Predicting mechanical properties of pristine and defective carbon nanotubes using random forest model: Electronic Supplementary Information

Ihtesham Ibn Malek[a], Koushik Sarkar[a], Ahmed Zubair[a,*]

[a]*Department of Electrical and Electronic Engineering, Bangladesh University of Engineering and Technology, Dhaka, 1205, Bangladesh*

S1. Reconstructing Defect Geometry

The nonreconstructed defect geometries consisting of three dangling bonds were generated by removing one atom from approximately the midpoint of pristine CNTs. Three reconstructed single-vacancy configurations can be generated by creating a bond between a pair of dangling bonds. It must be mentioned that two carbon atoms are considered to be bonded if the distance between them is less than 1.85 Å. The created defect geometries are separated from each other by an angle of 120°. To find the stable configuration, each CNT structure was energy minimized separately. Firstly, three initial structures were created by moving different sets of two adjacent dangling atoms surrounding the defect closer to each other, thus creating a pentagonal ring. Energy minimization in the steepest descent algorithm with AIREBO [1] potential was carried out, and the potential energy at the end of the simulation was noted for each configuration. The structure with the lowest energy is the most stable out of the three initial structures and was used in subsequent simulations. This process is shown in Fig. S1 for chiral CNT (19, 6). This process is repeated for all the CNTs, and the initial defective structures were generated.

*Corresponding author

Email addresses: shanto.bin.malek@gmail.com (Ihtesham Ibn Malek), koushik1564@gmail.com (Koushik Sarkar), ahmedzubair@eee.buet.ac.bd (Ahmed Zubair)

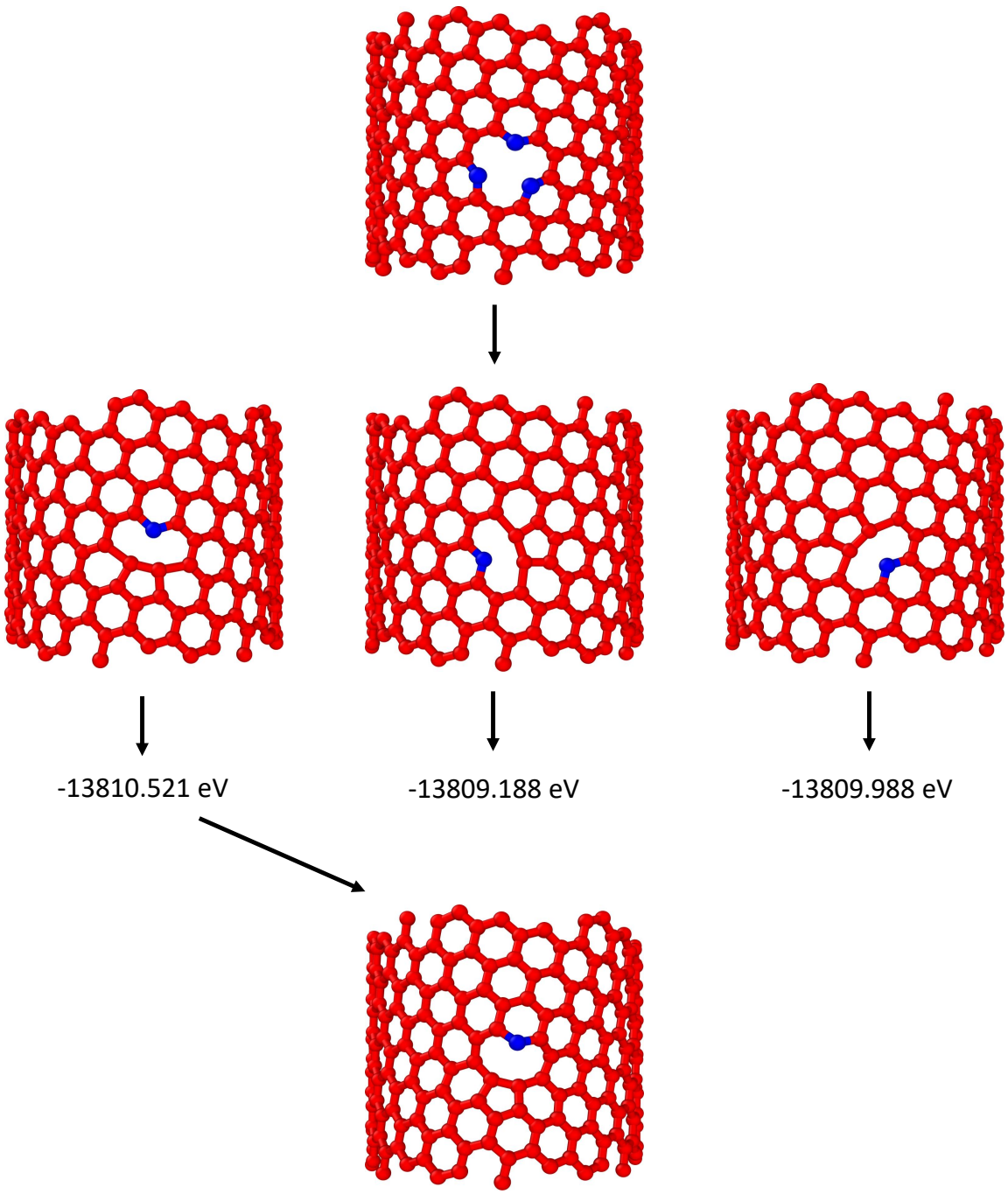

Figure S1: A concise representation of the process of generating stable CNT with single vacancy from nonreconstructed defect geometry. The top figure is the nonreconstructed defect geometry. The middle row shows the structures before energy minimization. The potential energy at the end is compared, and the corresponding reconstructed defect geometry is shown at the bottom.

