

Artificial neural network approach for atomic coordinate prediction of carbon nanotubes

Mehmet Aci¹ · Mutlu Avcı²

Received: 15 February 2016 / Accepted: 23 May 2016
© Springer-Verlag Berlin Heidelberg 2016

Abstract In this paper, four artificial neural network (ANN) models [i.e., feed-forward neural network (FFNN), function fitting neural network (FITNET), cascade-forward neural network (CFNN) and generalized regression neural network] have been developed for atomic coordinate prediction of carbon nanotubes (CNTs). The research reported in this study has two primary objectives: (1) to develop ANN prediction models that calculate atomic coordinates of CNTs instead of using any simulation software and (2) to use results of the ANN models as an initial value of atomic coordinates for reducing number of iterations in calculation process. The dataset consisting of 10,721 data samples was created by combining the atomic coordinates of elements and chiral vectors using BIOVIA Materials Studio CASTEP (CASTEP) software. All prediction models yield very low mean squared normalized error and mean absolute error rates. Multiple correlation coefficient (R) results of FITNET, FFNN and CFNN models are close to 1. Compared with CASTEP, calculation times decrease from days to minutes. It would seem possible to predict CNTs' atomic coordinates using ANN models can be successfully used instead of mathematical calculations.

1 Introduction

Carbon nanotubes (CNTs) have been introduced as the alternatives for copper/aluminum metallic interconnects to overcome problems caused from miniaturization. CNTs are 2-D graphene crystal as rolled-up sheets. They have electronic structures depending on their direction since they are rolled up [1].

A deep impact on the investigation of material properties is made by ab initio calculations for years. Ab initio methods are parameter-free and do not require any other input than the atomic number. These are the reasons for the enormous success of ab initio methods. Also these methods are applied to a steadily increasing number of physical and chemical phenomena through improvements in computer performance and algorithms [2]. Probably, the local density functional theory proposed by Kohn and Sham [3] is the most successful method. Given the chemical composition and the crystalline structure of a periodic system, the aim of ab initio computational methods is to calculate its chemical and physical properties of as accurately as possible at a reasonable cost, without the need for empirical a priori information [4].

The formula of the ground state energy proposed by Kohn and Sham is given as

$$E = \sum_i^V \epsilon_j - \frac{1}{2} \int \frac{n(r)n(r')}{|r-r'|} dr dr' - \int V_{xc}(r)n(r)dr + E_{xc}(n(r)) \quad (1)$$

where the ϵ_j and n are the self-consistent quantities, V_{xc} is the exchange correlation potential energy, E_{xc} is the exchange correlation energy, and $n(r)$ is the electron density [5].

For calculating the properties of materials from first principles, BIOVIA Materials Studio CASTEP (CASTEP)

✉ Mehmet Aci
maci@mersin.edu.tr
Mutlu Avcı
mavci@cu.edu.tr

¹ Department of Computer Engineering, Mersin University, Mersin, Turkey

² Department of Biomedical Engineering, Çukurova University, Adana, Turkey

can be defined as a leading code. It can simulate a wide range of properties of materials proprieties using density functional theory. Some of these proprieties can be listed as structure at the atomic level, energetics, electronic response and vibrational properties [6].

Density functional theory that is the most successful method calculates atomic coordinates faster than other mathematical approaches, and it also reaches more accurate results. However, the elapsed time for calculation of high number atoms is quite long which is yielded in this study using CASTEP. Calculation may take several days due to computer's/server's power (CPU, RAM, etc.). On the other hand, users need to use more powerful machines and parallel computers which are too expensive to reduce the calculation time.

The motivation of this paper is reducing the calculation time for atomic coordinates from days to minutes. It is known that the current mathematical methods cannot reduce the calculation time up to this level. Here we approach the problem from a different angle, instead of calculation; the atomic coordinates are predicted accurately in such a short time. These predicted atomic coordinates can be used as preliminary coordinates for the simulation software, and thus, the exact atomic coordinates will be calculated within minutes or hours instead of days. Also usage of the predicted atomic coordinates in researches also helps to achieve approximate results very fast. Then we directed our studies to machine learning methods that are used in the literature for prediction problems to obtain fast and accurate results.

