Composites Part B 269 (2024) 111099

Contents lists available at ScienceDirect

Composites Part B

journal homepage: www.elsevier.com/locate/compositesb

Optimization of mechanical properties of multiscale hybrid polymer nanocomposites: A combination of experimental and machine learning techniques

Elizabeth Champa-Bujaico [a], Ana M. Díez-Pascual [b,*], Alba Lomas Redondo [a], Pilar Garcia-Diaz [a]

[a] Universidad de Alcalá, Departamento de Teoría de la Señal y Comunicaciones, Ctra. Madrid-Barcelona Km. 33.6, 28805 Alcalá de Henares, Madrid, Spain
[b] Universidad de Alcalá, Facultad de Ciencias, Departamento de Química Analítica, Química Física e Ingeniería Química, Ctra. Madrid-Barcelona Km. 33.6, 28805 Alcalá de Henares, Madrid, Spain

ARTICLE INFO

Handling Editor: Prof. Ole Thomsen

Keywords:
Machine learning
Multiscale polymer nanocomposites
Mechanical properties
Poly(3-hydroxybutyrate) (P3HB)
Optimization
Predictive algorithms

ABSTRACT

Machine learning (ML) models provide fast and accurate predictions of material properties at a low computational cost. Herein, the mechanical properties of multiscale poly(3-hydroxybutyrate) (P3HB)-based nanocomposites reinforced with different concentrations of multiwalled carbon nanotubes (MWCNTs), WS_2 nanosheets and sepiolite (SEP) nanoclay have been predicted. The nanocomposites were prepared via solution casting. SEM images revealed that the three nanofillers were homogenously and randomly dispersed into the matrix. A synergistic reinforcement effect was attained, resulting in an unprecedented stiffness improvement of 132% upon addition of 1:2:2 wt% SEP:MWCNTs:WS_2. Conversely, the increments in strength were only moderates (up to 13.4%). A beneficial effect in the matrix ductility was also found due to the presence of both nanofillers. Four ML approaches, Recurrent Neural Network (RNN), RNN with Levenberg's algorithm (RNN-LV), decision tree (DT) and Random Forest (RF), were applied. The correlation coefficient (R^2), mean absolute error (MAE) and mean square error (MSE) were used as statistical indicators to compare their performance. The best-performing model for the Young's modulus was RNN-LV with 3 hidden layers and 50 neurons in each layer, while for the tensile strength was the RF model using a combination of 100 estimators and a maximum depth of 100. An RNN model with 3 hidden layers was the most suitable to predict the elongation at break and impact strength, with 90 and 50 neurons in each layer, respectively. The highest correlation (R^2 of 1 and 0.9203 for the training and test set, respectively) and the smallest errors (MSE of 0.13 and MAE of 0.31) were obtained for the prediction of the elongation at break. The developed models represent a powerful tool for the optimization of the mechanical properties in multiscale hybrid polymer nanocomposites, saving time and resources in the experimental characterization process.

1. Introduction

Artificial Intelligence (AI) is one of the branches of computer science that has aroused most interest today, due to its potential application in numerous and diverse fields [1]. AI as a field of study, has its roots in the pioneering work of British mathematician Alan Turing in the 1950 [2]. Turing is best known for his influential paper entitled "Computing Machinery and Intelligence" published in 1950. In this article, Turing proposed the famous Turing Test, a test designed to assess the ability of a machine to exhibit intelligent behavior like or indistinguishable from human behaviour [3]. There are two main categories within AI, strong AI, and weak AI [4]. The former is an advanced form of AI regarded as conscious and intentional in a similar way to humans. This type of AI would be able to reason, have emotions and maintain self-perception [5]. However, so far, the existence of a fully developed strong AI is in the realm of speculation and has not yet been achieved. Weak AI, on the other hand, lacks consciousness and intentionality. It is designed to perform specific and limited tasks, based on predefined algorithms and rules. Although weak AI can be highly efficient and outperform humans in specific tasks, it lacks true understanding and awareness of its actions [6]. Both types of AI have important applications in various fields and are the subject of research and development.

AI is constantly evolving and integrating more and more new developments. Machine Learning (ML) is one of the types of AI that is

* Corresponding author.
E-mail address: am.diez@uah.es (A.M. Díez-Pascual).

https://doi.org/10.1016/j.compositesb.2023.111099
Received 26 June 2023; Received in revised form 31 October 2023; Accepted 5 November 2023
Available online 17 November 2023
1359-8368/© 2023 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

currently on the rise. It allows machines to learn and improve automatically through experience and data [7], and includes three main subtypes, as depicted in Fig. 1: a) Supervised Learning: The algorithm learns from labelled examples, where it is provided with a training data set with desired inputs and outputs. It can then be used to predict the outputs on new data. Two common techniques are regression and classification. b) Unsupervised Learning: The algorithm learns from unlabeled data and looks for hidden patterns or structures in it. No desired outputs are provided, and the algorithm must discover for itself the relationships and groupings in the data. Two prominent techniques are clustering and reduction [8]. c) Reinforcement Learning: This is a combination of supervised and unsupervised machine learning. The algorithm learns through interaction with an environment. It is provided with feedback in the form of rewards or punishments based on its actions. The goal is to maximize the cumulative reward over time.

ML can be used at various stages of the development and characterization, from material design to property prediction. It contributes to the acceleration of the research and development process, as well as to the understanding and optimization of properties and processes related to polymer nanocomposites [8,9]. The use of ML to predict properties of polymer nanocomposites based on their composition and structure is a promising approach in materials science. Composition refers to the proportion and type of components present, such as the base polymer and nanomaterials. Structure refers to the organization and distribution of these components, as well as the morphological characteristics at different scales. Training ML models from data sets learns the relationships between the composition, structure and properties of polymer nanocomposites. These datasets can include information on different combinations of polymers, nanomaterials, and associated properties [10,11]. By training ML algorithms with this data, different models can be developed that are able to accurately predict properties of new polymer nanocomposites.

ML models can use different approaches, such as regression, classification, or deep learning, depending on the nature and the complexity of the data. These models can be used to predict mechanical, thermal, optical, electrical, or other properties, depending on the specific composition and structure of the polymer nanocomposite [12]. For instance, Özkan et al. [13] implemented artificial neural network (ANN), random forest (RF) and multiple linear regression (MLR) methods to predict mechanical properties of three-component nanocomposite films. They used mechanical properties of two-component nanocomposite films as input to the prediction system. The study showed that ANN had the best accuracy, closely followed by RF, while MLR had the lowest accuracy. Sahu et al. [14] explored the mechanical properties of high-density polyethylene (HDPE)-based nanodiamond nanocomposites using an ANN. The developed model predicted the Young's modulus and hardness of the nanocomposite with a correlation coefficient higher than 0.99, indicating high accuracy in the predictions. Natrayan et al. [15] used an ANN for the prediction of the flexural strength and laminar shear strength of epoxy-based nanocomposites reinforced with graphene. They used a feed-forward training algorithm and were able to generate forecasts with a reliability rate of 95% and an absolute error rate of less than 1%. These results suggest that ANN is effective for estimating mechanical properties of polymeric nanocomposites. Khan et al. [16] used advanced machine learning approaches, such as ANN, to predict the effect of haloisite clay composition and temperature on the mechanical properties of polymer clay nanocomposites (PCNs). The ANN model developed showed an average relative error of 0.0701, indicating good prediction accuracy. This prediction approach can reduce the need for numerous laboratory tests to develop PCNs with the desired properties. In another study by Adel et al. [17], the compressive and flexural strength of carbon nanotube (CNT)-reinforced cementitious nanocomposites were described using various machine learning techniques, including decision tree (DT). For compressive strength, this model was able to explain 98.2% of the variability in the training data and performed well in predicting new data with 86.9% accuracy. On the other hand, for bending strength, the model was able to explain 92.7% of the variability in the training data and performed acceptably in predicting new data with 78.2% accuracy. The abovementioned studies hint that ML methods, such as ANN, DT and RF, can be effective in predicting the mechanical properties of nanocomposites. However, studies on this topic are still scarce, and to the best of our knowledge, no previous work on the prediction of the mechanical properties of multiscale hybrid polymeric nanocomposites has been reported to date.

Polyhydroxyalkanoates (PHAs) are biopolyesters, stored within cells as energy storage materials by various microorganisms. Owed to their biocompatibility and biodegradability, PHAs have a wide range of

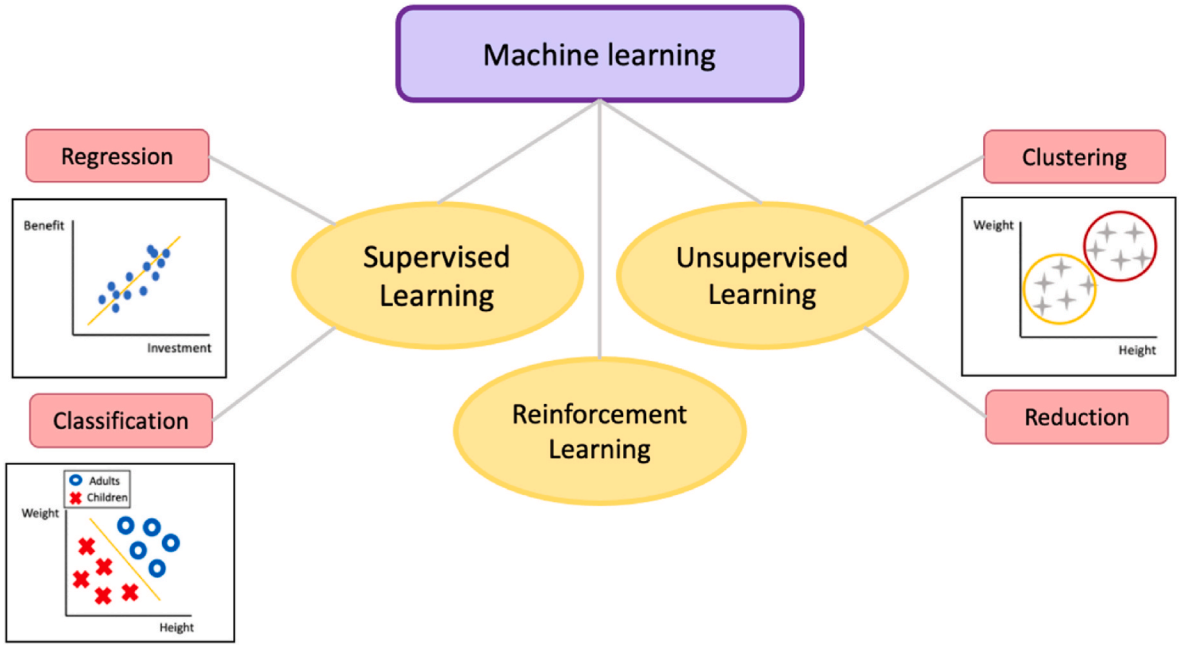


Fig. 1. Main types of Machine Learning: Supervised, Unsupervised and Reinforcement Learning and their most common techniques.

applications in food packaging [18] and the biomedical sector including tissue engineering, bio-implant patches, drug delivery, surgery and wound dressing. PHAs are green plastics and have positive social and environmental impact when compared with conventional plastics in terms of production and recycling [19]. The simplest and most investigated PAH is poly(3-hydroxybutyrate) (P3HB), that possess properties comparable to those of conventional synthetic plastics like polypropylene (PP). However, PHB presents several weaknesses including low impact resistance, poor thermal stability and high water vapor permeability [20]. To address these problems, it can be reinforced with nanomaterials such as multiwalled-carbon nanotubes (MWCNTs) and inorganic nanoclays or nanoparticles [21,22].

Carbon nanotubes (CNTs) are an allotrope of carbon with a 1D nanotubular structure that can be classified as single-walled, double-walled, and multi-walled (SWCNT, DWCTN and MWCNT, respectively). The electrical conductivity of CNTs ranges from metallic to semiconducting based on the diameter and the rolling angle, which also impart chirality in the tubes. CNTs have excellent chemical, mechanical, and electronic properties that make them promising for various applications [23,24]. Both noncovalent and covalent functionalization of CNTs with aromatic molecules and π-conjugated polymers have been extensively investigated [25–27]. These modifications improve their dispersion and solution processability, rendering them suitable for device applications.

On the other hand, nanoclays are nanoparticles of layered mineral silicates that possess outstanding mechanical properties. They provide composites with improved properties owed to their stability, swelling capacity, interlayer spacing, elevated hydration and high chemical reactivity. One of the most used is sepiolite (SEP), a one-dimensional fibrous silicate mineral with octahedral sheets in a continuous arrangement and an ideal formula of $Mg_8Si_{12}O_{30}(OH)_4(OH_2)_4.8H_2O$ [28]. SEP has a very high specific surface area (about 200 ~ 300 m^2/g), and has been extensively used as reinforcement to improve the mechanical properties of polymer matrices [29].