S2. Smoothing Radius Variation with Cubic Spline

At each timestamp, the radius was calculated by averaging the distance of each atom belonging to the middle group from the CNT axis. At 300 K, the displacement of atoms in the horizontal plane due to thermal vibration was quite significant compared to tensile strain. As a result, the radius variation with time appeared to be noisy and fluctuating. To remove the thermal noise, radius variation was approximated by cubic smoothing spline interpolation. [2] The radius variation with strain extracted from MD simulation and corresponding cubic spline curve for CNT (14, 10) is shown in Fig. S2. The calculated radius differed significantly from the theoretical radius even before deformation due to energy minimization and thermal equilibration. Moreover, radial compression during tensile elongation introduced different degrees of variation in radius for different chirality. As a result, accurate calculation of the current radius was essential for determining the correct cross-sectional area, stress, and Poisson's ratio with varying strain.

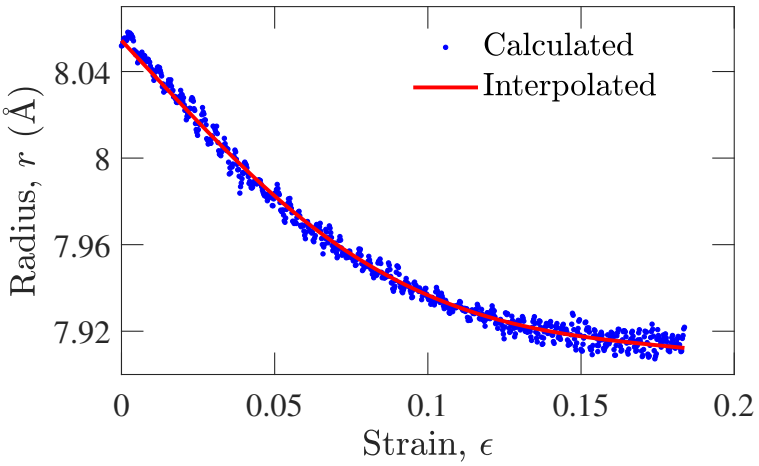

Figure S2: Variation of the radius with strain calculated from atom coordinates of CNT (14,10) and cubic smoothing spline fitted curve to the raw data.

S3. Fluctuation in fracture strain and tensile strength

Fig. S3 and S4 show the tensile strength and fracture strain for pristine and defective CNTs, respectively, as calculated from three molecular dynamics simulation runs with different seeds. To include the data from all CNTs in a single graph, the CNTs are sorted by their chiral indices (n, m), and the CNT order in the x-axis of these figures, is the position of a CNT in that list.

S4. Impact of strain rate and nanotube length

The dataset was generated by subjecting CNTs to a constant loading rate of 0.001 ps^{-1}. The length of these CNTs were set to ∼5 times their diameter. In this section, the change in Young's modulus (E), tensile strength (σ_{max} and fracture strain (ϵ_{max}) is observed by varying the initial length of CNTs and strain rate. Strain rates of 0.0005, 0.005, and 0.01 ps^{-1} were applied to the CNTs with lengths of ∼5 times the diameter. Then, the initial lengths were changed to ∼10, ∼25 times the diameter, and the nanotubes were elongated at a strain rate of 0.001 ps^{-1}.

A slightly increasing trend is observed in the parameters due to increasing strain rate. This observation conforms to the intuition that the fracture strain of a material decreases due to the fatigue from being held longer at a highly stretched position caused by a slow strain rate. A