Some paradigms and approaches are used to convert the learning studies to machine learning. They can be briefly listed as statistical pattern recognition, symbolic processing, case-based learning, genetic algorithms, connectionist systems and evolutionary programming. The aim for the machine learning can be defined as realizing the human learning job by computers. During this learning, various methods and algorithms are used [7]. Artificial neural network (ANN) algorithms such as feed-forward neural network (FFNN), function fitting neural network (FITNET), cascade-forward neural network (CFNN) and generalized regression neural network (GRNN) are several of them.

For many problems, results can be calculated by using long and hard formulas in real or simulation environment. Generally, these situations cause long time and expensive hardware/software. Instead of this, ANN algorithms can be used to predict results of these problems using dataset that obtained from real or simulation environment. Many studies exist on various research areas in the literature, and most of them predict fast and accurate results using some ANN algorithms.

In the literature, some studies were carried out on transistors and CNTs: Cheng et al. [8] implemented a model on graphene metal-oxide-semiconductor field-

effect transistor with ANN. The computational time for the MOSFET model was decreased significantly. The model for graphene MOSFET was realized in HSPICE software as a sub-circuit, which might obviously increase the efficiency of simulations on graphene large-scale integrated circuits. In another study, for constructing mathematical models to predict mechanical properties of the carbon nanotubes/epoxy composites according to an experimental dataset, Cheng et al. [9] combined support vector regression (SVR) with particle swarm optimization for its parameter optimization.

Also in other research areas, many researchers studied on prediction using ANNs. To predict the performance measures of a message-passing multiprocessor architecture, Zayid and Akay [10] developed multilayer FFNN models. To simulate the message-passing multiprocessor architecture and create the training and testing datasets, OPNET modeler was used. Standard error of estimate (SEE) and multiple correlation coefficient are used to evaluate the performance of the multilayer FFNN prediction models. A practical method for solar irradiance forecast using ANN was presented by Mellit and Pavan [11]. They showed that, using the present values of the mean daily solar irradiance and air temperature, forecasting the solar irradiance on a base of 24 h becomes possible using the proposed multilayer perceptron (MLP) model. Sharma et al. [12] predicted the performance parameters of a single cylinder four-stroke diesel engine by an ANN at different injection timings and engine load using blended mixture of polanga biodiesel. With calculated correlation coefficient for performance parameters, the developed ANN model predicted the engine performance and exhaust emissions quite well. Xiong et al. [13] presented a study that highlights application of an ANN and a second-order regression analysis to predict bead geometry in robotic gas metal arc welding for rapid manufacturing. By using GRNN a more general forecasting method was proposed to predict the sound absorption coefficients and the average sound absorption coefficient by Liu et al. [14]. For the Greek long-term energy consumption prediction, ANNs were used by Ekonomou [15]. Ekonomou showed the accuracy after the comparison of the produced ANN results with other methods' results and real records. An ANN model was developed and applied by Ferlito et al. [16] to a real case consisting in a dataset of monthly historical building electric energy consumption. The occurred surface roughness during turning according to different cutting parameters was measured by Asilturk and Cunkas [17]. For modeling the surface roughness, ANN and multiple regression approaches were used. GRNN and conventional Box-Jenkins time series models were yielded with a comparative study by Yip et al. [18] for prediction of the maintenance cost of construction equipment. Some ANN,

regression and adaptive neuro-fuzzy inference system models were constructed, trained and tested by considering concrete constituents as input variables for predicting the 28-days compressive strength of no-slump concrete by Sobhani et al. [19]. For predicting the compressive strength of concrete that contains various amounts of blast furnace slag and fly ash, a multiple regression analysis and an ANN were applied by Atici [20]. For predicting the uniaxial compressive strength and modulus of elasticity of intact rocks, Dehghan et al. [21] used regression analysis and ANNs. The coefficients of determination to more acceptable levels improved with GRNN and FFNN.

