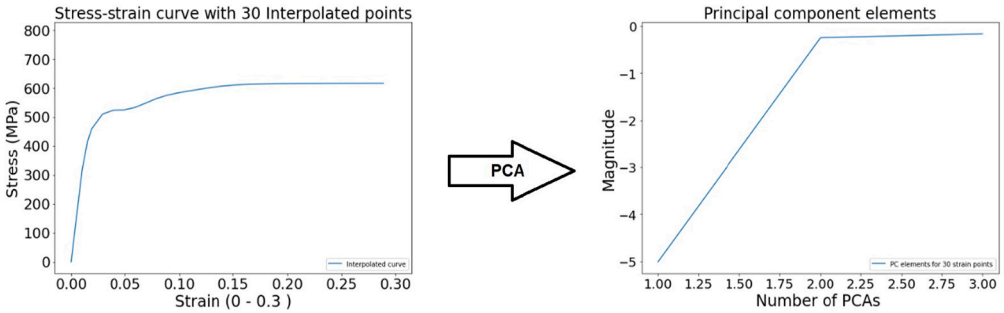


Fig. 14. Transition of stress vector from higher dimension (1000x30) to lower dimension (1000x3) with three PCs.

Table 4
Variance by two PCs for stress vector with 10 strain points.

PCA components	PC 1	PC 2
Explained variance	98.2%	1.5%

Table 5
Variance by three PCs for stress vector with 30 strain points.

PCA components	PC 1	PC 2	PC 3
Explained variance	92.2%	5.7%	1.2%

The Cumulative Explained Variance (CEV) is the sum of eigenvalues of selected PCs normalized by the sum of all eigenvalues. Then, based on CEV, 3 and 2 PCs for $\boldsymbol{X}$ when defined out of 30 and 10 strain points are considered, respectively.

In this study, PCA was implemented by using 'sklearn's PCA method from its decomposition module. Tables 4 and 5 show the CEV for the two cases. i.e. when $\boldsymbol{X}$ is defined out of 10 and 30 strain points respectively. Likewise, Figs. 13 and 14 depict the transitions from dimensionally higher spaces to lower ones for a given simulation.

2.2.4. Machine learning model building

Two end-to-end ML model pipelines were created to predict the stress vector. The first one employs traditional ML algorithms, whereas the second one is based on Deep Learning, namely CNNs.

1. **Machine learning model 1.** The prediction of the output stress vector constitutes a multi-output regression problem, since it consists of more than one numerical value. Unlike standard regression, which predicts a single value for each observation, multi-output regression needs specific ML algorithms such as RF, linear regression, k-Nearest Neighbors (KNN) and Decision Trees (DTs).

   In this model, several multi-output regressors are employed, which are listed in Fig. 15, where the flow of the ML model is also depicted. In principle, the linear regression algorithm with default parameters given by sklearn's linear model package in Python was trained with the given input and output datasets. After that, the Ridge regression algorithm with default parameters given by sklearn's linear model package by 'RidgeCV' in Python was employed. The given input and output datasets have been analyzed with the DT regression algorithm using entropy as purity criteria, making the tree with a maximum depth of five. Additionally to the previous ML algorithms, RF as an ensemble ML method for regression and classification has been implemented. RF is a type of bagging approach that originates from a significant advancement in DT variance. RF aggregates the result from multiple trees built on sample datasets. Unlike DT, RF creates multiple trees, which are executed in parallel without any interaction between them. In short, it creates multiple DT and aggregates the output generated by all predictors. The RF model is implemented by using 'sklearn's linear model package by 'RandomForestRegressor' in Python. The RF regression algorithm was set with default parameters of 100 trees and a maximum depth of 4. Along with the mentioned regression algorithms, others multi-output regression methods like KNN, Lasso, ElasticNet, Gradient Boosting Regressor, and Ada Boost Regressor were also employed for the investigation of the best fit.

   In ML, Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) and Mean Squared Error (MSE) are commonly used evaluation metrics for regression problems. MAE, according to Eq. (4), is defined as the average of the differences between true Y_i and predicted $\hat{Y}_i$ n-components of the stress vector; then, the average magnitude of error is measured, being a higher value an indicator of worse performance

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|Y_i - \hat{Y}_i\right|. \tag{4}$$

   MSE is similar to MAE; however, it uses the squared difference instead, see Eq. (5). Generally, MSE is widely used to train models because it is differentiable, unlike MAE.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2. \tag{5}$$