2D Transition Metal Dichalcogenides (TMDCs) incorporating transition metals such as Ti, Mo, W etc. crystallize into layered structure, thereby offering diverse material properties from that of an insulator to a metal. The strong intra-layer covalent bond between the chalcogenides and weak inter-layer Van der Waal attraction between the layers facilitates the transition of these materials from 3D to 2D, analogous to graphene [30]. Among the semiconducting TMDCs, WS_2 has brought a lot of attention owing to unique features such as higher stability and wider operational temperature. They have also been widely used as reinforcement of polymeric matrices, giving significantly enhanced properties, with an effect comparable to that of graphene or CNTs [31,32].

To further improve the properties of polymeric materials, different nanofillers can be incorporated. Hybridization involving the combination of two or more nanofillers in a polymer matrix results in reduction of water absorption properties and improved mechanical properties due to synergistic effects. For instance, P3HB/graphene/carbon nanofiber hybrids have already been developed as sustainable materials with enhanced mechanical properties [21]. In the present study, multiscale hybrid nanocomposites based on a P3HB matrix incorporating different amounts of MWCNTs, SEP and WS_2 nanosheets have been developed by an easy, cheap, and ecological solvent casting method to enhance the mechanical properties of P3HB matrix. By tuning the percentages of both nanofillers, the stiffness and strength can be carefully tailored to attain specific mechanical properties for targeted applications. Further, three ML-based algorithms (RNN, DT and RF) have been applied to predict their mechanical properties. The basis of the models are described in detail in the next section, and their minimum squared errors (MSE), mean absolute errors (MAE) and coefficient of determination (R^2) are compared to determine the optimum model for each of the studied properties.

2. Methodology

In this study, three types of regression models have been selected to predict the mechanical properties of the hybrid nanocomposites: RNN, DT and RF. These models are widely used in scientific research due to their ability to capture non-linear relationships between input and output variables, their flexibility, their high predictive accuracy, and their ability to learn complex patterns in the data. The main characteristics of each model are detailed below.

2.1. Artificial neural network (ANN)

ANN is a type of computational model inspired by the structure and functioning of the human brain, with the purpose of establishing the underlying relationships in a data set [33]. A set of interconnected nodes, called artificial neurons, are constructed, and organized into layers that process information. Layers are structured in three types: the input layer, the output layer and one or more hidden layers. Each neuron has an activation function that introduces non-linearity into the model. Neurons in the input layer receive input data, those in the hidden layers perform computations and transformations, and those in the output layer generate the final outputs of the network [34,35]. The neurons are connected via weighted connections, which determine the relative importance of the corresponding input in calculating the output of a unit. These weights are adjusted during the training process to optimize the performance and predictions of the network [35,36]. The basic structure of a neural network can vary depending on the problem and the specific architecture used [37]. Some of the most common structures are: a) Perceptron, which consists of a single hidden layer of units, where each neuron is connected to the input layer and produces an output based on an activation function. It is suitable for linearly separable binary classification problems [38]. b) Neuronal networks Feedforward [39], also known as multilayer neural networks, wherein information flows in a single direction, from the input layer to the output layer, without cycles. They are suitable for complex classification and regression problems and can learn feature representations. c) Convolutional neural networks (CNN), which use convolutional layers to extract spatial features and reduce the dimensionality of the data. They are very powerful in image recognition and classification [40]. d) Recurrent Neural Networks (RNNs), which have recurrent connections, meaning that outputs from a layer can feed back to themselves or to previous layers, allowing previous layers to benefit from information generated in later layers and to hold such information over time. RNNs are highly suitable for processing sequence and time series data, such as text and audio [41].

In this study, input data representing the amount of each filler incorporated into the composite, and output data corresponding to the specific mechanical properties have been used. Each system is composed of three input variables, which are related to the weight percentage of each nanofiller (MWCNT, SEP and WS_2), and one output variable representing a mechanical property, such as Young's modulus, tensile strength, elongation at break, or impact strength. These mechanical properties have been studied in independent networks. The aim is to use ML to establish relationships and predict the mechanical properties of polymer nanocomposites as a function of their chemical composition.

Two optimization techniques in the RNN are compared: optimization by stochastic gradient descent and the Levenberg-Marquard optimization algorithm [42,43]. The former utilizes a standard backpropagation technique to compute the gradients and adjust the network parameters during the training process, whereas the Levenberg-Marquardg algorithm (RNN-LV) employs the Gauss-Newton technique to minimize the error of the objective function, often used in nonlinear regression problems. It is important to note that, in this case, backpropagation is still an integral part of the process, as it is used to calculate the local gradients needed for the Levenberg-Marquardt algorithm. The main objective of this comparison is to evaluate the performance of each of

these approaches in terms of accuracy and training efficiency to predict the properties under study. This will allow us to determine which of the training methods best suits the objectives of our research and provide a deeper understanding of how they influence the results obtained.

The network architecture used is a Feedforward neural network [44], which makes predictions based on the chemical composition of the nanocomposites. In this study, the neural network consists of an input layer with three neurons, which represent the three input variables, hidden layers whose number can vary depending on the problem being addressed, and finally an output layer with a single neuron for the output variable. The hidden layer uses the ReLU activation function, which is a non-linear function commonly used in ANN [45]. As for the parameters of the training algorithm, an initial learning rate of 0.01 for RNN and 0.09 for RNN-LV and a maximum number of iterations of 1000 have been set. These values can be adjusted according to the problem and the network architecture. The maximum number of iterations is set to avoid over-fitting the model and to control the length of the training process. It is important to correctly tune the model parameters, which may require a trial-and-error approach for each specific problem.

The data is divided into training and test sets to evaluate the accuracy and performance of the model. To implement the neural network, the TensorFlow functional API is used, which allows building complex and flexible models for regression tasks. The model implementation has been carried out in Python v.3.9.6 using the Keras v.2.10.0 and TensorFlow v.2.10.0 libraries. These tools provide an efficient and powerful environment for developing and training neural networks effectively.

2.2. Decision tree (DT)

DTs are a type of machine learning models that make decisions by following a set of rules. In DTs, these questions are based on features or attributes of the data. Each node in the tree represents a question, and the branches are the possible answers to that question [46]. You keep going through the tree until you reach a leaf that contains the prediction or result (Fig. 2). These DTs are easy to understand and visualise, which makes them useful in classification and prediction problems [47].

In regression DTs, data are split according to characteristic values to predict a target-continuous variable [48]. The regression DT follows the same structure as a conventional decision tree, where each internal node represents a feature, and each branch represents a possible choice or value of that feature.

The construction of a regression DT involves selecting the best feature and the best split point to minimize the variance of the values of the target variable within each resulting subset. The goal is to obtain partitions that reduce variability and are as close as possible to the actual values of the target variable. Once the regression DT is constructed, it can be used to make predictions on new data instances. Following the decision paths based on the input features, one arrives at a sheet containing the predicted value for the target variable [49].

Hyperparameters are adjustable settings that control the behavior and structure of the tree during the training process. Among the most common are: a) Maximum tree depth, which controls the maximum number of splits the tree can have. Greater depth allows more complex relationships to be captured but may increase the risk of overfitting. b) Criterion, which specifies the metric used to measure the quality of a split in the tree nodes. The most common criterion is the mean squared error (MSE), which measures the discrepancy between predictions and actual values c) Partitioning criterion, which refers to the strategy used to choose how to split the nodes in a decision tree. d) Minimum samples to split, which indicates the minimum number of samples needed at a node for a split to be performed. e) Minimum samples per leaf, which establishes the minimum number of examples required for a node to be considered a leaf, i.e., a terminal in the decision tree. f) Maximum leaf nodes, which determine the maximum limit of leaf nodes that a decision tree can have. g) Maximum features, which represents the maximum number of prediction variables that are considered when performing a split at each node in the DT [50]. Hyperparameters are set prior to the training process and affect the structure and behavior of the tree [51]. It is important to properly tune the hyperparameters to optimize the performance of a regression DT.

2.3. Random forest (RF)

It is an ensemble technique that combines the simplicity and interpretability of DTs with the power of combining multiple models [52,53]. A set of independent DTs is constructed using different subsets of training data and random features [54]. Each tree is trained individually, dividing the data according to specific characteristics and creating a set of rules to make predictions (Fig. 3).

When a prediction is made with the RF, each tree issues its own prediction and either takes most of the votes or averages the result, depending on whether it is a classification or regression problem, respectively. The key idea behind RFs is that by combining multiple TDs, bias and variance are reduced, which can improve the accuracy and generalizability of the model [52]. In addition, RFs have the advantage of being able to handle large datasets and correlated features. The most common hyperparameters in these algorithms are: a) n estimators, which indicates the number of decision trees to be used in the model assembly. b) minimum samples to split, which sets the minimum number of samples required for a node to be split during the construction of each decision tree. c) maximum tree depth, which sets the maximum depth allowed for each decision tree in the assembly. It refers to the maximum path length from the root to a leaf. d) Bootstrap, which determines whether Bootstrap sampling is used when constructing the decision trees in the assembly. e) Warm start, which indicates whether to use the previously fitted trees in the model when refitting the estimator.

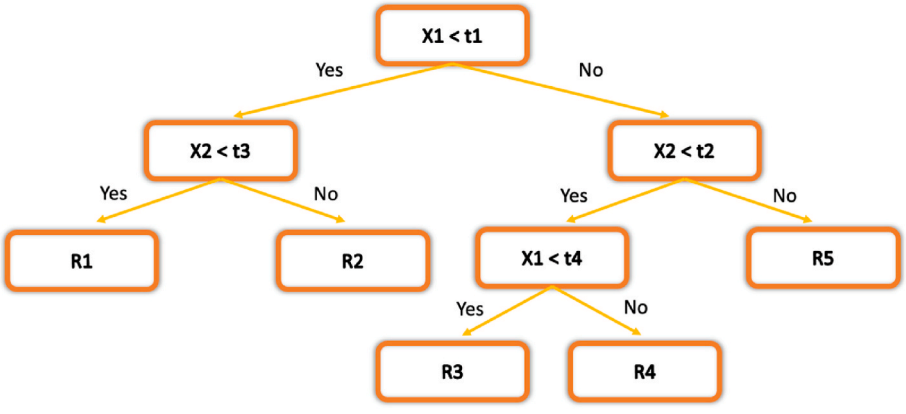


Fig. 2. Representation of the regression decision tree model: Visualising decision making.

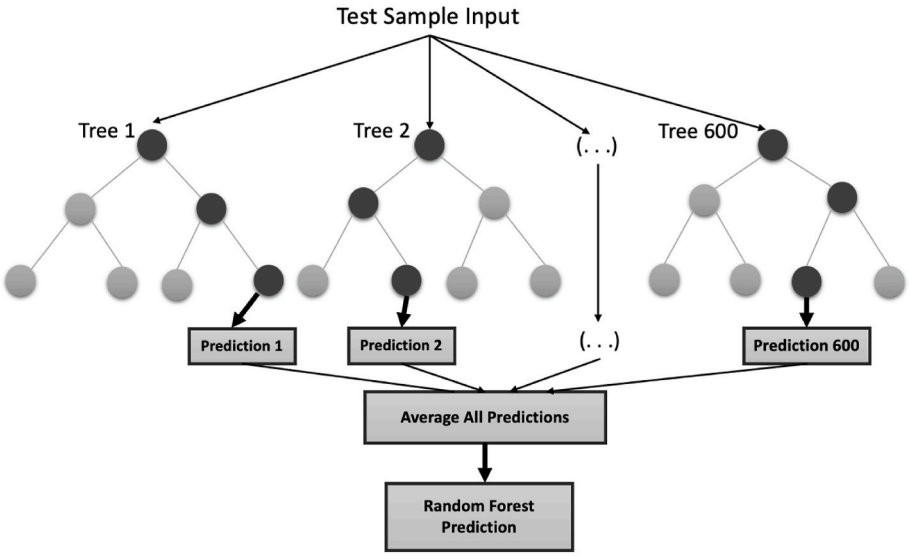


Fig. 3. Representation of the Random Forest: an ensemble learning approach.

For both DT and RF, it is necessary to adjust the hyperparameters of the model to obtain optimal performance. Here is where search techniques such as GridSearchCV and RandomSearchCV come into play [55]. These techniques are used to explore the hyperparameter space and find the optimal combination that maximizes model performance. GridSearchCV performs an exhaustive search in a predefined hyperparameter space. A grid of possible values is specified for each hyperparameter, and the performance of the model is evaluated for each combination. The former performs a cross validation on all possible combinations and returns the configuration with the best performance [56], while the other randomly selects a given set of combinations and evaluates their performance [57]. By randomly exploring the hyperparameter space, it is possible to find optimal configurations more efficiently [58], especially when the search space is large. Both search techniques allow to systematically explore the hyperparameter space of the regression DT and find the optimal combination that maximizes the model performance, thereby improving its predictive capability. Both DT and RF were implemented using the Python V.3.9.6 programming language and the Scikit-learn v.0.23.2 library.

The mean absolute error (MAE), the mean square error (MSE), and the coefficient of determination (R^2) were the statistical indicators used to assess the performance of the developed models. MSE is calculated as the mean of the squares of the differences between the predicted values and the actual values. It is defined as follows:

$$MSE = \frac{1}{n}\sum (y' - y)^2$$

where y' is the predicted value, y is the corresponding value and n is the total number of samples in the data set.

MSE penalizes large errors more than small errors due to the squaring operation, which can make the model more sensitive to outliers or grossly erroneous predictions.