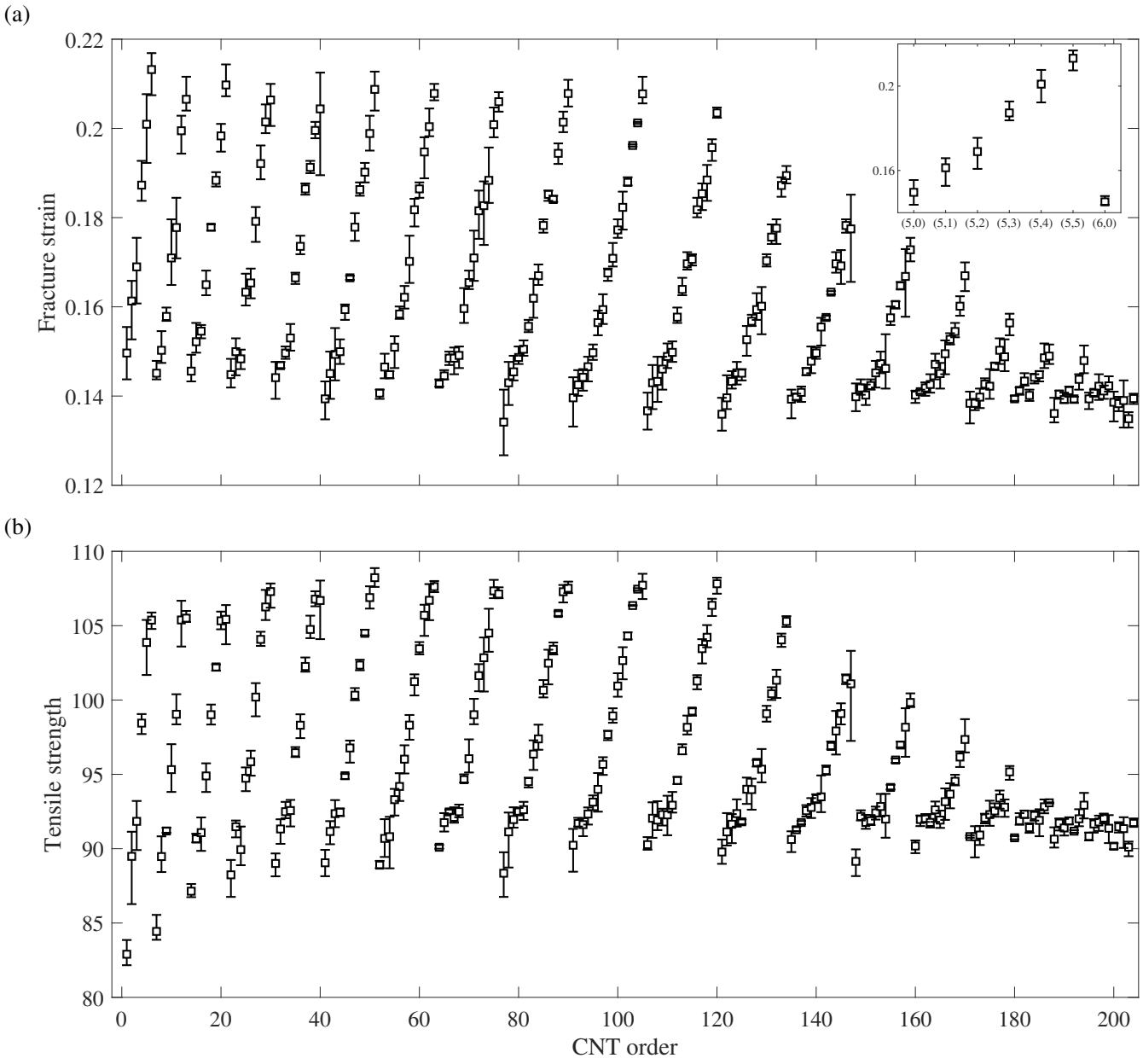

Figure S3: Variation of (a) fracture strain, (b) tensile strength of pristine carbon nanotubes with various chiral indices. The insert shows the first 7 data points depicting the ordering scheme of CNTs.