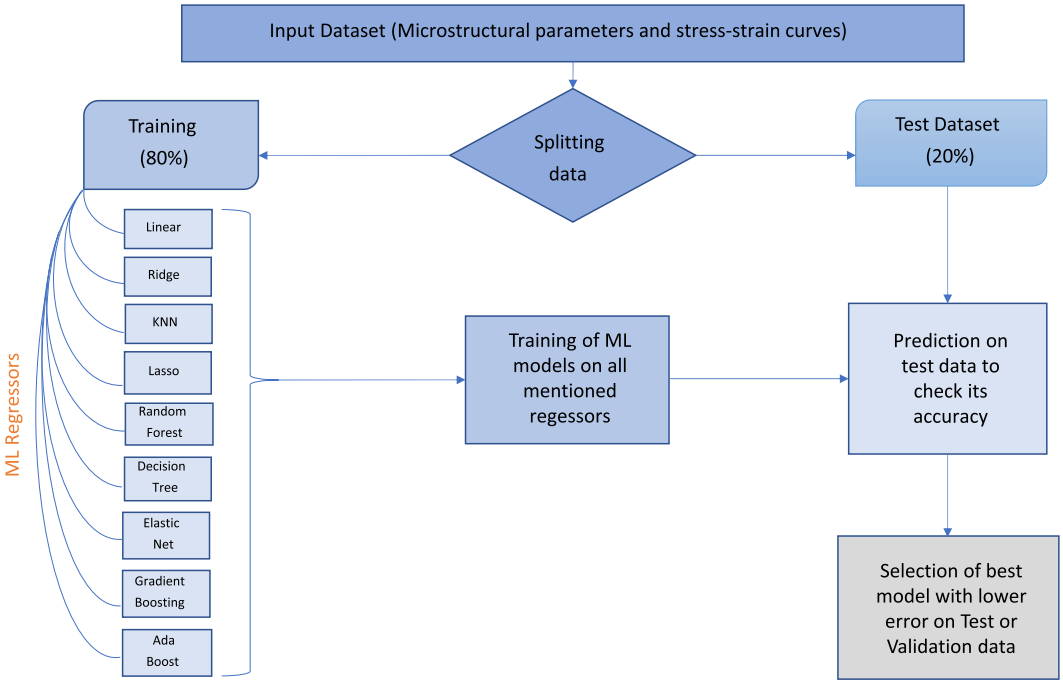


Fig. 15. Flow of ML Model 1 with several multi-output regressors.

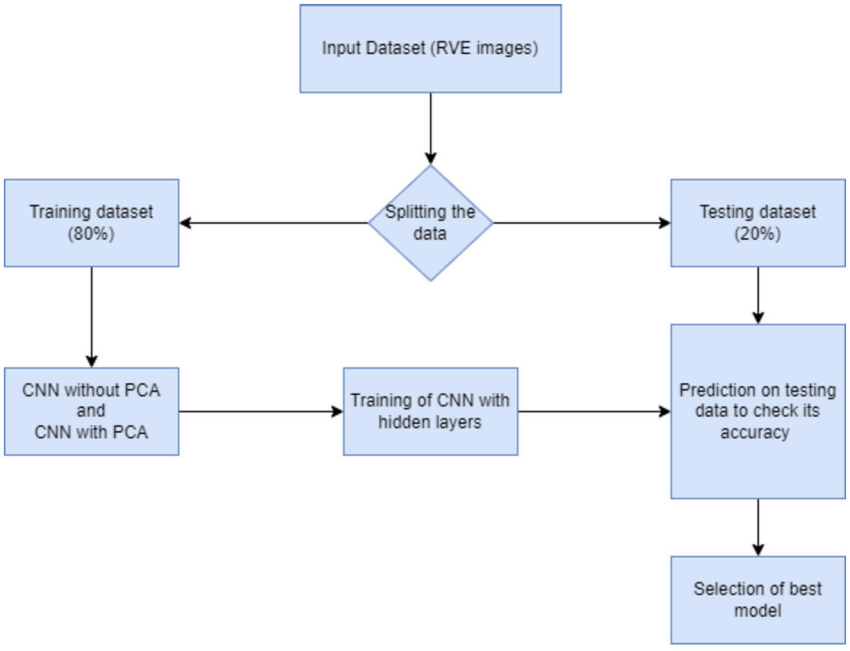


Fig. 16. Flow of ML model 2 with multiple CNN.

RMSE can be any non-negative value, including values greater than 1. RMSE as presented in Eq. (6), has the same units of the dependent variable. The lower RMSE is, the more accurate the prediction is

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2}{n}}. \tag{6}$$

2. **Machine learning model 2.**

When traditional ML algorithms can not solve a given problem due to its complexity, Deep learning-based algorithms will be used. Deep Learning is a branch of ML, which uses a multi-layered ANN. This approach involves simple Mathematics, Statistics, Physics, and Signal Processing methods. CNN is a type of ANN that mainly deals with images and videos as input. The main idea behind implementing CNN in this study is to predict the stress-strain curve of a given dual phase material out of its RVE; for this, images generated from Neper (as input) together with 30-point stress vectors are to be trained. Fig. 16 illustrates the steps regarding ML model 2.