MAE is calculated as the average of the absolute differences between the predicted and actual values:

$$MAE = \frac{1}{n}\sum |y' - y|$$

MAE is less sensitive to outliers than MSE as it uses absolute values instead of squares. However, by not squaring errors, MAE may underestimate the impact of large errors.

R^2, is the proportion of the variation in the dependent variable that is predictable from the independent variable(s), and is calculated as:

$$R^2 = \left(\frac{n(\sum yy') - (\sum y)(\sum y')}{\sqrt{[n\sum y^2 - (\sum y)^2][n\sum y'^2 - (\sum y')^2]}} \right)^2$$

R^2 is a measure of goodness of fit that is commonly used to assess the accuracy of ML models. It represents the degree of correlation between the model prediction and the target values. The accuracy of a model improves as R^2 approaches to 1.0.

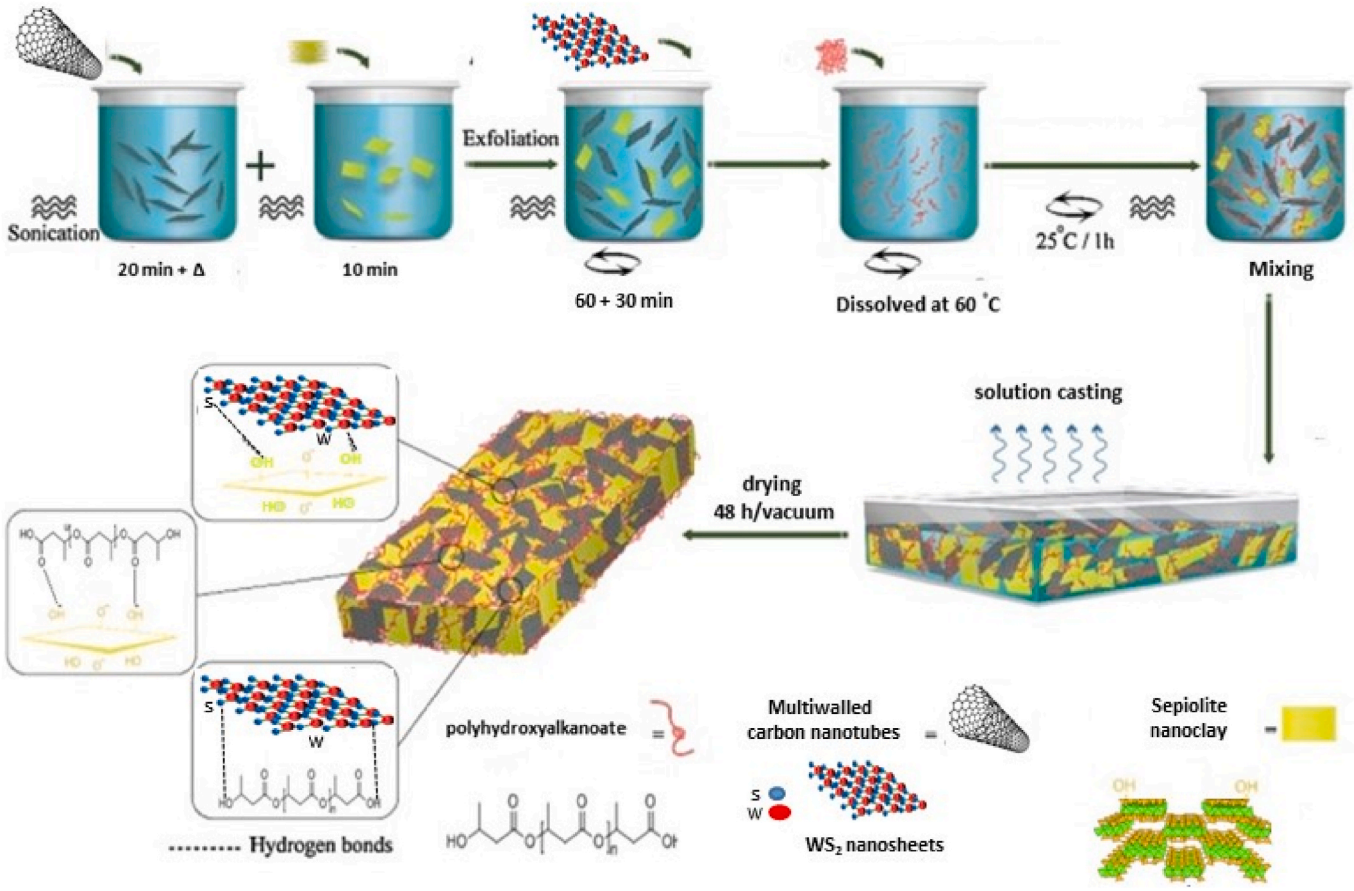
3. Experimental

3.1. Materials

Poly(3-hydroxybutyrate) (P3HB) with a $M_W \sim 80.000$ g/moL; $T_g \sim 20$ °C, $T_m \sim 175$ °C, $d_{25\,°C} = 1.25$ g/cm^3 was purchased from Biomer Ltd. (Krailling, Germany) and dried in an oven at 80 °C overnight before use. Multiwalled carbon nanotubes (MWCNTs) were provided by Nanothinx (Rio Patras, Greece). Their outer average diameter was 20–40 nm, and length ranged between 0.5 and 2 μm, with a purity of 97%. WS_2 nanosheets prepared by lithium intercalation were obtained from XFNANO Materials Tech Co., Ltd. (Nanjing, China). Sepiolite powder $Mg_2H_2(SiO_3)_3.xH_2O$ (95%, 220–270 m^2/g) was provided by Sigma-Aldrich (Madrid, Spain). All chemicals and solvents were used as received.

3.2. Nanocomposite preparation

The nanocomposites were synthesized via solution casting, following a multi-step process, as depicted in Scheme 1. Initially, the necessary amount of MWCNTs were dispersed in chloroform via bath sonication for 20 min. Separately, the required amount of montmorillonite powder was dispersed in water by sonication for 10 min and then added to the MWCNT dispersion, which was sonicated for another 60 min. Then, the bulk WS_2 nanosheets were added and sonicated once again for 30 min to attain an exfoliated mixture with few layered WS_2 nanosheets. Separately, the PHB powder was dissolved in chloroform at 60 °C and then added to the MWCNT/nanoclay/WS_2 dispersion and the mixture was then sonicated for another 60 min at 25 °C. The hybrid mixture was then cast onto a glass Petri dish and finally dried under vacuum for 48 h. A schematic representation of the synthesis process and the potential interactions among the nanocomposite components is shown in Scheme 1.



Scheme 1. Representation of the synthesis of the nanocomposites and the interactions between them.

3.3. Characterization techniques

All the samples were conditioned for 24 h before the measurements. Tensile tests were carried out following the ASTM D 638-03 standard on a servo-hydraulic testing machine (858 Mini Bionix, MTS Systems Corporation, MN, US) at a crosshead speed of 1 mm/min and a load cell of 100 kN, under 23 °C and 50% RH.

Charpy impact strength tests were carried out according to the ASTM D 6110-10 standard under the same environmental conditions on a CEAST Fractovis dart impact tester (Instron, MA, US). Notched specimen bars and a hammer mass with an energy of 7.10 J were used. Five specimens for each type of nanocomposite were measured to check for repeatability, and average value is reported.

The surface morphology of the nanocomposites was examined by scanning electron microscopy (SEM) using a scanning electron microscope (Zeiss, SIGMA VP-500, Germany), at an acceleration voltage of 25 kV. Samples were first cryo-fractured and then sputtered with a gold layer under vacuum to avoid charging during electron irradiation.

4. Results and discussion

4.1. Surface morphology of PHB/MWCNT/nanoclay/WS_2 composites

The morphology of the hybrid PHB-based nanocomposites was investigated by SEM, and typical images at different magnifications of the cross section of the sample with 2 wt% MWCNT, 1 wt% nanoclay and 2 wt% WS_2 are shown in Fig. 4. Similar micrographs were observed for the other hybrid composites. The nanocomposites show a 3D structure

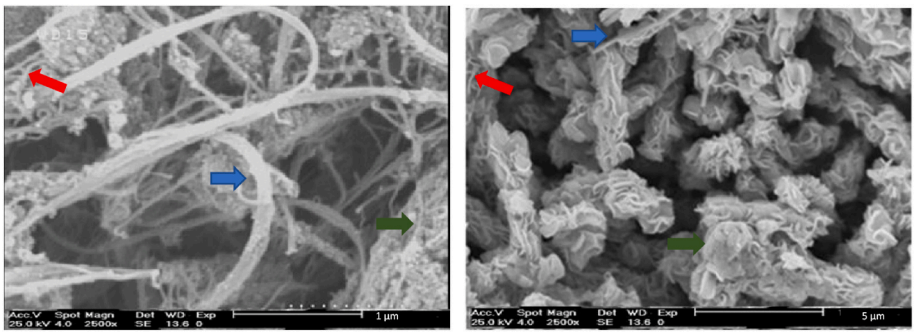


Fig. 4. SEM images from fractured surfaces of PHB/MWCNT(2 wt%)/nanoclay (1 wt%)/WS_2 (2 wt%) nanocomposite at different magnifications. The red, blue and green arrows point towards the MWCNT, the nanoclay and the WS_2, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

made of wrinkled WS_2 nanoflakes, curled MWCNTs and sepiolite nanofibers embedded into the continuous semicrystalline phase of PHB. The three nanomaterials appear intercalated. The nanoclay fibers are commonly forming aggregates of tangled fibers with diameters in the range of 150–300 nm. On the other hand, WS_2 nanosheets stacked compactly in the form of irregular plates can be observed, which are uniformly decorating the surface of sepiolite nanofibers and the MWCNTs, leading to a compact flower-like structure. The presence of sepiolite reduces the agglomeration of the WS_2 nanosheet, leading to well exfoliated flakes with thicknesses in the range of 10–50 nm.

In the images at lower magnification, the three nanofillers are found to be homogenously and randomly dispersed into the matrix, forming a dense and entangled network, like due to the hydrogen bonding interactions between the three nanocomposite components (see Scheme 1). Further, other interactions such as polar and hydrophobic can contribute to the formation of a reinforcement network strongly adhered to the matrix.

4.2. Experimental mechanical properties of the nanocomposites

Table 1 gathers the results of the tensile and Charpy notched impact strength tests while Fig. 5 plots the data as a function of the total nanofiller loading.

The pure polymer has a stiffness of approximately 3.7 GPa (Young's modulus, E). However, upon addition of 5 wt% nanofiller loading, such as SEP, MWCNT or WS_2, the stiffness increases by about 80%, 100% and 68% respectively (Fig. 5A). Proper dispersion of nanofillers in the polymer matrix is crucial to achieve a significant stiffening effect. In this case, the uniform dispersion of the nanoclay, CNT or WS_2 in the polymer allows for efficient load transfer and reinforcement of the structure. As a result, a highly effective reinforcement effect is obtained in the binary nanocomposites, which is reflected in a linear increase of the stiffness within the range of nanofiller concentration studied. The improvements in mechanical properties found in this study are considerably larger than those previously reported for nanocomposites made of PHBV and organomodified nanoclay [59] or MWCNTs fabricated by the melt blending technique. This significant difference could be attributed to the partial degradation of the polymer that occurs during the fabrication process. It is worthy to note that for the same nanofiller loading, the strongest reinforcement effect is systematically attained for 1D MWCNTs compared to 1D fibrous SEP or 2D WS_2 nanosheets. Given that both SEP and MWCNTs have fiber-like shape, the more effective reinforcement is attributed to the stronger E of MWCNTs [60] (i.e. 300–950 GPa) compared to that of SEP or WS_2 (250 GPa) [61]. It is noticeable that the reinforcement effect of SEP is also ascribed to the formation of H-bonds between the surface silanol groups of the nanoclay and the ester groups of the matrix (Scheme 1). This combined with polar interactions leads to a more compact structure, thus restricting the movement of the polymeric chains, which in turn is reflected in enhanced rigidity, in agreement with results reported previously for SEP-reinforced (alkyl) methacrylate-based composites [62].

Regarding the ternary nanocomposites, a significant increase in modulus can also be observed (Fig. 5A), clearly higher than that of the binary samples for the same total nanofiller loading. The strongest rise (~92%) is detected upon addition of 1 wt% SEP and 4 wt% MWCNTs. This extraordinary increase is again attributed to the high modulus of MWCNTs and their homogeneous dispersion within the matrix. Similar trend, even more pronounced is found in the quaternary nanocomposites (Table 4). Thus, the addition of 1 wt% of each nanofillers results in about 45% rise in stiffness. More importantly, upon addition of 1:2:2 wt% SEP:MWCNTs:WS_2, an increase of 132% is found, and the strongest increase (154%) is attained upon addition of 2:2:1 wt% of the indicated nanofillers. This suggests again the stronger reinforcement of fibrous 1D nanomaterials, irrespective of their nature, highlighting the significant contribution of CNTs in improving the mechanical properties of the matrix. Besides, this indicates a synergistic effect of the three

Table 1
Relationship between Mechanical Properties and Nanofiller Concentration in PHB/MWCNT/nanoclay/WS_2 multiscale nanocomposites.