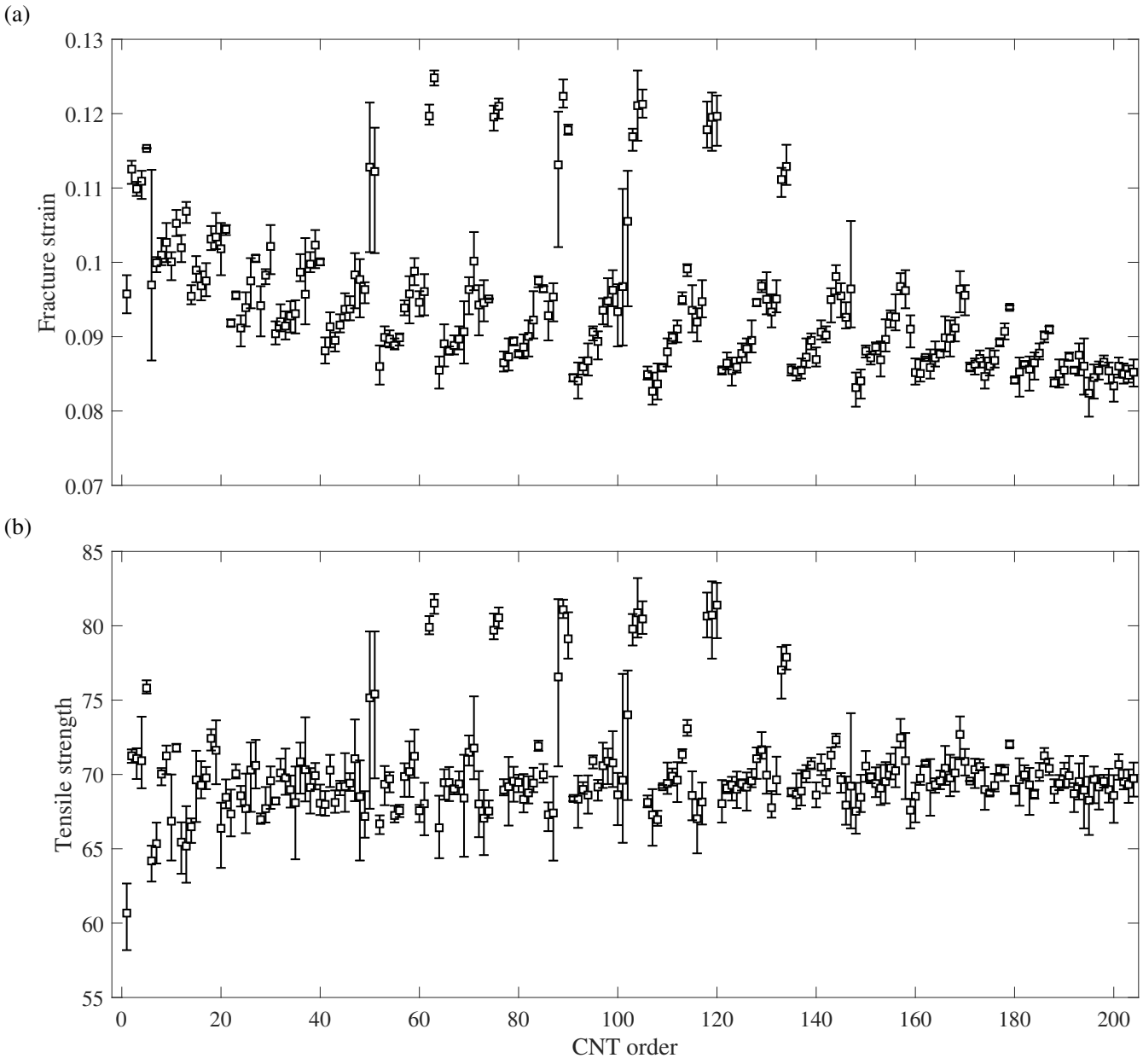

Figure S4: Variation of (a) fracture strain, (b) tensile strength of defective carbon nanotubes with various chiral indices.

similar decreasing trend is observed with increasing length. This can be attributed to the increase in the number of atoms i.e. points of failure at which fracture can start. More points of failure increase the probability of the breakdown of a bond when the strain is close to fracture strain. Despite these trends, the deviations are very small compared to the calculated properties of the original simulation, so our machine-learning model would still be able to predict the properties with reasonable accuracy, given that the strain rate and sizes are not too different.

Table S1: Variation of Young's modulus (E), tensile strength (σ_{max}) and fracture strain (ϵ_{max}) of several pristine carbon nanotubes (CNTs) with different strain rates and nanotube lengths.

Parameter	Chirality (n,m)	Data from dataset	Strain rate (ps^{-1})			Length ($\times\oslash$)	
			0.0005	0.005	0.01	10	25
ϵ_{max}	(6,6)	0.206	0.2065	0.212	0.22	0.204	0.195
	(7,5)	0.188	0.179	0.181	0.198	0.184	0.177
	(9,2)	0.149	0.145	0.153	0.154	0.142	0.138
	(10,0)	0.139	0.138	0.143	0.147	0.138	0.134
σ_{max}	(6,6)	105.14	105.35	106.38	106.88	105.07	103.33
	(7,5)	103.16	100.92	100.62	103.46	101.14	99.16
	(9,2)	92.47	90.41	91.66	92.54	89.39	88.8
	(10,0)	88.36	88.78	89.59	90.41	88.36	87.45
E	(6,6)	907.49	907.18	907.79	906.55	907.4	905.44
	(7,5)	926.53	927.57	925.91	924.54	925.93	924.38
	(9,2)	1045.25	1037.95	1043.61	1045.01	1037.66	1035.81
	(10,0)	1077.43	1075.94	1076.23	1080.03	1073.71	1062.57

S5. Principal Component Analysis

Principal Component Analysis (PCA) is a dimensionality reduction technique used in statistics and machine learning (ML). It transforms the original variables into a new set of variables called principal components (PC). ML regression helps assess how effectively the features contribute to predicting the data. To illustrate, let's consider a scenario with two independent variables, namely features 'n' and 'θ' (where 'n' is one of the chiral indices of a carbon nanotube denoted as (n, m), and 'θ' is its chiral angle). The dependent variable is the target, for instance, the diameter of a CNT measured in Å, as given in the following Table and shown in the accompanying figure. It must be centralized first if we want to perform PCA on the data. Each variable is presented in the subsequent table S2 and figure S5 by subtracting its corresponding mean; for example, 'n' is centered at 8, and 'θ' is centered at 9.5.