CNN is the most popular artificial neural network because of its ability to deal with large datasets; as a result, it is commonly used to analyze and find patterns in images. Digital photos provide pixel-level information about an object. For three separate RGB channels, each pixel contains brightness intensity values ranging from 0-255. In this study, only one CNN model was implemented, which takes

Table 6
Characteristics of each assigned layer for CNN.

	Filters	Kernel size	Activation function	Max pooling
Convolutional Layer1	32	(3,3)	ReLU	(2,2)
Convolutional Layer2	64	(3,3)	ReLU	(2,2)
Convolutional Layer2	128	(3,3)	ReLU	(2,2)
Output layer	31 (Dense nodes)	-	-	-

Table 7
Metrics of ML model 1 on A) 10 interpolated stress-strain points, B) 2 PCs, C) 30 interpolated stress-strain points, and D) 3 PCs.

Algorithms/Metrics	A			B			C			D		
	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE	MSE	RMSE
KNN	3.868	25.786	5.078	3.868	25.783	5.075	4.826	45.491	6.745	4.932	48.259	6.947
Linear regression	**3.662**	24.205	4.920	**3.662**	24.207	4.920	4.553	40.873	6.393	4.552	40.876	6.393
Ridge	**3.664**	24.228	4.922	**3.664**	24.230	4.992	4.541	40.842	6.391	4.551	40.845	6.391
Lasso	3.998	28.514	5.340	3.998	28.514	5.340	4.693	42.088	6.488	4.587	40.819	6.389
Elastic Net	3.998	28.514	5.340	3.871	26.851	5.182	4.693	42.088	6.488	4.577	40.739	6.383
RF Regressor	3.745	24.825	4.982	3.746	24.845	4.984	4.580	40.773	6.385	4.580	40.794	6.387
DT Regressor	4.113	30.424	5.516	4.117	30.455	5.519	4.820	46.131	6.792	4.821	46.265	6.802
Gradient Boosting Regressor	3.865	26.486	5.146	3.871	26.470	5.145	4.575	40.496	6.364	4.580	39.441	6.280
Ada Boost Regressor	3.732	24.987	4.999	3.751	25.093	5.009	4.657	41.408	6.435	4.739	43.889	6.625

a 430×430-sized image as an input and a stress-strain curve with 30 interpolated strain points as an output. Again, the 1000 samples were split into 80% training and 20% testing data. Each convolutional layer in the CNN model consists of five main components that perform classification or regression operation on images by transforming an n-dimensional image into a single dimension vector.

CNNs are used with python scripting language employing the TensorFlow module, an open-source library provided by Google. The development of the CNN model was carried out with the characteristics presented in Table 6 for each convolutional layer.

3. Results and discussion

This section is based on the metrics evaluated in Section 2.2.4, which were calculated for structured (ML model 1) and unstructured (ML model 2) data.

3.1. Evaluation of ML model 1

The training was carried out by using 9 regressors, which were implemented using Python's 'scikit-learn' module. For this analysis, four general cases were considered:

A. ML model 1 trained by 10 interpolated stress-strain points.
B. ML model 1 trained by 2 PCs (obtained from 10-point stress-strain curves).
C. ML model 1 trained by 30 interpolated stress-strain points.
D. ML model 1 trained by 3 PCs (obtained from 30-point stress-strain curves).

Accordingly, the output data for training, namely the stress vector, is given in four formats, all of them carrying the same information, resulting in 36 (9 × 4) ML models.

According to Table 7, a better agreement between ML prediction and testing data was achieved for cases A and B, where output training was based on 10 stress-strain points. Additionally, the maximum difference between the metrics obtained by considering the whole set of 10 or 30 points and the PCA is less than 6.1%, i.e. PCA proved to be a suitable procedure to accelerate the calculations without any important loss of accuracy. Further, the metrics of the best prediction obtained after the Linear Regressor for Cases A and B are practically the same.

As already mentioned, the stress-strain curve after a tensile test for a given material provides valuable information from a macroscopical point of view. Among this information, the strength of the material, sometimes referred to as the maximum stress, has been determined for both actual and predicted stress-strain curves. Fig. 17 depicts the stress-strain curve and maximum stress, which can be also termed the material's ultimate tensile strength.