Sample	%Nanoclay	%CNT	%WS_2	Young Modulus (Gpa)	Tensile Strength (MPa)	Strain at break (%)	Impact strength (J m^{-1})
1	0	0	0	3.71	38.46	5.78	40.94
2	1	0	0	4.29	39.67	5.25	44.34
3	2	0	0	5.18	40.06	3.96	30.13
4	3	0	0	5.99	40.25	3.48	26.83
5	5	0	0	6.66	39.97	2.88	19.76
6	0	1	0	4.13	38.72	5.47	39.03
7	0	2	0	5.39	38.96	5.12	35.96
8	0	3	0	6.48	37.11	3.89	28.65
9	0	5	0	7.43	35.44	2.76	21.74
10	0	0	1	4.06	38.72	5.22	37.26
11	0	0	2	4.97	40.01	3.84	28.43
12	0	0	3	5.76	40.11	3.51	25.83
13	0	0	5	6.24	39.66	2.97	21.97
14	1	1	0	4.72	39.89	4.16	30.11
15	1	2	0	6.23	40.07	3.78	27.88
16	1	4	0	7.11	37.76	3.16	22.13
17	0	1	1	4.93	39.95	4.02	29.89
18	0	1	2	5.98	39.86	3.98	29.41
19	0	1	4	6.55	38.34	3.49	24.67
20	1	0	1	4.45	40.19	5.66	42.05
21	1	0	2	5.37	41.31	5.33	41.51
22	1	0	4	6.28	41.12	4.99	37.98
23	2	1	0	6.12	39.13	3.56	25.69
24	4	1	0	6.79	38.78	3.89	25.17
25	2	0	1	5.56	40.36	5.03	37.34
26	4	0	1	6.09	41.34	4.88	37.15
27	1	1	1	5.38	39.17	3.16	21.56
28	2	1	1	7.23	36.84	2.43	17.86
29	1	2	1	7.77	40.56	2.71	20
30	1	1	2	6.82	37.65	2.81	19.66
31	2	2	1	9.44	43.62	2.33	19.29
32	2	1	2	8.36	38.01	2.22	15.55
33	1	2	2	8.63	42.15	2.47	20.03

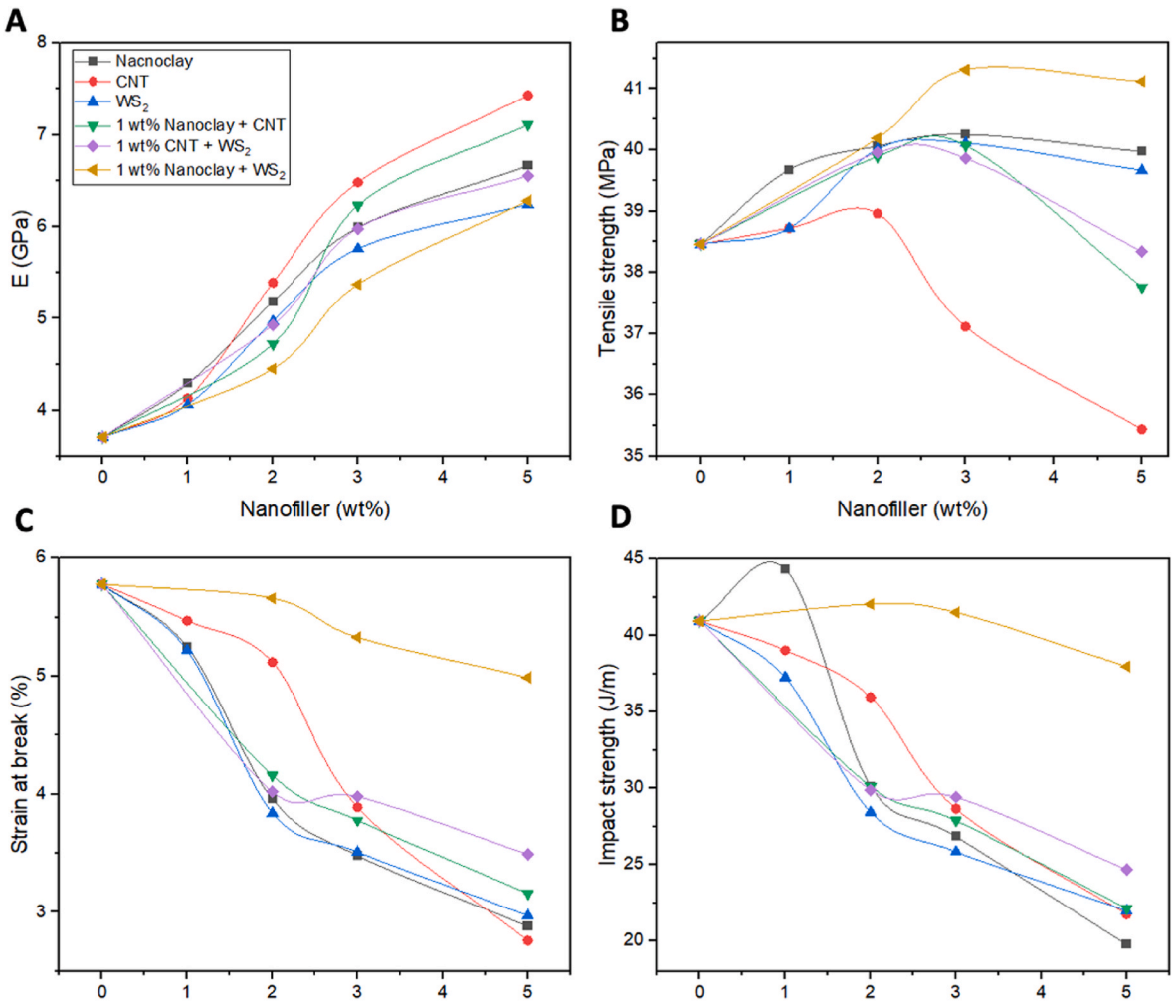


Fig. 5. Experimental mechanical properties of PHB/MWCNT/nanoclay/WS_2 nanocomposite as a function of total nanofiller loading. Comparison of binary and ternary nanocomposites.

nanofillers, resulting in an unprecedented stiffness improvement. Thus, the combination of the three nanofillers has a positive and additive effect on enhancing the elastic modulus of the nanocomposite.

Focusing on the tensile strength, the trend observed is similar to that described above, although the improvements are significantly smaller. Thus, the largest rise (13.4%) is again found for the nanocomposite with 2:2:1 wt% of SEP:MWCNTs:WS_2. The smaller strength enhancement compared to the modulus could also be related to the lower strength of the nanofillers (i.e. 10–60 GPa for MWCNTs [60], 3–7 GPa for WS_2 [61], and 100–500 MPa for SEP). Further, the orientation and state of dispersion of the nanofillers have been reported to influence more the composite stiffness than the strength. However, different results were reported for multiscale epoxy nanocomposites filled with graphene and amine-functionalized MWCNTs, in which the increase in tensile strength (52%) was about double that of the modulus. The high strengthening efficiency found in that work was attributed to a synergistic behavior of both carbon nanomaterials well dispersed and embedded within the matrix [63]. This suggests that the main role of the nanofillers is a strengthening of the fiber-matrix interface rather than reinforcing the matrix itself. An opposite behavior is observed for the impact strength (Fig. 5b) and strain at break (Fig. 5c), that drop with increasing nanofiller loading. This is the expected trend since the nanofillers restrict the ductile flow of the polymeric chains [17]. In addition, they can limit the shear deformation of the matrix, thus reducing the ductility. Interestingly, for the same total nanofiller loading, the reduction in ductility found in the ternary nanocomposites is less pronounced than for the binary ones (Fig. 5c), indicating also a beneficial effect due to the presence of both nanofillers. Similar trend has been previously reported for nanocomposites reinforced with SWCNTs and WS_2 [31]. The presence of both nanofillers could reduce the stress concentrations at the nanofiller-matrix interface, hence resulting in better ductility than in the binary composites. The highest impact strength is attained with 1 wt% SEP, suggesting that the fibrous clays can effectively reduce crack propagation. On the other hand, the impact strength is approximately preserved for ternary composites with SEP and WS_2. It has been shown that numerous factors govern the impact strength of multiscale reinforced composites [63], including a lessening effect of the fillers due to a decrease in the elongation at break and stress concentrations formed around the filler ends. The impact strength is also dependent on the size, orientation and interfacial adhesion with the matrix. A beneficial effect is that the fillers decrease crack propagation and increase impact strength by energy dissipating mechanisms. The combination of fibrous clay and layered WS_2 seems to be the more effective for hindering the growth of cracks.

4.3. *Mechanical properties predicted using ML regression models*

In this research, four different models, namely RNN, RNN-LV, DT and RF were applied to predict the mechanical properties of nanocomposites. These models were selected for their ability to handle nonlinear data, generalize, identify patterns, and improve the accuracy of predictions.

The available dataset was split 75% for training and 25% for testing. During the training stage, it is of crucial to find the best hyperparameter configuration. In the case of neural networks, these hyperparameters are predefined values selected before running the model and have a significant impact on its performance.

The fundamental tuning parameters in RNN and RNN-LV are: the number of hidden layers, the number of neurons in each intermediate layer, the learning rate, the activation function and the optimization algorithm. The number of hidden layers or number of intermediate layers determines the depth of the network. The number of neurons in each layer impacts the complexity and learning capacity of the network at each level. The learning rate adjusts the rate at which the network updates the weights during the training process, affecting the speed of convergence and the stability of the training process. The activation function defines the mathematical function of the neuron output from the input, which provides nonlinearity to the network so that it can learn and model complex relationships. Finally, the optimizer is the algorithm used to adjust the weights of the network during training, progressively approaching the optimal weights that minimize the loss function. For both RNN and RNN-LV, the activation function used in each layer is ReLU, and the learning rate used is equal to 0.09. For the key hyperparameters, the first step is to set the optimal number of hidden layers, and then the performance of the network is evaluated by varying the number of neurons in these layers. DT and RF models typically have more hyperparameters to tune compared to RNNs. GridSearchCV and RandomizedCV were used to find the optimal hyperparameter settings and maximize model performance. These techniques explore various predefined combinations of values for the hyperparameters in the DT and RF models. The aim is to find the ideal combination of hyperparameters that improves the performance of both models.

The choice of the optimal combination of hyperparameters is based on evaluation metrics for predictive models, accuracy being one of the most used. In the following sections, the coefficient of determination obtained for several hyperparameter configurations in the different models will be presented. These metrics will allow to identify which hyperparameter configuration provides the best performance in terms of accuracy for each model.

4.3.1. *Young's modulus*

4.3.1.1. RNN. Fig. 6A and B shows the performance of an RNN as a function of the number of hidden layers and the number of neurons in each layer, respectively. R^2 of the training data increases with increasing the number of hidden layers, Fig. 6A. This means that the model can fit better the training data and capture more complex patterns as hidden layers are added. However, in the case of the test data, R^2 peaks when three hidden layers are used, indicating that the model achieves good generalizability by capturing patterns that apply not only to training data, but also to new data. Interestingly, when a fourth hidden layer is added, the RNN performance on test data decreases, suggesting that the model becomes too complex and begins to over-fit the training data, which hampers its generalizability.Once the optimal performance was set to three hidden layers, further analysis was performed on the number of neurons in those layers. Fig. 6B shows the change in R^2 of the training and test data with the number of neurons. Despite no clear trend is observed, the best performance is attained when 90 neurons are used in each of the three hidden layers, leading to a training and test R^2 of 1 and 0.9564, respectively. Additional neurons beyond this optimal point can lead to over-fitting and deterioration in the generalization ability of the model. Instead of capturing more relevant patterns, the model may be overfit to details and noise present in the training data, resulting in worse performance on unseen data. It is important to note that R^2 values obtained herein are comparable or even higher than those reported by Ho et al. [64] for the prediction of E of polymer/carbon nanotube nanocomposites using an RNN model (0.986 and 0.978 for the training and testing sets, respectively). They are also better than those reported for low density polyethylene (LLDPE)/graphene nanocomposites [65], in which the highest R^2 value for the training set was 0.933 when 13 neurons were used in the hidden layer, and those found for ternary poly (ether ether ketone) PEEK/Ti/graphene nanocomposites, with R^2 of this set equal to 0.96 [66] for 78 neurons.