Table S2: Example Table with 15 Columns and 3 Rows

n	4	7	8	13
θ (°)	11	17	6	4
d (Å)	4	7	6	10

N=n-$\bar{n}$	-4	-1	0	5
$\Theta = \theta - \bar{\theta}$	1.5	7.5	-3.5	-5.5
D=d-$\bar{d}$	-2.75	0.25	0.75	3.25

PC1	-3.2	-7.1	3.1	7.2
PC2	2.9	-2.5	1.6	-1.9

Next, we have to find a line for which the sum of the squared distances between each point's projected point on the line and the origin is maximum, and the line should pass through the

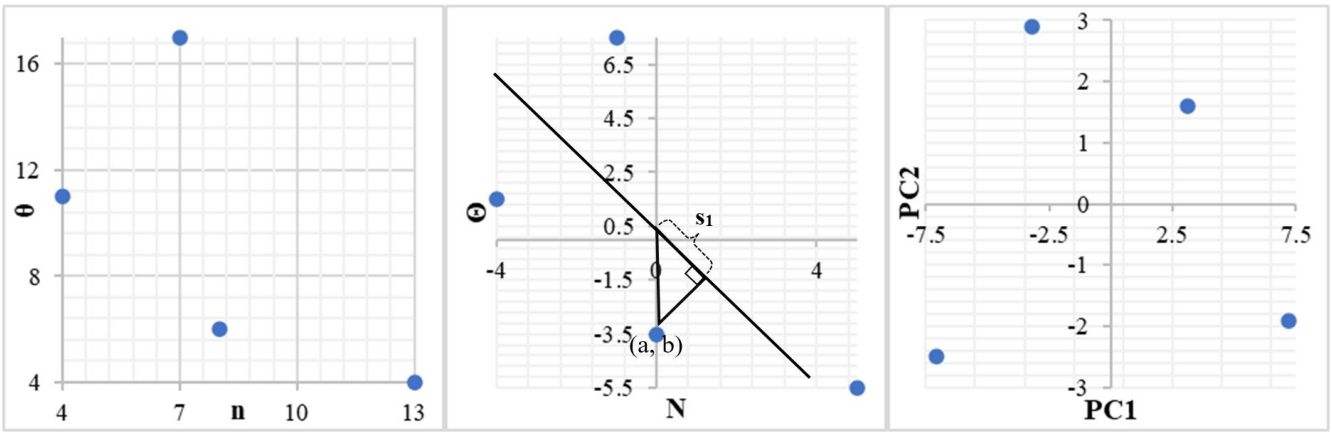

Figure S5: Example problem for n vs 'θ', N vs 'Θ' and PC1 vs PC2.

origin. Assuming the line has a slope M, the equation of the line is given by $y = Mx$. The distance between a point (a, b) and its projected point on this line to the origin is denoted by s_1, where, ${s_1}^2(a, b) = a^2 + b^2 - \frac{(Ma-b)^2}{M^2+1} = \frac{(Mb+a)^2}{M^2+1}$.So, $s_1^2(0, -3.5) = \frac{(-3.5M)^2}{M^2+1}$. Now, for maximization, we have to find M for which $\frac{d}{dM}(\sum_{i=1}^{4} s_i^2) = 0$ is maximum, and that gives $M = -1.95$. Unit vector through this line is $\phi_\mathbf{1} = \begin{bmatrix}\phi_{11}\\ \phi_{12}\end{bmatrix} = \begin{bmatrix}\frac{1}{\sqrt{1^2+(-1.95)^2}}\\ \frac{-1.95}{\sqrt{1^2+(-1.95)^2}}\end{bmatrix} = \begin{bmatrix}0.46\\ -0.89\end{bmatrix}$, where ϕ_{11} and ϕ_{12} are the components of ϕ_1 towards the direction of N and Θ, respectively. Perpendicular vector of ϕ_1 can be $\phi_\mathbf{2} = \begin{bmatrix}\phi_{12}\\ -\phi_{11}\end{bmatrix} = \begin{bmatrix}\phi_{21}\\ \phi_{22}\end{bmatrix} = \begin{bmatrix}-0.89\\ -0.46\end{bmatrix}$. Hence, the first PC (PC1) of first CNT ($n_1 = 4, \theta_1 = 11°$) is $\begin{bmatrix}\phi_{11} & \phi_{12}\end{bmatrix}\begin{bmatrix}N_1\\ \Theta_1\end{bmatrix} = \phi_{11}N_1 + \phi_{12}\Theta_1 = -4 \times 0.46 - 1.5 \times 0.89 = -3.2$ and second PC (PC2) of first CNT is $\begin{bmatrix}\phi_{21} & \phi_{22}\end{bmatrix}\begin{bmatrix}N_1\\ \Theta_1\end{bmatrix} = \begin{bmatrix}-0.86 & -0.46\end{bmatrix}\begin{bmatrix}-4\\ 1.5\end{bmatrix} = 0.89 \times 4 - 0.46 \times 1.5 = 2.9$. If $x_1, x_2, \ldots, x_q$ represent the original variables, and X is the data matrix with q variables (rows) and p observations (columns), the i-th principal component for m-th observation can be expressed as $PC_{i,m} = \sum_{k=1}^{q} \phi_{ik} \cdot x_{km}$, where coefficients ϕ_{ik} are chosen to maximize the variance of PC_i, subject to the constraint that $\sum_{k=1}^{q} \phi_{ik}^2 = 1$. Each PC should be orthogonal to each other. In general, $PC = \Phi^T X = \begin{bmatrix}0.46 & -0.89\\ -0.89 & -0.46\end{bmatrix}\begin{bmatrix}-4 & -1 & 0 & 5\\ 1.5 & 7.5 & -3.5 & -5.5\end{bmatrix}$. In this manner, we have calculated PCs for all the CNTs, as presented in the final table and illustrated in the last figure. The figure provides clear insights, indicating that decision-making becomes more straightforward after performing PCA. In this new dimension of PC, we can roughly infer that if the data falls within the first quadrant, the diameter (D) is 6 Å; for the second quadrant, D = 4 Å; in the third quadrant, D = 7 Å; and within the fourth quadrant, D = 10 Å. The decision-making process was not as straightforward as before. Variance ratio of $PC1$ can be calculated by $VR_1 = \frac{\sum_{i=1}^{4} PC1_i^2}{\sum_{i=1}^{4} PC1_i^2 + \sum_{i=1}^{4} PC2_i^2} \approx$ 90%. So, $VR_2 = 10\%$. As $PC1$ for the considered features captures substantial variance, decisions can be effectively made using this principal component alone. The high variance in $PC1$ indicates its enriched information nature, potentially making it a powerful predictor compared to features where $PC1$ with lower variance ratios.