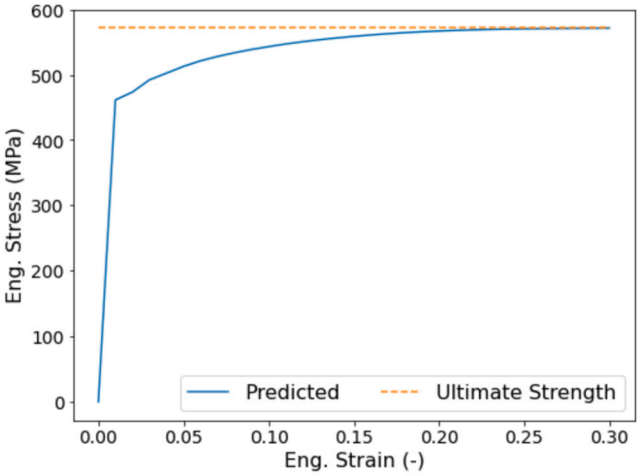


Fig. 17. Actual and predicted ultimate tensile strength of the material.

Fig. 18 shows the difference between actual (testing dataset) and predicted maximum stress; here, the x-axis represents the number of samples (the first 100 out of 200 testing samples), while the y-axis represents the strength for every chosen testing sample. For a better assessment of the relative prediction error in the prognosis of the maximum stress, Fig. 19 was prepared, where the variation of the absolute error (actual - predicted) of tensile strength is presented. It shows a maximum discrepancy of 16 MPa, that represents less than the 3% of the strength of the material.

3.2. Evaluation of ML model 2

In ML model 2, the training was based on RVE images as input data, in contrast to ML model 1, where microstructural parameters were used instead.

Deep learning-based algorithms were employed to predict the stress-strain curves. Even though synthetic RVE images are generated from microstructural parameters, the main reason to work with deep learning-based algorithms is their innovative learning with neurons, which is not

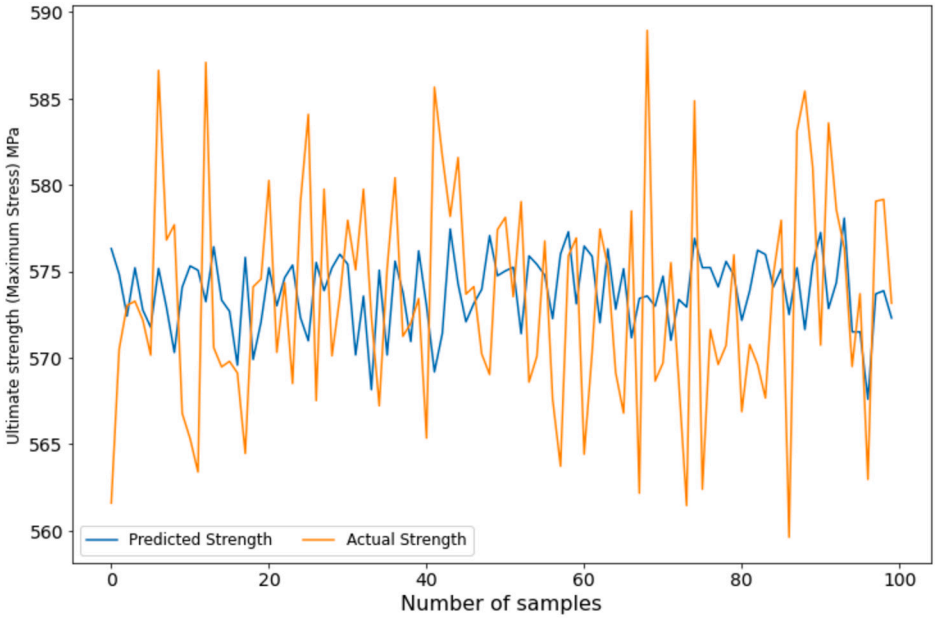


Fig. 18. Difference between actual and predicted strength (maximum stress).