4.3.1.2. RNN-LV. In the case of an RNN-LV model, Fig. 7A, it is found that as the number of hidden layers in a neural network is changed, the performance of the training data remains almost constant and with high values. However, the performance on test data increases until the network has three hidden layers but decreases when a fourth layer is added. Therefore, a network with three hidden layers was selected. Fig. 7B analyses the effect of the number of neurons in these three hidden layers. The performance of the training data remains high and approximately constant as the number of neurons is increased. However, in the test data, a decrease in performance is noticed as the number of neurons in the hidden layers increases, ascribed to factors such as overfitting. Despite this overall decline, a sweet spot in network performance is highlighted with 50 neurons in each of the three hidden layers. At this point, a balance is achieved between the model's tunability and its generalizability, leading to a training and test R^2 of 0.9974 and 0.9710, respectively. The results obtained in this study are superior to those found in the study by Liu et al. [67], who used an RNN-LV to predict the Young's modulus of graphene-reinforced

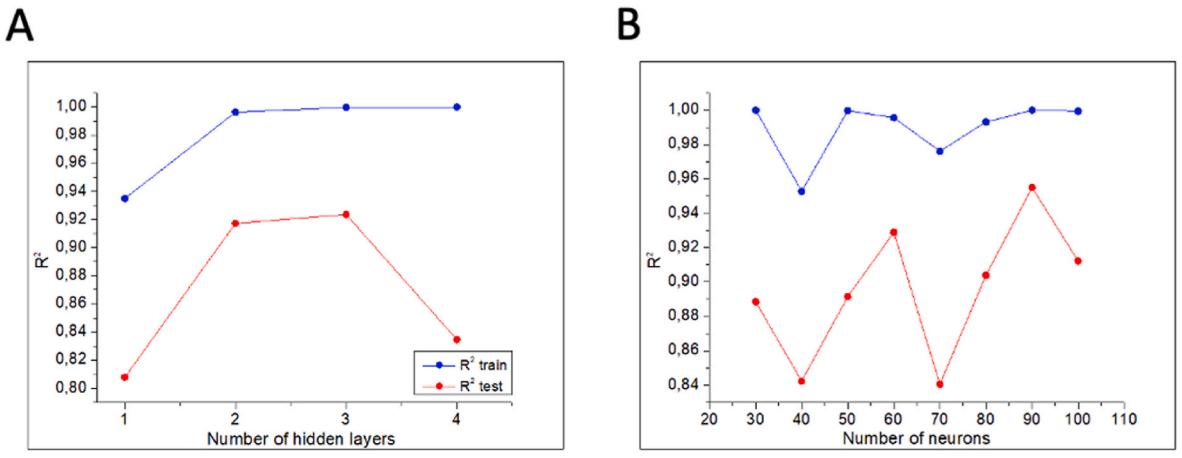


Fig. 6. R^2 of an RNN architecture for the prediction of the Young's modulus as a function of: A) hidden layers and, B) number of neurons of those hidden layers in a functional API RNN.

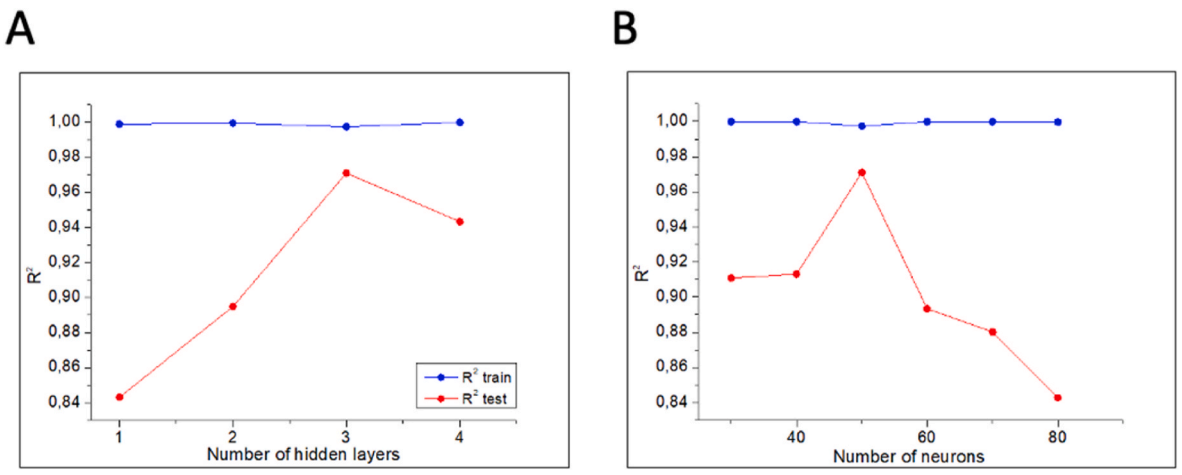


Fig. 7. R^2 of RNN-LV architecture for the prediction of the Young's modulus as a function of A) hidden layers and, B) number of neurons of those hidden layers in an RNN-LV.

aluminum nanocomposites, achieving a R^2 coefficient of 0.951.

4.3.1.3. DT. As mentioned earlier, to evaluate the performance of a DT-based regression model, two hyperparameter search techniques were used: GridSearchCV and RandomizedSearchCV. Before applying these techniques, the ranges of values for the common hyperparameters of the regression DTs were defined (Tables 2 and 3).

The results of the DT using the two techniques are shown in Table 4, where all possible combinations of values of the specified hyperparameters are explored. In this case, the hyperparameters maximum depth and maximum leaf nodes have the strongest impact on the model performance. The values of minimum split samples, minimum leaf samples and maximum features were fixed at 2, 1 and "auto" respectively. As the maximum tree depth increases, the performance on the training data improves. However, in the test data, the performance tends to decrease slightly as the tree depth increases. An optimal behavior is found for a maximum leaf node equal to 19 and values of maximum depth of 20 or 12, both in the training and test data.

On the other hand, when using RandomizedSearchCV, the extreme values of both the minimum and maximum for maximum leaf nodes show superior performance on both training and test data, indicating that a very low or very high number of this parameter leads to better results. Furthermore, R^2 equal to 1 is obtained for the highest values of the tree depth, indicating that the deeper the DT, the more complex relationships in the data can be captured, hence improved algorithm performance. These results are consistent with the conclusions reported by Qi et al. [68] for carbon fiber reinforced plastics. In such study, a tree depth of 7 was selected as optimal for the prediction of E. When the tree was deeper, over-fitting was observed, while for smaller depths, the model was less precise in describing the details of the training set. Therefore, in order to obtain an appropriate algorithm, experiments should be carried out on model trees with multiple depths.

On the other hand, the comparison of both techniques indicates that RandomizedSearchCV leads to slightly better performance on this specific algorithm, with a R^2 value of 1 and 0.7555 for the training and test data, respectively. This can be attributed to its randomised approach and the ability to explore a wider range of hyperparameter combinations in a shorter time compared to GridSearchCV.

Table 2
Hyperparameter ranges for regression DT searching.

Hyperparametres	Values
Maximum depth	0–100
Minimum samples split	1–5
Minimum samples leaf	1–60
Maximum leaf nodes	1–80
Maximum features	Auto, Sqrt, log2
Splitter	Best, Random

Table 3
Hyperparameter ranges for regression RF searching.

Hyperparametres	Values
Number of estimators	1–300
Maximum depth	1–300
Minimum samples split	2
Bootstrap	False
Warm start	True
Maximum features	sqrt

Table 4
R^2 values for different hyperparameters and search techniques for the prediction of the Young's Modulus with regression DT.

Search techniques	Maximum depth	Maximum leaf nodes	R^2 train	R^2 test
GridSearchCV	25	23	1.0000	0.7308
	20	19	0.9916	0.7405
	12	19	0.9916	0.7405
RandomizedSearchCV	26	79	1.0000	0.7555
	19	29	1.0000	0.7555
	5	45	0.9940	0.7307

4.3.1.4. RF. The results of this model are collected in Table 5. It can be observed that increasing the number of estimators does not systematically improve the model performance on the test set. For example, the comparison of the combinations with 100 and 200 estimators in GridSearchCV shows that the former combination yields a slightly higher R^2

Table 5
R^2 values for different hyperparameters and search techniques for the prediction of the Young's Modulus with regression RF.

search techniques	Number of estimators	Maximum depth	R^2 train	R^2 test
GridSearchCV	100	200	1.0000	0.6277
	100	20	1.0000	0.6135
	200	20	1.0000	0.5923
RandomizedSearchCV	42	71	1.0000	0.6153
	10	94	1.0000	0.5216
	102	94	1.0000	0.5897

Table 6
R^2 for training set and test set in different predictive models for the studied mechanical properties.

	Determination coefficient	RNN	RNN-LV	DT	RF
Young's modulus	R^2 training	1.0000	0.9974	1.0000	1.0000
	R^2 testing	0.9549	0.9710	0.7555	0.6277
Tensile Strength	R^2 training	0.9868	1.0000	1.0000	1.0000
	R^2 testing	0.6440	0.6784	0.6033	0.7400
Elongation at break	R^2 training	1.0000	1.0000	1.0000	1.0000
	R^2 testing	0.9203	0.9119	0.7737	0.7046
Impact Strength	R^2 training	1.0000	1.0000	1.0000	1.0000
	R^2 testing	0.9427	0.8820	0.7778	0.7877

(0.6277 vs. 0.5923). In addition, the combination with only 42 estimators in RandomizedSearchCV yield slight better results than that with 102 estimators (0.6153 vs. 0.5897). On the other hand, as the maximum depth of the trees decreases, the performance on the test set tends to decrease. For example, in GridSearchCV, the combination with a maximum depth of 200 yields a R^2 value in the test set of 0.6277, while that with a maximum depth of 20 led to comparable R^2 (0.6135) for the same number of estimators. The same occurs for RandomizedSearchCV, where the combination with a maximum depth of 71 yielded higher R^2 than that with 94 (0.6153 vs. 0.5897), suggesting that a higher maximum depth may allow the model to take more complex relationships in the data and improve prediction performance. Overall, the highest R^2 value in the train and test set (1 and 0.6277, respectively) are obtained for the combination with 100 estimators and a maximum depth of 200 in the GridSearchCV technique (see Table 6).

It is worthy to note that, despite the models used in this study work well to fit the experimental Young's modulus data, they present some limitations that could reduce the accuracy of the fitting. Indeed, it has been reported that the nanocomposite modulus is dependent on the particle size and shape, particle agglomeration and the particle/matrix interfacial adhesion, to mention the most important factors [69]. For the same nanofiller loading, and nanoparticle size below 100 nm, the modulus decreases with increasing size. This could be explained considering that smaller nanoparticles are usually better dispersed within the matrix, thus can more effectively restrict the deformation of the polymer chains. They also lead to a larger interphase, hence improved stress transfer. Further, the deformation mechanisms of nanoparticles, namely nanoparticle dislodging, shear-band formation, and cracking are strongly dependent on the size. For instance, with decreasing the particle size of SiO_2 from 300 nm to 10 nm, their stiffening effect was increased by 15-fold [69]. The modulus is also conditioned by the filler-matrix interfacial bond strength. Enhanced interfacial adhesion leads to improved stress transfer across the nanofiller–matrix interface, which is reflected in higher modulus. Further, nanoparticle agglomeration significantly reduces the effective elastic modulus of composites. The high surface area of nanoparticles results in the interaction between particles and aggregation/agglomeration. As a result, the nanofiller-matrix interfacial area is reduced and this weakens the performance. Further, the orientation of the nanofillers conditions the modulus. Nanofillers aligned parallel to the applied load result in higher composite stiffness. It should also be mentioned that the presence of defects in the nanofillers can significantly reduce their modulus, resulting in nanocomposites with decreased stiffness.

4.3.2. Tensile strength

4.3.2.1. RNN. Regarding the predictions of the tensile strength with RNN, R^2 remains almost constant as the number of hidden layers increases in the training data (Fig. S1 A). However, in the test data, R^2 peaks at two hidden layers and then decreases as additional layers are added, suggesting that this number is enough to capture the complexity of the data and avoid overfitting. Thus, the number of hidden layers was set to two, and the change in R^2 with the number of neurons in those two layers was analyzed (Fig. S1 B). In the training data, R^2 remains almost constant, while in the test data, maxima are observed at 50 and 80 neurons, the optimal being 50 neurons, that yields R^2 values of 0.9868 and 0.6440 for the train and test data, respectively. Upon addition of more neurons, the performance decreases, likely due to over-fitting in the test data. Similar trend was reported for low density polyethylene (LLDPE)/graphene nanocomposites [64], in which the highest R^2 value for the training set was 0.98 when 12 neurons were used in the hidden layer. Further addition of more neurons resulted in a noticeable reduction of R^2, by more than 10%. Analogous behavior was also found for carbon-black filled rubber composites [70], in which the best performance (0.999) was attained with 6 neurons, while more neurons led to data overfitting. It is worthy to note that the system studied herein is much complex than those reporter earlier, which accounts for the slighter lower R^2 values obtained.

4.3.2.2. RNN-LV. Regarding RNN-LV model (Fig. S2 A), R^2 remains constant over almost the whole range studied in the training data, while in the test data, it shows a maximum at two hidden layers and then decreases as more layers are added. Performance was then analyzed as a function of the number of neurons in these two hidden layers (Fig. S2 B). An increasing trend is observed for both training and test data as the number of neurons is increased. However, the best performance is reached at around 60 neurons, yielding R^2 for the training and test data of 1 and 0.6784, respectively. Beyond this point, the performance decreases, showing similar trend to that described above for the RNN model.

4.3.2.3. DT. Table S1 shows R^2 analysis as a function of the hyperparameters using the different search techniques for the DT model. In the case of GridSearchCV, this parameter remained almost constant for the training set, regardless of the maximum depth and maximum leaf nodes values considered. However, R^2 of the test set dropped as the value of maximum leaf nodes decreased, ascribed to the fact that fewer leaf nodes limit the model's ability to fit and generalize the data correctly. Regarding the test set, the decline was found to be less pronounced at maximum depth values between 117 and 113, suggesting that the model reaches an optimal point of complexity in that range, where it can capture the structure of the data effectively. However, when the maximum depth was reduced to 108, an increase in test set performance was observed, likely because a lower maximum depth allows for better regularisation, thus avoiding overfitting and improving the generalizability of the model.