S6. Decision Tree Hierarchy in Random Forest Regression

In a Random Forest, regression involves combining predictions from multiple decision trees to provide a more robust and accurate outcome. Each decision tree in the forest independently predicts the target variable based on a subset of features. The final prediction is often an average or a weighted combination of these individual tree predictions. Focusing on a single decision tree within the Random Forest, it utilizes recursive splitting of feature space to make decisions. For instance, in a dataset with features n and θ predicting d, a decision tree may split the data based on conditions like 'if $n \leq 5$, it predicts $d = 10$ Å. Otherwise, if $n > 5$, it considers θ. If $\theta \leq 5.5°$, it predicts $d = 4$ Å. For $n > 5$ and $\theta > 5.5°$, it further refines predictions: $\theta \leq 7.5$ leads to $d = 7$ Å, and $\theta > 7.5$ results in $d = 6$ Å. Therefore, predicting the diameter for new data with $n = 6$ and $\theta = 14$ yields $d = 6$ Å (as, $n > 5$, $\theta > 7.5°$). This hierarchical structure allows the Random Forest to capture complex relationships in the data, combining the strength of multiple trees for robust predictions. The process is illustrated in the Fig. S6.

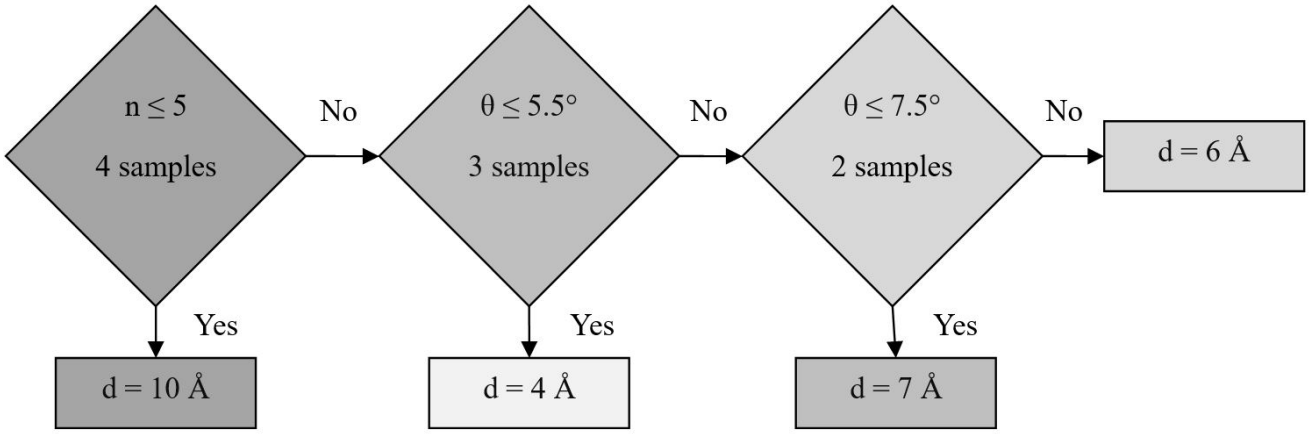

Figure S6: RF model for example problem.