The present study has two purposes: (1) to develop FFNN, FITNET, CFNN and GRNN prediction models for estimating the atomic coordinates of CNTs and (2) to create initial atomic coordinates using results of ANN models for decreasing number of iterations in the simulation software. Two separate datasets (i.e., input and output) are prepared for usage of prediction models. The input dataset contains 5 parameters (i.e., initial atomic coordinates u , v , w and the pair of integers (n, m) which specifies the chiral vector [22]), and the output dataset contains three parameters (calculated atomic coordinates u' , v' , w'). CASTEP is used to simulate the CNTs, calculate the geometry optimization and get output of the initial and calculated atomic coordinates for the selected chiral vector. To evaluate the performance of ANN prediction models, mean squared normalized error (MSE), mean absolute error (MAE), sum squared error (SSE) and multiple correlation coefficient (R) are calculated. It is shown that FITNET and FFNN models perform better than CFNN model. The worst performance was yielded by GRNN model.

This paper is organized as follows. Section 2 gives an overview of the CNTs. Section 3 explains the basics of ANN prediction models. Then Sect. 4 describes dataset generation and applied models. Section 5 gives the results and discussion. Finally, Sect. 6 concludes the paper.

2 Carbon nanotubes

In the solid phase, three allotropic forms exist for carbon. These forms are graphite, diamond and buckminsterfullerene [23]. As the newest forms of carbon, CNTs are discovered by Iijima in 1991 [24]. CNTs possess a unique combination of high stiffness, high strength, low density, small size and a broad range of electronic properties from metallic to p- and n-doped semiconducting [25].

CNTs have two types of form. First one is multiwalled CNT and has a structure of nested concentric tubes. The second-type CNT which is used in simulations of this study is the basic form of a rolled-up graphitic sheet and is called as single-walled CNT [25].

CNTs can be thought of as single sheets of graphite (graphene), rolled into a cylindrical shape with axial symmetry and diameters between 0.7 and 10 nm. The chiral vector describes this graphene sheet and is expressed as

$$C_h = na_1 + ma_2 \equiv (n, m) \quad (2)$$

where n and m are integer chiral indices and $|a_1| = |a_2| = 2.49 \text{ \AA}$ is the lattice constant of graphite. 'Zigzag' $(n, 0)$ and 'armchair' (n, n) nanotubes are the special cases, otherwise they called as 'chiral' (n, m) (Fig. 1). In this study, we did not use zigzag and armchair nanotubes in simulations.

Depending on the chirality, nanotubes can be semiconducting or metallic. In our simulations, we used both semiconducting and metallic CNTs according to their chiralities [25].

3 Artificial neural network models

ANNs are a family of massively parallel architectures that are capable of learning and generalizing from examples and experience [26]. A neural network has processing elements similar to the neuron in the brain. Many simple computational elements arranged in layers are consisted of processing elements. An ANN can reproduce and approximate the experimental results [27]. For forecasting and modeling of engineering problems, different types of ANN models are used in the literature which is given in Sect. 1. In this study, four ANN models are considered as follows.

3.1 Feed-forward neural networks (FFNNs) and function fitting neural networks (FITNETs)

The simplest ANN model is FFNN, and it has three layers named as input, hidden and output (Fig. 2). These networks use backpropagation learning algorithm for learning. A nonlinear activation function transfers the summation of weighted input signals. The actual observation results are compared with the response of network. Then the error of network is calculated. The calculated network error is propagated backward through the system. Afterward the weight coefficients are updated [28].

In Fig. 2, X_i is the neuron input, W_{ij} and W_{kj} are the weights, M represents the neuron number in the hidden layer, and Y represents the neuron output [29].

ANN transfer functions are the way to simulate phenomena's reaction using input and out parameters [30]. The log-sigmoid transfer function (LOGSIG) is one of the most commonly used functions (Fig. 3a). The input squashed the output into the range 0–1 by this transfer function. Hyperbolic tangent transfer function (TANSIG, Fig. 3b) in terms of neural networks, is related to a bipolar sigmoid which has an output in the range of -1 to $+1$. Pure linear