In the case of RandomizedSearchCV, three hyperparameters were explored: Splitter, maximum depth, and maximum leaf nodes. By using the value "Random" for Splitter, the performance of the training set remained constant. However, when "Best" option was chosen, a decrease in R^2 was observed, indicating that a random selection provided a better fit to the training data. A model that is highly tuned to the training data may have difficulty generalising and fitting to new data. Therefore, using "Random" option for Splitter resulted in more balanced performance on both the training set and the test set.

Regarding the test set, the performance increased as the maximum depth value was reduced. This can be explained considering that a lower maximum depth limits the complexity of the model, thus avoiding overfitting and improving its ability to generalize the test data. The best performance of the test set was attained for a maximum leaf nodes value of 40 in RandomizedSearchCV. However, as this parameter increased to 85, the performance was worsened. This indicates that a moderate number of leaf nodes allows for a better capture of variability in the test data, while a higher number leads to a more complex model that is prone to overfitting.

Overall, the results indicate that RandomizedSearchCV performed

better compared to GridSearchCV, yielding a higher R^2 on both the training and test sets (1 and 0.6033, respectively, for the optimal combination of a random splitter, a maximum depth of 35 and a maximum leaf node of 40).

4.3.2.4. RF. The results obtained with the RF model are gathered in Table S2. Regarding GridSearchCV, increasing the number of estimators from 100 to 200 does not seem to have a significant impact on the model performance, as R^2 in the test set decreases slightly from 0.7240 to 0.7147 and 0.7118 for maximum depths of 20 and 200, respectively. For RandomizedSearchCV, the highest R^2 (0.7400) in the test set is attained for the combination of 100 estimators and a maximum depth of 100. Fewer estimators [10,16] lead to lower R^2 values (0.6473 and 0.5755, respectively), suggesting that a larger number of estimators has a positive impact on model performance. On the other hand, combinations with larger maximum depths (176 and 254) underperform. It is worthy to note that R^2 in the train set was equal to 1 for all the combinations investigated. Overall, the combination that leads to the best performance is found for RandomizedSearchCV, which leads to R^2 values of the train and test of 1 and 0.74, respectively. The values obtained herein are significantly higher than those obtained by Gupta et al. [71], who used a RF model to predict the tensile strength of geopolymer composites reinforced with different amounts of fly ash and calcined clay. They achieved R^2 of 0.88 and 0.57 for the training and test sets, respectively. They are also better than those reported for polypropylene filled with $CaCO_3$ [72], in which the best performance (R^2 of 0.9147 for the train set) was attained for the hyperparameter combination [1,15,55], corresponding to the number of DTs, maximum depth and minimum number of samples in a node, respectively. The authors also found that the optimal hyperparameter combination changed for the different physical properties (ie. modulus, strength, specific gravity, melting index), in agreement with the behavior found in this work.

As mentioned earlier for the Young's modulus, the strength of nanocomposites is influenced by several factors, including the nanofiller size, level of dispersion within the matrix and interfacial adhesion. These parameters are not considered by the ML models, hence can cause inaccurate fittings and deviations between experimental and theoretical values. Smaller nanoparticles and thicker interphases result in improved strength. Accordingly, the aggregated/agglomerated nanoparticles have lower tensile strength than the individual particles, and negatively affect the interfacial/interphase properties and tensile strength of polymer nanocomposites [69].

4.3.3. Elongation at break

4.3.3.1. RNN. Similarly to the tensile strength, R^2 of the training data for the prediction of the elongation at break with an RNN model increases with increasing number of hidden layers Fig. S3A. However, for the test data, R^2 shows a maximum at the third hidden layer and then drops when a fourth layer is added. Thus, addition of more layers does not improve the generalizability of the network and may lead to overfitting. Once the number of hidden layers was set to three, the performance was evaluated as a function of the number of neurons in the hidden layers, Fig. S3B. For the training data, a constant trend is observed with increasing number of neurons. However, for the test data, R^2 increases until 50 neurons are reached, then decreases for 60 neurons, and gradually increases again from 60 to 90 neurons, peaking at 90 neurons with R^2 of the training and test set of 1 and 0.9203, respectively. However, for 100 neurons, performance starts to drop on a regular basis. The best R^2 obtained for this model are higher than those found in the study by Palomba et al. to predict the elongation at break of a group of linear polymers, in which a maximum R^2 of 0.88 was achieved [73]. Further, they are slightly better than those reported for organoclay-reinforced rubber [74] nanocomposites, in which the lowest error and higher accuracy (R^2 of 0.9873 in the training set) was attained for 5 neurons in the hidden layer.

4.3.3.2. RNN-LV. Regarding RNN-LV model, although the performance of the training data remains constant, a decrease in R^2 is observed for the test data when only 2 hidden layers are selected (Fig. S4A). The best performance is attained with 3 hidden layers, hence it was selected as optimal and the change in R^2 as a function of the number of neurons in these 3 hidden layers was investigated (Fig. S4B). Regarding the training values, the performance remained constant with increasing the number of neurons, indicating that the network was able to fit the training data well, regardless of the number of neurons used. However, for the test set, a gradual rise in R^2 was observed, showing a maximum at 50 neurons (R^2 of training and test of 1 and 0.9119 respectively), and then dropped slightly until 100 neurons were reached in the hidden layers.

4.3.3.3. DT. The results obtained from DT model are collected in Table S3. The parameters minimum samples split, minimum samples leaf and maximum features were set to 2, 1 and "auto" respectively. In the case of GridSearchCV, R^2 of the training data remains high and almost constant regardless of the maximum depth values, while for test data it increases as this parameter decreases, indicating that a smaller value of maximum depth allows to obtain a better performance in the generalization of the model. In addition, the best performance on the test data is obtained with a maximum leaf nodes value of 43, the highest among the values tested for GridSearchCV. These results suggest that properly tuning the values of maximum depth and maximum leaf nodes can have a significant impact on the model performance.

Regarding RandomizedSearchCV, R^2 of the training data increases upon decreasing the value of maximum depth, suggesting that reduced values of this parameter can avoid overfitting, hence the generalization of the model is improved. It is interesting to note that the minimum maximum depth value, 106, provides the best performance on the training data, suggesting that a model with less depth may fit the training data more effectively. This value of maximum depth also leads to the best R^2 in the test set, 0.7670, for the highest value of maximum leaf nodes [37]. This implies that a larger number of nodes allowed for more complexity in the model, since it can capture more pinpoints patterns present in the test data, thus leading to a better fit. On the other hand, R^2 of 1 on the training set and 0.7737 on the test set was attained using the GridSearchSv algorithm with a maximum depth of 54 and a maximum number of leaf nodes of 43 (Table S3).

4.3.3.4. RF. The results obtained with the RF model for the prediction of the elongation at break are collected in Table S4. Regarding GridSearchCV, the model performance drops both in the training test set as the number of estimators decreases. This can be clearly observed by comparing the three combinations with different values of number of estimators (200, 150, and 100), where R^2 for the test set gradually decreased from 0.7005 to 0.7018 and then to 0.6687. On the other hand, no clear trend in the performance is found upon modifying the maximum depth. Combinations with the same maximum depth value (100) have similar results in both the training set and the test set, with R^2 hovering around 0.7. For RandomizedSearhCV, combinations with different values of number of estimators [14,21,28] do not show a clear trend in model performance. R^2 of the test set changes slightly between combinations, being in the range of 0.6969–0.7085 for all of them. Also, hardly change in model performance is observed with increasing maximum depth. The combination with a maximum depth of 259 and the smallest number of estimators [14] shows the highest R^2 for the test set (0.7046), while the combinations with lower maximum depths display just slightly lower values (R^2 for the test set of 0.6969 and 0.6959). Likely, a smaller number of estimators can avoid overfitting and improve the generalization of the model. It is significant that R^2 of the training set was systematically 1 for all the combinations tested. These results are better than those reported by Khayyami [75] for

LLDPE/carbon black using RF, in which the maximum R^2 value in the train set was 0.951. The authors concluded that this model did not generalize well, likely due to overfitting and to an artefact of random forest working with categorizations of data. Thus, the model requires a large number of data to provide reliable predictions.

4.3.4. Impact strength

4.3.4.1. RNN. Regarding the impact strength, results from the RNN model in Fig. S5 A show that R^2 of the training data rise with increasing number of layers, indicating that the model is learning better and getting better results in the training phase. For the test data, the performance increases up to layer 3 and then starts to decrease, showing similar pattern to that described previously for the other properties. To further analyze the performance, the change in R^2 with the number of neurons in the 3-layer model was investigated (Fig. S5B). In the training data, the performance is hardly influenced by the number of neurons, while in the test data, it rises as the number of neurons increases, showing a maximum for 50 neurons (R^2 of 0.9427), and then gradually decreased.

4.3.4.2. RNN-LV. Regarding the RNN-LV structure for the prediction of the impact strength (Fig. S6A), R^2 of the training data remains constant as the number of layers increases, for the test data the performance improves as the number of layers increases. This means that increasing the depth of the neural network can help to improve the overall performance. Based on this analysis, a four-layer network was selected, and the effect of the number of neurons in the 4 layers was investigated (Fig. S6B). In this case, the performance of the training data remains constant, regardless of the number of neurons in each layer, while in the test data, an increase in R^2 is observed up to 50 neurons. At this point, the maximum performance of the neural network is reached (R^2 of 1 and 0.882 for the training and test set, respectively), similarly to the RNN-LV model for the prediction of the elongation at break and the two RNN models for the Young's Modulus.

4.3.4.3. DT. With regard to the DT model (Table S5), the performance in the training data remains constant with a value of 1 regardless of the hyperparameters set and the search technique used. However, in the test data, it increases as the maximum depth decreases. Regarding the maximum leaf nodes, the behavior is slightly different for each search technique. In the case of GridSearchCV, no clear trend is observed upon changing this parameter, while slightly better performance is found in the data test data for the lowest value of maximum leaf nodes. On the other hand, in the case of RandomizedSearchCV, as the maximum leaf nodes decrease, the performance of the test data increases. It is important to note that the maximum throughput on the test data (1 and 0.7778, for the training and test sets, respectively) is achieved when using a maximum depth value of 6 and a maximum leaf nodes value of 38 in RandomizedSearchCV.

4.3.4.4. RF. Results regarding RF for the prediction of the impact strength are collected in Table S6. In the case of GridSearchCV, combinations with different values of number of estimators (100 and 150) provide similar results in terms of performance on the test set, indicating that they do not cause overfitting. The combination with 100 estimators and a maximum depth of 20 shows slightly lower performance on the test set, likely, because a maximum depth of 20 limits the model's ability to capture more complex relationships in the data. On the other hand, the combination with a maximum depth of 200 shows the best performance in the test set. In the case of RandomizedSearchCV, no clear trend is found with the values of number of estimators [22],]80]. It is important to note that performance may depend on other hyperparameters and dataset specifics. Furthermore, the combination with a maximum depth of 41 shows the best performance in the test set (R^2 of 0.7877). Combinations with lower maximum depths (22 and 83) may limit the model's ability to capture more complex patterns or may be causing overfitting. Noticeably, R^2 of the training set was systematically 1 for all the combinations tested.

It should be noted that the differences between theoretical and experimental data could be attributed to many factors that influence the nanocomposite performance and are not considered by the models. Indeed, the level of nanoparticle dispersion strongly influences the nanocomposite impact strength. Thus, agglomerated particles can act as stress concentration sites, consequently the impact strength is reduced [69].

4.4. Error assessment of predictive models. Measured vs. predicted values

To choose the most effective prediction model for each of the analyzed mechanical properties, a comparison based on the coefficient of determination was carried out (table 6). This coefficient is used as a measure to assess the ability of a model to explain the data variability. It is important to note that a higher value of R^2 indicates a better fitting for the model, since it can explain a greater proportion of the variability present in the data.

Regarding the Young's modulus, the most effective predictive model is the RNN-LV, with 3 hidden layers and 50 neurons in each layer, leading a R^2 of 0.9710. For the tensile strength, the RF model stands out, yielding a R^2 of 1 and 0.7400, respectively. With regard to the elongation at break, the best model is the RNN, with 3 hidden layers and 90 neurons in each layer, giving a R^2 of 0.9203. Finally, for the impact strength, the best model is the RNN with 3 layers and 50 neurons per layer, with a R^2 of 0.9427.

These findings indicate that the choice of the optimal ML model depends on the specific mechanical property to be predicted, rather than solely on the nature of the system itself. In our study, the nanocomposites present a complexity that implies the presence of four different components, and each mechanical property is influenced by several factors, as mentioned earlier, such as the adhesion between the matrix and the filler, the interactions between filler particles, the matrix modulus, strength, and ductility, as well as the nanofiller modulus, amongst others. This additional complexity makes it difficult to generalize to a single model that works for all the investigated mechanical properties.