Random Forest Regression, a notable application of Ensemble Learning, operates by aggregating predictions from multiple Decision Trees. The training process involves bootstrapping, where subsets of the original dataset are randomly sampled with replacements for each tree, ensuring diversity. This process, known as bagging, enhances model generalization. The testing process leverages out-of-bag (OOB) scores, utilizing the samples not included in a tree's training set for evaluation. Key features include the ability to handle non-linearity and outliers effectively. Hyperparameters, such as the number of trees, depth of trees, and minimum samples per leaf, are crucial in optimizing the model. Predictions are made by averaging or taking a majority vote of individual tree predictions. Advantages of Random Forest Regression include robustness against overfitting due to its ensemble nature, resilience in handling non-linear relationships in data, and effective management of outliers through the averaging effect. These attributes collectively make Random Forest Regression a powerful and versatile tool in the realm of machine learning regression tasks.

Table S3 listed the default hyperparameters used in the Random Forest implementation provided by the sklearn library in Python.

S7. Robustness of Model

The excellent prediction by the RF model for CNTs beyond the radius (¿ 2 nm) limit of the training dataset is demonstrated by plotting the calculated and predicted stress-strain curves of

Table S3: Default hyperparameters of Random Forest in scikit-learn

Hyperparameter	**Default Value**	**Description**
`n_estimators`	100	The number of trees in the forest.
`criterion`	"gini"	The function to measure the quality of a split: "gini" for Gini impurity, "entropy" for information gain.
`min_samples_split`	2	The minimum number of samples required to split an internal node.
`min_samples_leaf`	1	The minimum number of samples required to be at a leaf node.
`max_features`	"sqrt"	The number of features to consider when looking for the best split.
`bootstrap`	True	Whether bootstrap samples are used when building trees.

four pristine and defective CNTs as shown in Figs. S7 and S8, respectively.

References

[1] S. J. Stuart, A. B. Tutein, J. A. Harrison, A reactive potential for hydrocarbons with intermolecular interactions, The Journal of chemical physics 112 (14) (2000) 6472–6486.

[2] M. Čanaija, Deep learning framework for carbon nanotubes: Mechanical properties and modeling strategies, Carbon 184 (2021) 891–901.