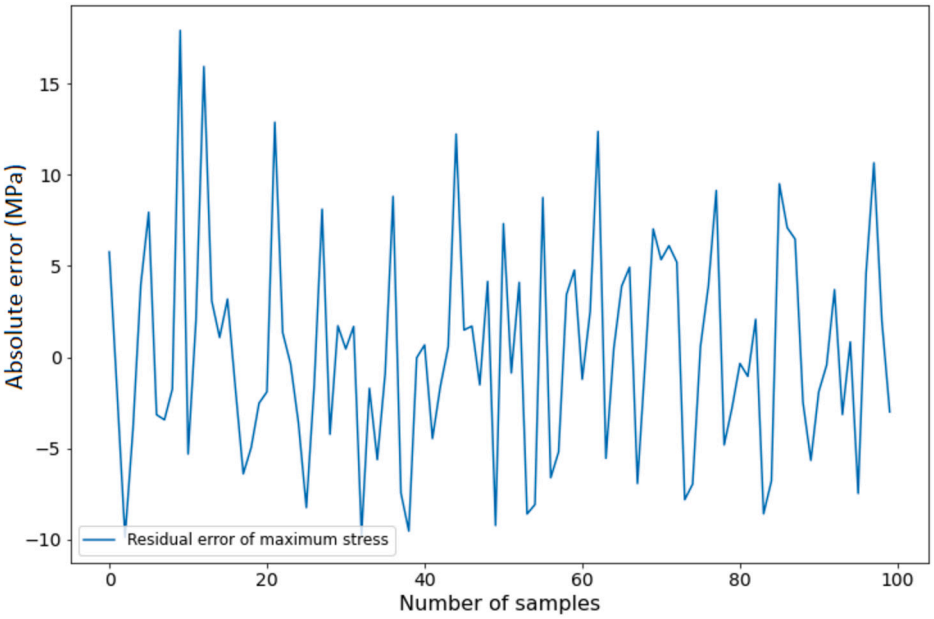


Fig. 19. Residual error of strength for the first 100 test samples.

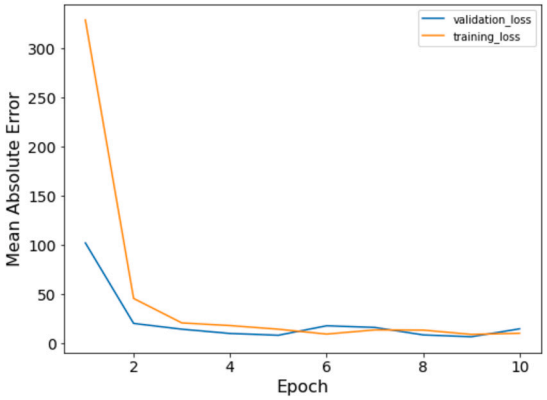


Fig. 20. Training and validation loss for 10 epochs.

Table 8
Evaluating metrics of CNN model.

Algorithm	C		
	MAE	MSE	RMSE
CNN	14.5	287.21	19.94

the case with traditional ML models; besides, it was intended the development of a tool that can directly analyze images that, for example, could be obtained from an Scanning Electron Microscope.

The same evaluation metrics of ML model 1 were used and they are shown in Table 8. Compared to that of the traditional ML model, losses are high. The mean absolute error for each epoch of the training and validation up to 10 epochs is presented in Fig. 20. As it can be seen, the errors drastically reduced after the 4th epoch, after which the training and validation losses are almost the same.