Fig. 8 shows the correlation between the predicted values and the measured data for the four mechanical properties investigated. The regression model with the highest R^2 for each particular property was plotted. The Young's modulus was evaluated using RNN-LV, the tensile strength with the RF model, while the RNN model was used for the elongation at break and impact resistance. To assess the reproducibility of the models, ten simulations were performed under the same selected optimal conditions.

The plots reveal a good correlation between the predicted values and the measured data, since the points are grouped close to the diagonal line, which indicates an optimal fit of the model. This is evidenced by the slope of the graphs, which is very close to 1.0 in all cases. The best fitting is obtained for the elongation at break ($R^2 = 0.964$, Fig. 8C), followed by the impact strength (Fig. 8D), while the least accurate is found for the tensile strength. Similar results were reported for the prediction of the mechanical properties of polyamide 6-based composites using DT [76], in which the best correlation for the tensile strength (0.839) was significantly lower than that for the elongation at break (0.985). This authors attributed this different behavior to the anisotropy of the films. Indeed, the preferential orientation of the nanofillers in a certain direction might induce an anisotropic behavior which would be more pronounced for the tensile strength, and this factor is not considered by the ML models. Despite in our study the three nanofillers seem to be randomly dispersed, the presence of small aggregates or higher degree of entanglement in certain regions could induce a certain level of anisotropy, that could influence more the tensile strength than the elongation at break. Overall, results obtained herein demonstrate the good

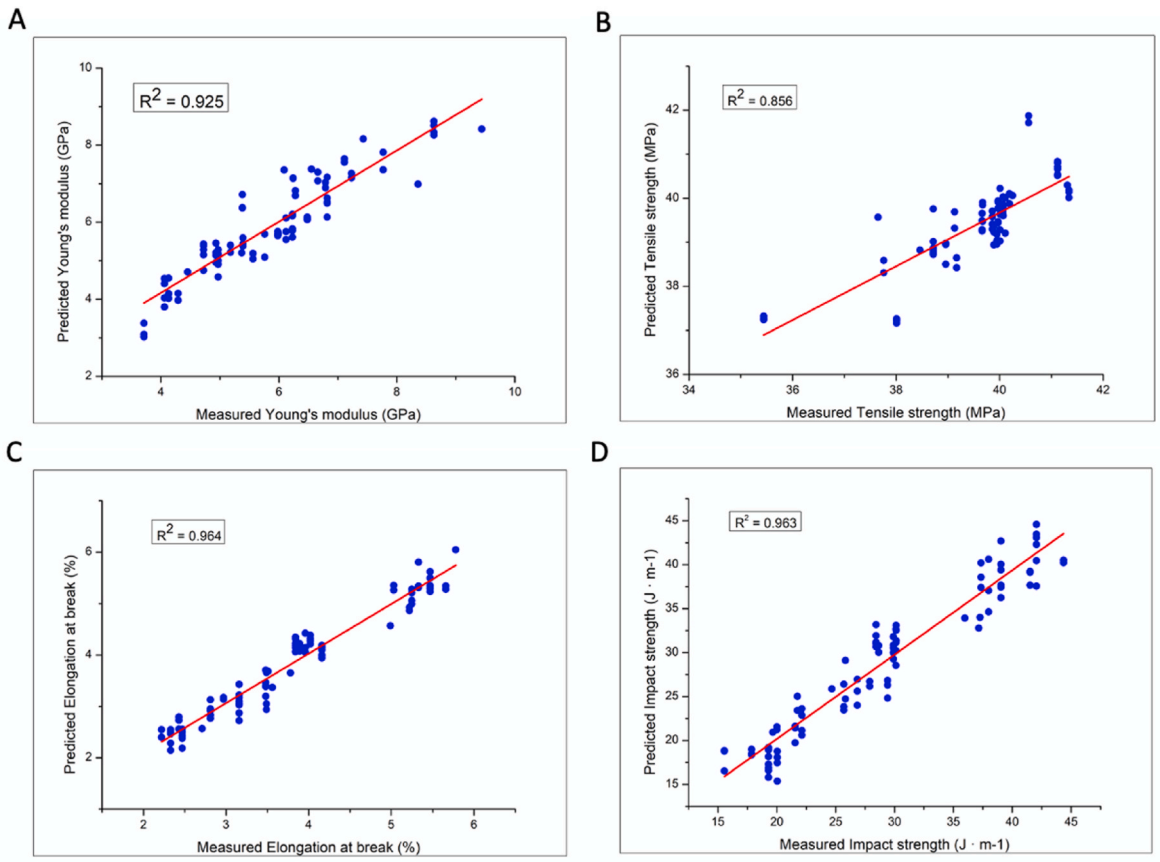


Fig. 8. Parity plots showing predicted values using the best regression models for each property vs. the measured values for: (A) Young's modulus; (B) Tensile strength; (C) Elongation at break; (D) Impact Strength.

agreement between the theoretical predictions and the experimental measurements.

To further assess the accuracy of the ML models implemented herein, the minimum squared errors (MSE) and the mean absolute errors (MAE) were also compared, and the results are collected in Table 7. The MSE is widely considered as an important validation criterion for ML models. The lower MSE, the most accurate the predictions of the desired property.

Regarding the Young's modulus, the RNN-LV model shows the lowest MSE (0.0806), which indicates a higher accuracy in the prediction of this property compared to the other models. However, in terms of MAE, the RNN model has a slightly lower value (0.2340), which implies a smaller average difference between the predictions and the experimental values. Regarding the elongation at break, the RNN model shows the lowest MSE (0.1295) and MAE (0.3095), indicating a better general approximation to the experimental values. This model also shows the lowest MSE (5.5034) and MAE (1.9010) for the prediction of the impact strength, indicating better accuracy compared to the other models.

Table 7
Comparison of MSE and MAE for the four prediction models for the Young's modulus, tensile strength, elongation at break and impact strength.

	Metrics	RNN	RNN-LV	DT	RF
Young's modulus	MSE	0.1212	0.0806	0.4170	0.6560
	MAE	0.2340	0.2517	0.5822	0.7652
Tensile Strength	MSE	0.435228	0.3932	0.9812	0.6429
	MAE	0.544431	0.5517	0.7800	0.6860
Elongation at break	MSE	0.1295	0.1433	0.3020	0.3942
	MAE	0.3095	0.3307	0.4667	0.5690
Impact Strength	MSE	5.5034	11.3285	17.0465	16.2880
	MAE	1.9010	2.8248	3.6911	3.4261

RNNs are ML models that can capture complex, non-linear relationships in data. The mechanical properties investigated herein are influenced by multiple intricate factors. Neural networks can learn and model these non-linear relationships, allowing them to provide more accurate predictions. Also, their hierarchical structure allows them to learn high-level features from input features, which contributes to better prediction. Neural networks are also adaptable to different data patterns and distributions and benefit from learning from large data sets. In summary, neural networks are powerful tools for tackling complex prediction problems in materials science and mechanics, as they can capture the complexity of mechanical properties and provide more accurate results. The choice of the RNN training algorithm can influence the performance and fit of the model to the data. For the Young's modulus, the RNN-LV seems to be more suitable to capture the non-linear relationships. This algorithm can fit the neuron weights and biases efficiently, which may allow a better fit of the model to the data and, consequently, a higher coefficient of determination. In contrast, for the elongation at break and impact resistance, it is possible that the relationships between the variables are less complex and more linear compared to Young's modulus. In these cases, RNNs may be sufficient to capture and model the linear relationships present in the data, resulting in higher R^2 compared to RNN-LVs.

Table 7 compares the experimental and predicted values by the four predictive models for the investigated mechanical properties. In general, a good agreement between the experimental and predicted values is observed, although in some cases there are discrepancies, in particular for the results of the DT model, since it shows lower R^2 and higher MSE and MAE values compared to RNN models.

5. Conclusions

The development of hybrid polymer nanocomposites at different scales is a process that requires a large economic investment and takes a long time of experimental work. The behavior of these materials is strongly influenced by the composition used, making it difficult to accurately predict their properties. In this study, hybrid nanocomposites based on PHB as a matrix and reinforced with multi-walled carbon nanotubes (MWCNTs), sepiolite nanoclay, and WS_2 nanosheets were developed. Experimental stress-strain and impact tests were performed to measure the Young's modulus, tensile strength, elongation at break, and impact strength. Furthermore, the effectiveness of different ML-based regression models, namely RNN, RNN-LV, DT and RF, to predict these mechanical properties was evaluated, and the effect of different factors on the accuracy of the models was investigated. Regarding the prediction of the Young's modulus, the best model is an artificial neural network with RNN-LV structure, comprising three hidden layers with 50 neurons in each layer. This model shows R^2 values of 0.997 and 0.971 for the training and test sets, respectively. In addition, it presents low MSE values of 0.0806 and MAE of 0.2517. On the other hand, the most favorable model to predict the tensile strength is the RF model. Although its performance is worst compared to the other properties, it can adequately predict this property using a combination of 100 estimators and a maximum depth of 100, with R^2 values of 1 and 0.74 for the training and test set, respectively, and values of MSE and MAE of 0.6429 and 0.686, respectively. Regarding the prediction of the elongation at break, an RNN model with 3 hidden layers and 90 neurons in each layer gave the best results. This model fits accurately the data, with R^2 of 1 and 0.9203 for the training and test set. In addition, MSE values of 0.1295 and MAE of 0.3095 were obtained. Concerning impact resistance, an RNN model with 3 hidden layers and 50 neurons is effective for predicting this property. The model perfectly fits to the data, with R^2 of 1 and 0.9247 for the training and test set, and MSE and MAE values of 5.5034 and 1.9010, respectively. The high correlation and small errors obtained in the models are strong indicators of their accuracy. This implies that the models can generalize and maintain a consistency in the results, reflected in the uniform values of the correlation coefficient and the mean square errors in the training and test data sets. It is important to note that despite the models used in this work fit very well the experimental data, they exhibit some limitations arising from the complexity that implies the presence of four different components, and each mechanical property is influenced by several factors, including the nanofiller shape, orientation and size, the nanofiller-matrix interfacial adhesion and the agglomeration/aggregation state of the nanoparticles, amongst others. These factors could account for the differences found between experimental and predicted data.

Overall, results obtained herein confirm the reliability of the ML models to accurately estimate and predict the specific mechanical properties of multiscale hybrid nanocomposites. This approach offers a series of significant benefits, among them, the substantial reduction of experimental work in the laboratory and the consequent optimization of the time and costs associated with procuring materials and performing repetitive physical tests. This allows greater efficiency in the design and development of new hybrid polymeric nanocomposites, since virtual tests can be carried out to evaluate different combinations and variations of materials, saving time and resources in the research and development phase.

Author statement

Elizabeth Champa-Bujaico: Data curation, investigation, writing original draft.

Ana Maria Diez Pascual: conceptualization, writing-review & editing, supervision, Funding acquisition.

Alba Lomas Redondo: Data curation, investigation.

Pilar Garcia Diaz: Writing - Review & Editing, validation, supervision.

Funding

Financial support from the Community of Madrid within the framework of the multi-year agreement with the University of Alcala in the line of action "Stimulus to Excellence for Permanent University Professors", Ref. EPU-INV/2020/012, is gratefully acknowledged.

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.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.compositesb.2023.111099.

References

[1] Jordan MI. Artificial intelligence—the revolution hasn't happened yet. Harvard Data Sci Rev 2019;1:1.
[2] Makridakis S. The forthcoming Artificial Intelligence (AI) revolution: its impact on society and firms. Futures: J Policy, Planning Fut Stud 2017;90:46–60.
[3] Nilsson NJ. In artificial intelligence: a new synthesis. Artificial intelligence: a new synthesis, vol. 1. Elsevier Science & Technology: United States; 1997. p. 1–17.
[4] Scerri M, Grech V. Artificial intelligence in medicine. Early Hum Dev 2020;145:105017.
[5] Ng GW, Leung WC. Strong artificial intelligence and consciousness. J Artif Intell Consciousness 2020;7:63–72.
[6] Bindushree V, Sameen R, Vasudevan V, Shrihari T, Devaraju D, Mathew N. Artificial intelligence: in modern dentistry. J Dent Res Rev 2020;7:27–31.
[7] Mittal U, Sharma DM. Artificial intelligence and its application in different areas of Indian economy. Int J Adv Res Sci, Commun Technol 2021;9001:160–3.
[8] Wang Y, Zhang M, Lin A, Iyer A, Prasad AS, Li X, Zhang Y, Schadler LS, Chen W, Brinson LC. Mining structure-property relationships in polymer nanocomposites using data-driven finite element analysis and multi-task convolutional neural networks. Mol Syst Des Eng 2020;5:962–75.
[9] Shen Z, Liu H, Shen Y, Hu J, Chen L, Nan C. Machine learning in energy storage materials. Interdisc Mater 2022;1:175–95.
[10] Guo K, Yang Z, Yu C, Buehler MJ. Artificial intelligence and machine learning in the design of mechanical materials. Mater Horiz 2021;8:1153–72.
[11] Sharma A, Mukhopadhyay T, Rangappa SM, Siengchin S, Kushvaha V. Advances in computational intelligence of polymer composite materials: machine learning assisted modeling, analysis, and design. Arch Comput Methods Eng 2022;29:3341–85.
[12] Champa-Bujaico E, García-Díaz P, Díez-Pascual AM. Machine learning for property prediction and optimization of polymeric nanocomposites: a state-of-the-art. Int J Mol Sci 2022;23:10712.
[13] Özkan M, Karakoç A, Borghei M, Wiklund J, Rojas OJ, Paltakari J. Machine Learning-assisted design of tailor-made nanocellulose films: a combination of experimental and computational studies. Polym Compos 2019;40:4013–22.
[14] Sahu SK, Sreekanth PSR. Artificial neural network for prediction of mechanical properties of HDPE-based nanodiamond nanocomposite. Porrimer 2022;46:614–20.
[15] Natrayan L, Bhaskar A, Patil PP, Kaliappan S, Dineshkumar M, Esakkiraj ES. Optimization of filler content and size on mechanical performance of graphene/hemp/epoxy-based hybrid composites using Taguchi with ANN technique. J Nanomater 2023;5:1–15.
[16] Khan A, Shamsi MH, Choi T. Correlating dynamical mechanical properties with temperature and clay composition of polymer-clay nanocomposites. Comput Mater Sci 2009;45:257–65.
[17] Adel H, Palizban SMM, Sharifi SS, Ilchi Ghazaan M, Habibnejad Korayem A. Predicting mechanical properties of carbon nanotube-reinforced cementitious nanocomposites using interpretable ensemble learning models. Construct Build Mater 2022;354:129209.
[18] Díez-Pascual AM, Díez-Vicente AL. ZnO-reinforced poly(3-hydroxybutyrate-co-3-hydroxyvalerate) bionanocomposites with antimicrobial function for food packaging. ACS Appl Mater Interfaces 2014;6:9822–34.
[19] Raza ZA, Abid S, Banat IM. Polyhydroxyalkanoates: characteristics, production, recent developments, and applications. Int Biodeterior Biodegrad 2018;126:45–56.