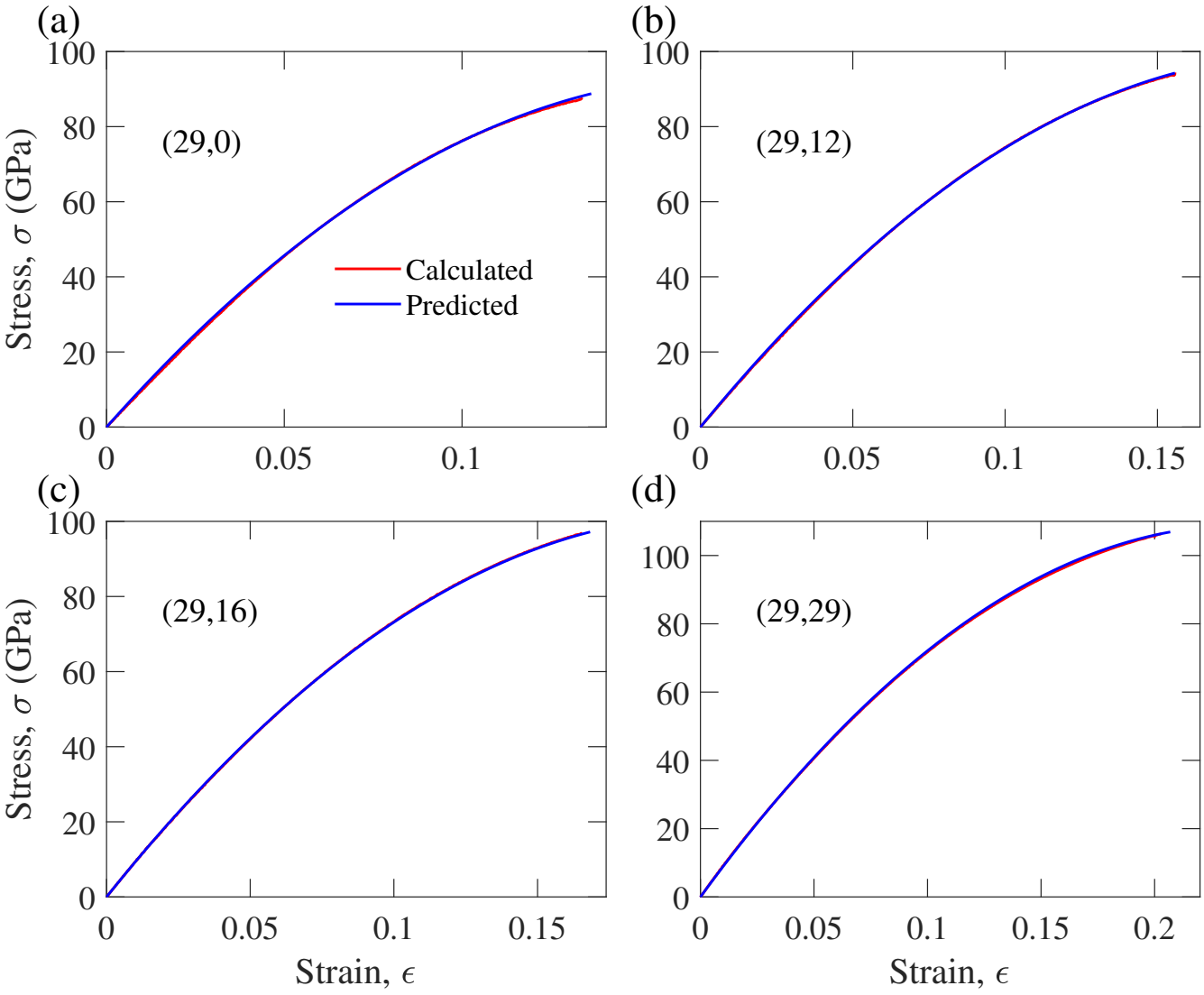

Figure S7: Comparison between the calculated and predicted stress-strain curves, where the curves corresponds to the pristine CNTs with $n = 29$ and (a) $m = 0$, (b) $m = 12$, (c) $m = 16$, (d) $m = 29$.

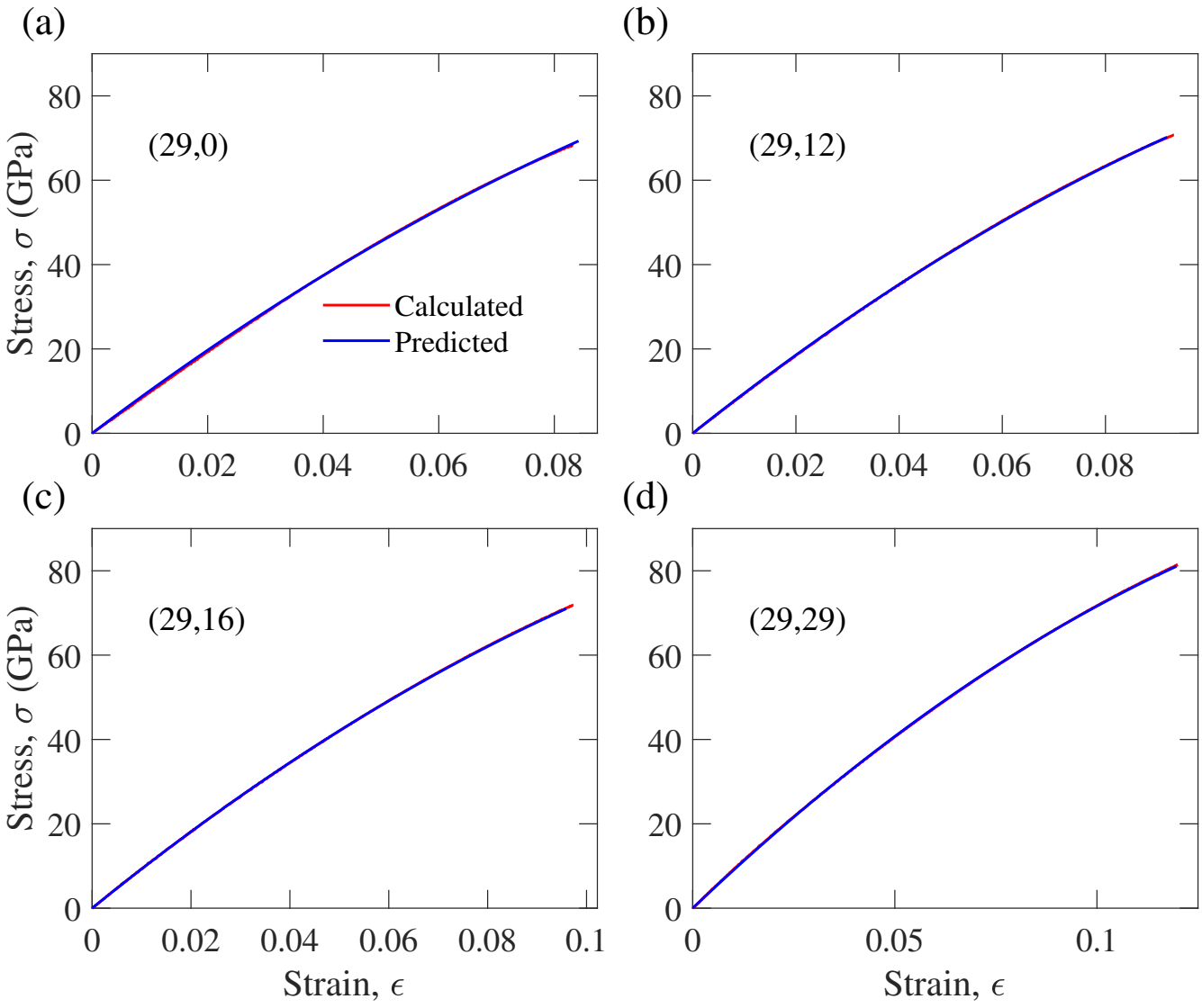

Figure S8: Comparison between the calculated and predicted stress-strain curves, where the curves corresponds to the defective CNTs with $n = 29$ and (a) $m = 0$, (b) $m = 12$, (c) $m = 16$, (d) $m = 29$.