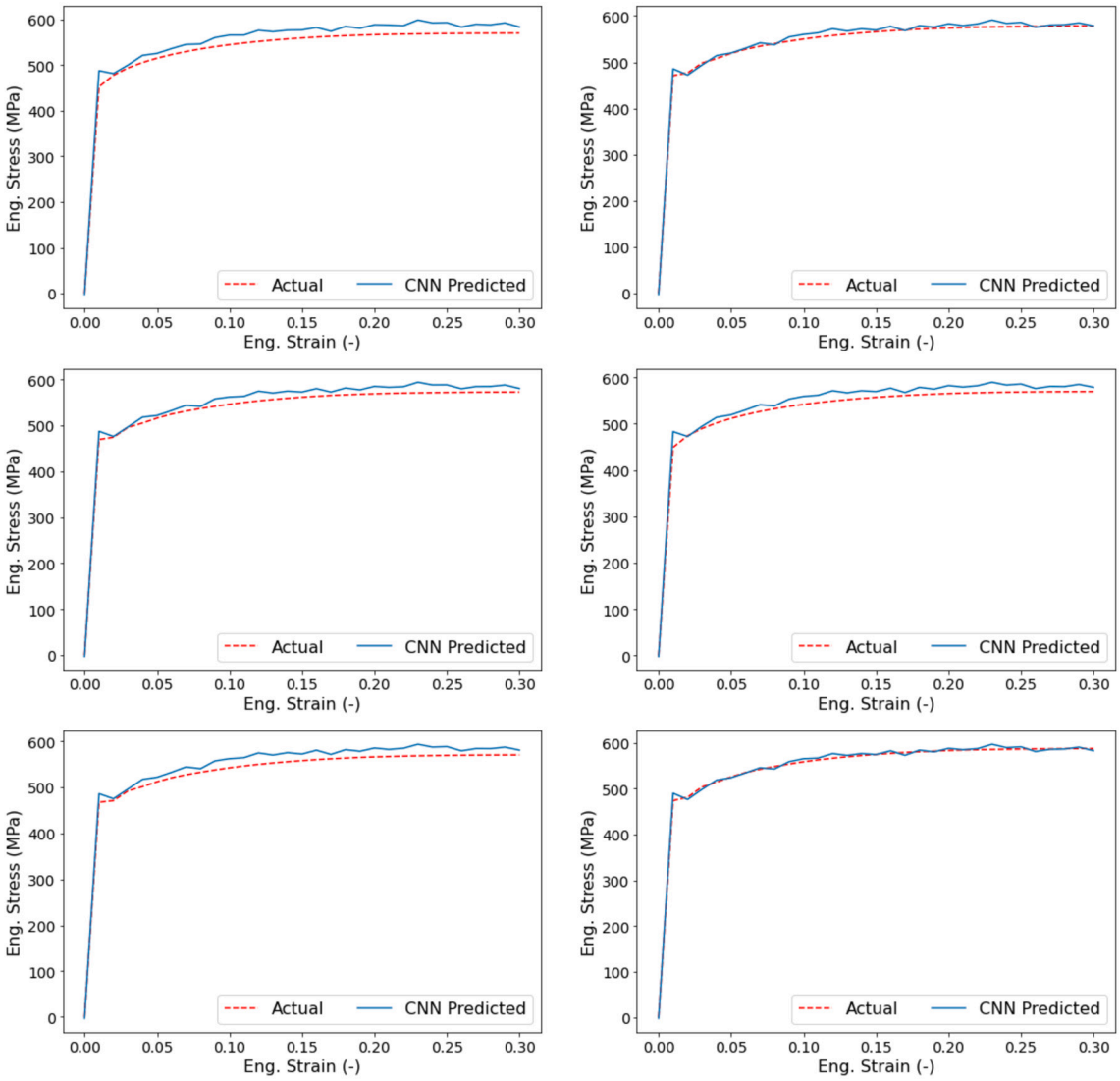


Fig. 21. Stress-strain predictions after employing CNN algorithm.

Some predictions based on the ML model 2 are depicted in Fig. 21.

4. Conclusion

A complete framework was developed from the generation of synthetic data to ML and DL model training. In addition, a web framework was developed, where inferences from the model can be drawn with a click of a button. For this purpose, 1000 synthetic samples of a given dual-phase steel were created with a semi-automatic workflow process. Then, a tensile test was simulated for each RVE using the FE Method to obtain the corresponding stress-strain curve. Afterwards, the input data is divided into structured (microstructural parameters) and unstructured data (RVE images); on which ML and DL-based models are employed to carry out the prognosis, respectively.

Three different metrics were used to assess the accuracy. On one hand, classic ML algorithms performed beyond expectations by predicting the whole non-linear stress-strain curve with a minor RMSE of approximately 5 MPa. On the other hand, although showing a satisfactory convergence, DL based predictions from RVE images had a relative larger error. Nevertheless, determining the stress-strain curve for a given dual-phase steel out of an RVE image, that could be extracted from an SEM device, without any further work, is undoubtedly much more pragmatic.

In pursuit of predicting the stress-strain curves, it was also possible to verify the effectiveness of PCA when reducing data almost without losing information. In future studies, some limitations can be improved and achieve a better and more robust method by having more data, additional computational resources and different materials.

Abbreviations

arc-DED	arc-Direct Energy Deposition
AI	Artificial Intelligence
AM	Additive Manufacturing
ANN	Artificial Neural Network
CAD	Computer Aided Design
CEV	Cumulative Explained Variance
CNN	Convolutional Neural Network
DL	Deep Learning
DT	Decision Tree
EDA	Exploratory Data Analysis
FE	Finite Element
KNN	k-Nearest Neighbors
LHSM	Latin Hypercube Sampling Method
MAE	Mean Absolute Error
ML	Machine Learning
MSE	Mean Squared Error

PC	Principal Component
PCA	Principal Component Analysis
RMSE	Root Mean Squared Error
RF	Random Forest
RVE	Representative Volume Element
SEM	Scanning Electron Microscope

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

The work is supported by almost equal parts by the following institutions / projects: Bauhaus HeriTech - Digitale Technologien für Handwerk und Kulturgut (TMWWDG, Federal State of Thuringia) under grant 5575/10-6; ODE_AM – Ontologien für die dezentrale Erfassung von mehrskaligen statischen und zyklische Kennwerten von additiv gefertigten Stahlstrukturen aus Experiment und Simulation (BMBF, Federal Ministry of Education and Research) under grant 13XP5117A; Research Group - Characterization and Functionalization of Materials and Components (Federal State of Thuringia), which is in each case highly acknowledged by the authors. Additionally, we acknowledge support for the publication costs by the Open Access Publication Fund of Bauhaus-Universität Weimar and the Deutsche Forschungsgemeinschaft (DFG).