[20] Díez-Pascual AM, Díez-Vicente AL. Poly(3-hydroxybutyrate)/ZnO bionanocomposites with improved mechanical, barrier and antibacterial properties. Int J Mol Sci 2014;15:10950–73.
[21] García-Quiles L, Cuello ÁF, Castell P. Sustainable materials with enhanced mechanical properties based on industrial polyhydroxyalkanoates reinforced with organomodified sepiolite and montmorillonite. Polymers 2019;11:696.
[22] Cataldi P, Steiner P, Raine T, Lin K, Kocabas C, Young RJ, Bissett M, Kinloch IA, Papageorgiou DG. Multifunctional biocomposites based on polyhydroxyalkanoate and graphene/carbon nanofiber hybrids for electrical and thermal applications. ACS Appl Polym Mater 2020;2:3525–34.
[23] Díez-Pascual AM, Ashrafi B, Naffakh M, González-Domínguez JM, Johnston A, Simard B, Martínez MT, Gómez-Fatou MA. Influence of carbon nanotubes on the thermal, electrical, and mechanical properties of poly(ether ether ketone)/glass fiber laminates. Carbon 2011;49:2817–33.
[24] Díez-Pascual AM, Gascón D. Carbon nanotube buckypaper reinforced acrylonitrile–butadiene–styrene composites for electronic applications. ACS Appl Mater Interfaces 2013;5:12107–19.
[25] Díez-Pascual AM. Chemical functionalization of carbon nanotubes with polymers: a brief overview. Macromolecules (Washington, DC, U S) 2021;1:64–83.
[26] Díez-Pascual AM, Rahdar A. Composites of vegetable oil-based polymers and carbon nanomaterials. Macromolecules (Washington, DC, U S) 2021;1:276–92.
[27] Díez-Pascual AM, Naffakh M. Enhancing the thermomechanical behaviour of poly (phenylene sulphide) based composites via incorporation of covalently grafted carbon nanotubes. Compos Appl Sci Manuf 2013;54:10–9.
[28] Biddeci G, Spinelli G, Colomba P, Di Blasi F. Halloysite nanotubes and sepiolite for health applications. Int J Mol Sci 2023;24:4801.
[29] Ruiz-Hitzky E, Aranda P, Darder M, Fernandes FM. In chapter 13.3 - fibrous clay mineral–polymer nanocomposites; developments in clay science, vol. 5. Elsevier Ltd; 2013. p. 721–41.
[30] Neog A, Biswas R. WS2 nanosheets as a potential candidate towards sensing heavy metal ions: a new dimension of 2D materials. Mater Res Bull 2021;144:111471.
[31] Naffakh M, Díez-Pascual AM, Gómez-Fatou MA. New hybrid nanocomposites containing carbon nanotubes, inorganic fullerene-like WS2 nanoparticles and poly (ether ether ketone) (PEEK). J Mater Chem 2011;21:7425–33.
[32] Naffakh M, Díez-Pascual AM, Marco C, Ellis GJ, Gómez-Fatou MA. Opportunities and challenges in the use of inorganic fullerene-like nanoparticles to produce advanced polymer nanocomposites. Prog Polym Sci 2013;38:1163–231.
[33] Fatai A. 158. Anifowose application of artificial intelligence in network intrusion detection A succinct Review. World Applied Programming; 2012.
[34] Drew PJ, Monson JRT. Artificial neural networks. Surgery 2000;127:3–11.
[35] Zou J, Han Y, So S. Overview of Artificial Neural Networks Artificial Neural Networks. 458. 2008. p. 14–22.
[36] Flórez López R, Fernández Fernández JM. Las Redes Neuronales Artificiales. Fundamentos teóricos y aplicaciones prácticas. La Coruña, Spain: Netbiblo; 2008. p. 21–31.
[37] Abraham A. Neuro-fuzzy systems. John Wiley & Sons, Ltd.; 2005. p. 901–8.
[38] Rosenblatt F. The perceptron: a probabilistic model for information storage and organization in the brain. Spartan Books; 1962. p. 79–92.
[39] Jain AK, Mao Jianchang, Mohiuddin KM. Artificial neural networks: a tutorial. MC 1996;29:31–44.
[40] Lo SB, Lou SL, Lin JS, Freedman MT, Chien KV. Artificial neural network in mammography interpretation. Neural Network 1995;8:1201–14.
[41] Danilo P, Mandic ADC. Adaptive and learning systems for signal processing, communications, and control. Adaptive and learning systems for signal processing, communications, and control. Willy; 2001. p. 2–8.
[42] Gavin HP. The Levenberg-Marquardt algorithm for nonlinear least squares curve-fitting problems. 2023.
[43] Hagan MT, Menhaj MB. Training feedforward networks with the Marquardt algorithm. TNN 1994;5:989–93.
[44] Xinghuo Y, Shi P. Artificial neural networks for control systems. TNN 2002;13:251–4.
[45] Berner J, Elbrächter D, Grohs P. How degenerate is the parametrization of neural networks with the ReLU activation function?. 32. 2019. p. 7790–801.
[46] Ibrahim MM, Alnuwaiser MA, Elkaeed EB, Kotb H, Alshehri S, Abourehab MAS. Computational modeling of Hg/Ni ions separation via MOF/LDH nanocomposite: machine learning based modeling. Arab J Chem 2022;15:104261.
[47] de Ville B. Decision trees. Wiley interdisciplinary reviews. Comput Stat 2013;5:448–55.
[48] Loh W. Classification and regression trees. Wiley Interdisc Rev. Data Min Knowl Discovery 2011;1:14–23.
[49] Apté C, Weiss S. Data mining with decision trees and decision rules. Future Generat Comput Syst 1997;13:197–210.
[50] Mantovani RG, Horváth T, Cerri R, Barbon Junior S, Vanschoren J, Ponce de Leon Ferreira de Carvalho AC. An empirical study on hyperparameter tuning of decision trees. arXiv.org. 2019.
[51] Hristeva T. Application of graphic processing units in deep learning algorithms. AIP Conf Proc 2022:2449.
[52] Segal MR. Machine learning benchmarks and random forest regression. 2004.
[53] Polimis K, Rokem A, Hazelton B. Confidence intervals for random forests in Python. J Open Source Softw 2017;2:124.
[54] Rodriguez-Galiano V, Sanchez-Castillo M, Chica-Olmo M, Chica-Rivas M. Machine learning predictive models for mineral prospectivity: an evaluation of neural networks, random forest, regression trees and support vector machines. Ore Geol Rev 2015;71:804–18.
[55] Panda SK, Mohapatra RK, Panda S, Balamurugan S. In car buying criteria evaluation using machine learning approach; the new advanced society. United States: John Wiley & Sons, Incorporated; 2022. p. 223–45.
[56] De S, Moss H, Johnson J, Li J, Pereira H, Jabbari S. Engineering a machine learning pipeline for automating metadata extraction from longitudinal survey questionnaires. IASSIST Q 2022:46.
[57] Kurian Pullolickal P. Automatic classification of cardiovascular age of healthy people by dynamical patterns of the heart rhythm. Linköpings universitet, Statistik och maskininlärning; 2022. p. 43.
[58] Khanna A, Gupta D, Bhattacharyya S, Hassanien AE, Anand S, Jaiswal A. In sentiment analysis of multilingual mixed-code, twitter data using machine learning approach; international conference on innovative computing and communications, vol. 1388. Singapore: Springer Singapore Pte. Limited; 2021. p. 683–97.
[59] Carli LN, Crespo JS, Mauler RS. PHBV nanocomposites based on organomodified montmorillonite and halloysite: the effect of clay type on the morphology and thermal and mechanical properties. Compos Part A, Appl Sci Manuf 2011;42:1601–8.
[60] Ruoff RS, Qian D, Liu WK. Mechanical properties of carbon nanotubes: theoretical predictions and experimental measurements. C R Phys 2003;4:993–1008.
[61] Adigilli HK, Pandey AK, Joardar J. In 2D-nanolayered tungsten and molybdenum disulfides: structure, properties, synthesis, and processing for strategic applications; handbook of advanced ceramics and composites. Cham: Springer International Publishing; 2021. p. 75–120.
[62] Besli NSO, Orakdogen N. Sepiolite-embedded binary nanocomposites of (alkyl) methacrylate-based responsive polymers: role of silanol groups of fibrillar nanoclay on functional and thermomechanical properties. React Funct Polym 2021;161:104844.
[63] Al Mahmud H, Radue MS, Chinkanjanarot S, Odegard GM. Multiscale modeling of epoxy-based nanocomposites reinforced with functionalized and non-functionalized graphene nanoplatelets. Polymers 2021;13:1958.
[64] Ho NX, Le T, Le MV. Development of artificial intelligence-based model for the prediction of Young's modulus of polymer/carbon-nanotubes composites. Mech Adv Mater Struct 2022;29:5965–78.
[65] Khanam PN, AlMaadeed M, AlMaadeed S, Kunhoth S, Ouederni M, Sun D, Hamilton A, Jones EH, Mayoral B. Optimization and prediction of mechanical and thermal properties of graphene/LLDPE nanocomposites by using artificial neural networks. Int J Polym Sci 2016;2016:1–15.
[66] Zakaulla M, Pasha Y, Siddalingappa SK. Prediction of mechanical properties for polyetheretherketone composite reinforced with graphene and titanium powder using artificial neural network. Mater Today Proc 2022;49:1268–74.
[67] Liu J, Zhang Y, Zhang Y, Kitipornchai S, Yang J. Machine learning assisted prediction of mechanical properties of graphene/aluminium nanocomposite based on molecular dynamics simulation. Mater Des 2022;213:110334.
[68] Qi Z, Zhang N, Liu Y, Chen W. Prediction of mechanical properties of carbon fiber based on cross-scale FEM and machine learning. Compos Struct 2019;212:199–206.
[69] Fu S-Y, Feng XQ, Lauke B, Mai YW. Effects of particle size, particle/matrix interface adhesion and particle loading on mechanical properties of particulate–polymer composites. Compos Part B 2008;39:933–61.
[70] Kopal I, Labaj I, Harničárová M, Valíček J, Hrubý D. Prediction of the tensile response of carbon black filled rubber blends by artificial neural network. Polymers 2018;10:644.
[71] Gupta P, Gupta N, Saxena KK, Goyal S. Random forest modeling for fly ash-calcined clay geopolymer composite strength detection. J Compos Sci 2021;5:271.
[72] Joo C, Park H, Lim J, Cho H, Kim J. Development of physical property prediction models for polypropylene composites with optimizing random forest hyperparameters. Int J Intell Syst 2021;37:3625–53.
[73] Palomba D, Vazquez GE, Díaz MF. Prediction of elongation at break for linear polymers. Chemometr Intell Lab Syst 2014;139:121–31.
[74] Nematollahi M, Jalali-arani A, Golzar K. Organoclay maleated natural rubber nanocomposite. Prediction of abrasion and mechanical properties by artificial neural network and adaptive neuro-fuzzy inference. Appl Clay Sci 2014;97–98:187–99.
[75] Khayyami S. Predicting mechanical properties of polymer films after extrusion coating using supervised machine learning algorithms. Lund Institute of technology, Lund University; 2019. p. 51–9.
[76] Baturynska I. Application of machine learning techniques to predict the mechanical properties of polyamide 2200 (PA12) in additive manufacturing. Appl Sci 2019;9:1060.