References

[1] K.V. Wong, A. Hernandez, A review of additive manufacturing, ISRN Mech. Eng. 1 (2012) 1–10.
[2] C. Cunningham, J. Flynn, A. Shokrani, V. Dhokia, S. Newman, Invited review article: strategies and processes for high quality wire arc additive manufacturing, Addit. Manuf. 22 (2018) 672–686, https://doi.org/10.1016/j.addma.2018.06.020, https://www.sciencedirect.com/science/article/pii/S2214860418303920.
[3] J. Lizarazu, L. Göbel, S. Linne, S. Kleemann, T. Lahmer, C. Rößler, J. Hildebrand, Experimental characterization and numerical analysis of additively manufactured mild steel under monotonic loading conditions, Prog. Addit. Manuf. 5 (3) (2020) 295–304.
[4] A. Agrawal, A. Choudhary, Perspective: materials informatics and big data: realization of the “fourth paradigm” of science in materials science, APL Mater. 4 (5) (2016) 053208, https://doi.org/10.1063/1.4946894.
[5] F. Harrou, A. Dairi, A. Dorbane, Y. Sun, Energy consumption prediction in water treatment plants using deep learning with data augmentation, Results Eng. (2023) 101428.
[6] K. Khan, M.A.M. Johari, M.N. Amin, M.I. Khan, M. Iqbal, Optimization of colloidal nano-silica based cementitious mortar composites using RSM and ANN approaches, Results Eng. 20 (2023) 101390.
[7] S.M. Chiew, I.S. Ibrahim, M.A.M. Ariffin, H.-S. Lee, J.K. Singh, Assessment and ann model development of natural light transmittance of light-transmitting concrete, Results Eng. 20 (2023) 101416.
[8] M.A. Yassin, A. Usman, S. Abba, D.U. Ozsahin, I.H. Aljundi, Intelligent learning algorithms integrated with feature engineering for sustainable groundwater salinization modelling: eastern province of Saudi Arabia, Results Eng. 20 (2023) 101434.
[9] L. Ward, A. Agrawal, A. Choudhary, C. Wolverton, A general-purpose machine learning framework for predicting properties of inorganic materials, Comput. Mater. 2 (1) (2016), https://doi.org/10.1038/npjcompumats.2016.28.
[10] S. Rezvan, M.J. Moradi, H. Dabiri, K. Daneshvar, M. Karakouzian, V. Farhangi, Application of machine learning to predict the mechanical characteristics of concrete containing recycled plastic-based materials, Appl. Sci. 13 (4) (2023) 2033.
[11] Z. Fei, S. Liang, Y. Cai, Y. Shen, Ensemble machine-learning-based prediction models for the compressive strength of recycled powder mortar, Materials 16 (2) (2023) 583.
[12] S. Ruggieri, A. Cardellicchio, A. Nettis, V. Renò, G. Uva, Using machine learning approaches to perform defect detection of existing bridges, Procedia Struct. Integr. 44 (2023) 2028–2035.
[13] K. Malek, A. Mohammadkhorasani, F. Moreu, Methodology to integrate augmented reality and pattern recognition for crack detection, Comput.-Aided Civ. Infrastruct. Eng. 38 (8) (2023) 1000–1019.
[14] E. Harirchian, S.E.A. Hosseini, K. Jadhav, V. Kumari, S. Rasulzade, E. Işık, M. Wasif, T. Lahmer, A review on application of soft computing techniques for the rapid visual safety evaluation and damage classification of existing buildings, J. Build. Eng. 43 (2021) 102536.
[15] E. Harirchian, K. Jadhav, V. Kumari, T. Lahmer, Ml-ehsapp: a prototype for machine learning-based earthquake hazard safety assessment of structures by using a smartphone app, Eur. J. Environ. Civ. Eng. 26 (11) (2022) 5279–5299.
[16] M.F. Işık, F. Avcil, E. Harirchian, M.A. Bülbül, M. Hadzima-Nyarko, E. Işık, R. İzol, D. Radu, A hybrid artificial neural network—particle swarm optimization algorithm model for the determination of target displacements in mid-rise regular reinforced-concrete buildings, Sustainability 15 (12) (2023) 9715.
[17] C. Tian, T. Li, J. Bustillos, S. Bhattacharya, T. Turnham, J.X. Yeo, A. Moridi, Data-driven approaches toward smarter additive manufacturing, Adv. Intell. Syst. 3 (2021).
[18] L. Meng, B. McWilliams, W. Jarosinski, H.-Y. Park, Y.-G. Jung, J. Lee, J. Zhang, Machine learning in additive manufacturing: a review, JOM 72 (6) (2020) 2363–2377.
[19] J. Jiang, A survey of machine learning in additive manufacturing technologies, Int. J. Comput. Integr. Manuf. (2023) 1–23.
[20] S. Kumar, T. Gopi, N. Harikeerthana, M.K. Gupta, V. Gaur, G.M. Krolczyk, C. Wu, Machine learning techniques in additive manufacturing: a state of the art review on design, processes and production control, J. Intell. Manuf. 34 (1) (2023) 21–55.
[21] M.G. Vineela, A. Dave, P.K. Chaganti, Artificial neural network based prediction of tensile strength of hybrid composites, Mater. Today Proc. 5 (9) (2018) 19908–19915.
[22] H. Zhao, Z. Huang, Z. Zou, Simulating the stress-strain relationship of geomaterials by support vector machine, Math. Probl. Eng. 2014 (2014).
[23] H. Li, J. Lin, X. Lei, T. Wei, Compressive strength prediction of basalt fiber reinforced concrete via random forest algorithm, Mater. Today Commun. 30 (2022) 103117.
[24] K. Yang, X. Xu, B. Yang, B. Cook, H. Ramos, N. Krishnan, M.M. Smedskjaer, C. Hoover, M. Bauchy, Predicting the Young’s modulus of silicate glasses using high-throughput molecular dynamics simulations and machine learning, Sci. Rep. 9 (1) (2019) 1–11.
[25] X. Liu, C.E. Athanasiou, N.P. Padture, B.W. Sheldon, H. Gao, A machine learning approach to fracture mechanics problems, Acta Mater. 190 (2020) 105–112, https://doi.org/10.1016/j.actamat.2020.03.016, https://www.sciencedirect.com/science/article/pii/S1359645420302032.
[26] R. Liu, Y.C. Yabansu, A. Agrawal, S.R. Kalidindi, A.N. Choudhary, Machine learning approaches for elastic localization linkages in high-contrast composite materials, Integr. Mater. Manuf. Innov. 4 (1) (2015) 192–208.
[27] Z. Yang, C.-H. Yu, M.J. Buehler, Deep learning model to predict complex stress and strain fields in hierarchical composites, Sci. Adv. 7 (15) (2021) eabd7416.
[28] Z. Yang, Y.C. Yabansu, D. Jha, W.-k. Liao, A.N. Choudhary, S.R. Kalidindi, A. Agrawal, Establishing structure-property localization linkages for elastic deformation of three-dimensional high contrast composites using deep learning approaches, Acta Mater. 166 (2019) 335–345.
[29] X. Zheng, X. Zhang, T.-T. Chen, I. Watanabe, Deep learning in mechanical metamaterials: from prediction and generation to inverse design, Adv. Mater. (2023) 2302530.
[30] S. Kumar, S. Tan, L. Zheng, D.M. Kochmann, Inverse-designed spinodoid metamaterials, npj Comput. Mater. 6 (1) (2020) 73.
[31] J.-H. Bastek, S. Kumar, B. Telgen, R.N. Glaesener, D.M. Kochmann, Inverting the structure–property map of truss metamaterials by deep learning, Proc. Natl. Acad. Sci. 119 (1) (2022) e2111505119.
[32] M. Maurizi, C. Gao, F. Berto, Inverse design of truss lattice materials with superior buckling resistance, npj Comput. Mater. 8 (1) (2022) 247.
[33] B. Deng, A. Zareei, X. Ding, J.C. Weaver, C.H. Rycroft, K. Bertoldi, Inverse design of mechanical metamaterials with target nonlinear response via a neural accelerated evolution strategy, Adv. Mater. 34 (41) (2022) 2206238.
[34] Thermo-calc software TCFE9 steels/Fe-alloys database, https://thermocalc.com/products/databases/steel-and-fe-alloys/.
[35] C.A. Schneider, W.S. Rasband, K.W. Eliceiri, NIH image to ImageJ: 25 years of image analysis, Nat. Methods 9 (7) (2012) 671–675.
[36] R. Quey, P. Dawson, F. Barbe, Large-scale 3d random polycrystals for the finite element method: generation, meshing and remeshing, Comput. Methods Appl. Mech. Eng. 200 (17) (2011) 1729–1745, https://doi.org/10.1016/j.cma.2011.01.002, https://www.sciencedirect.com/science/article/pii/S004578251100003X.
[37] R. Rodriguez, I. Gutiérrez, Unified formulation to predict the tensile curves of steels with different microstructures, in: Materials Science Forum, vol. 426, Trans Tech Publications Ltd., Zurich-Uetikon, Switzerland, 2003, pp. 4525–4530.
[38] S. Ma, X. Zhuang, Z. Zhao, Effect of particle size and carbide band on the flow behavior of ferrite–cementite steel, Steel Res. Int. 87 (11) (2016) 1489–1502.
[39] X. Zheng, T.-T. Chen, X. Guo, S. Samitsu, I. Watanabe, Controllable inverse design of auxetic metamaterials using deep learning, Mater. Des. 211 (2021) 110178.