Fig. 1 Armchair, zigzag and chiral nanotube models

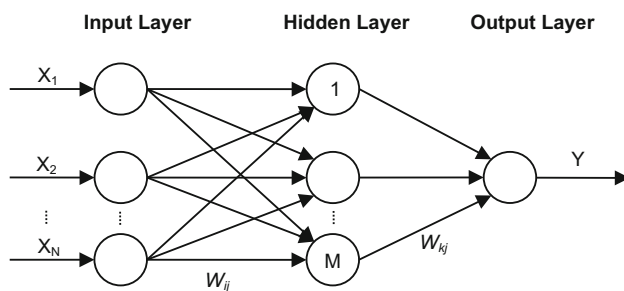
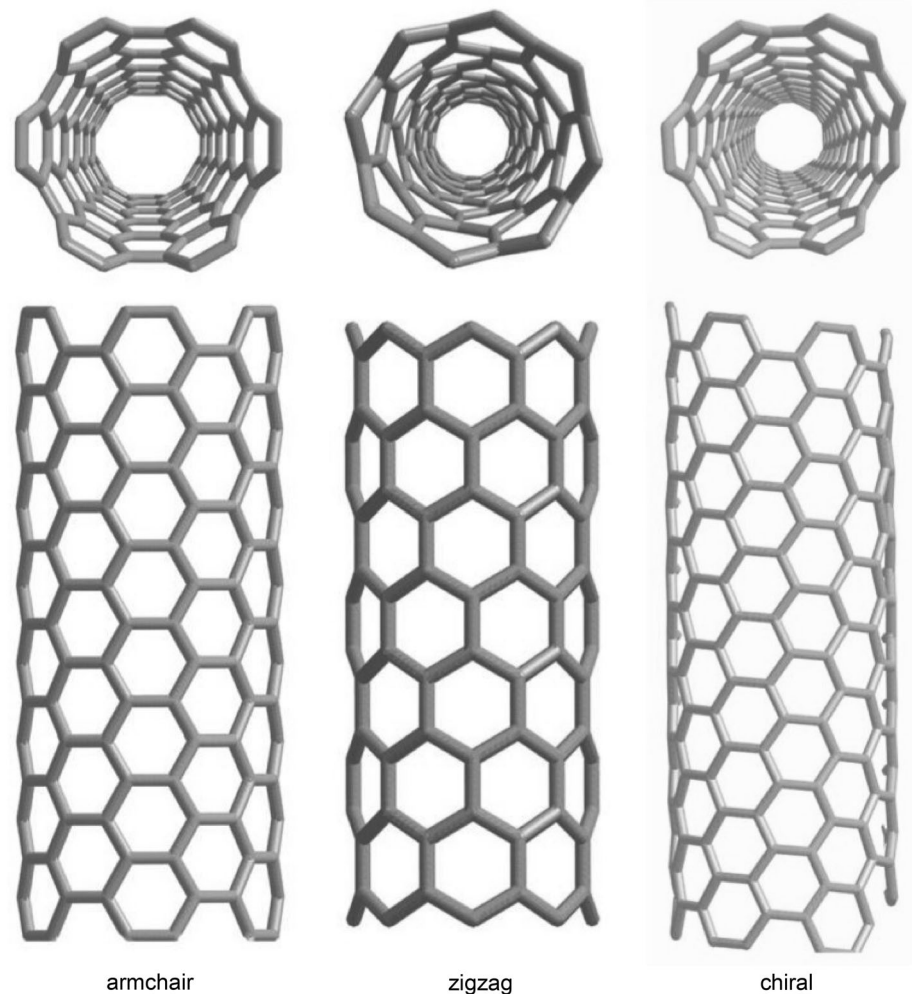


Fig. 2 Schematic diagram of FFNN

transfer function (PURELIN) (Fig. 3c) can be used to explain a linear input output relationship [31].

To fit an input–output relationship, a type of FFNN is used called as FITNET. Any finite input–output mapping problem can be fit by a FFNN with one hidden layer and enough neurons in the hidden layers. The default TANSIG transfer function is used by FITNET in the hidden layer. Also, linear transfer function is used in the output layer [29].

3.2 Cascade-forward neural networks (CFNNs)

CFNN consists of three or more layers. CFNN has one or more hidden layers, and each subsequent layer has weights and biases. Weights come from the input and all previous layers. Output layer is the last layer of the network [32]. CFNN has a weight connection from the input and every previous layer to the following layers, and this makes it different from FFNN. The speed at which ANN learns the desired relationship might be improved with additional connections (Fig. 4) [29].

3.3 Generalized regression neural network model (GRNN)

GRNN is a variation of the radial basis neural networks, and it is based on kernel regression networks. Unlike backpropagation networks, GRNN does not require an iterative training procedure. Between input and output vectors any arbitrary function is approximated by GRNN, drawing the function estimate directly from the training data [33].

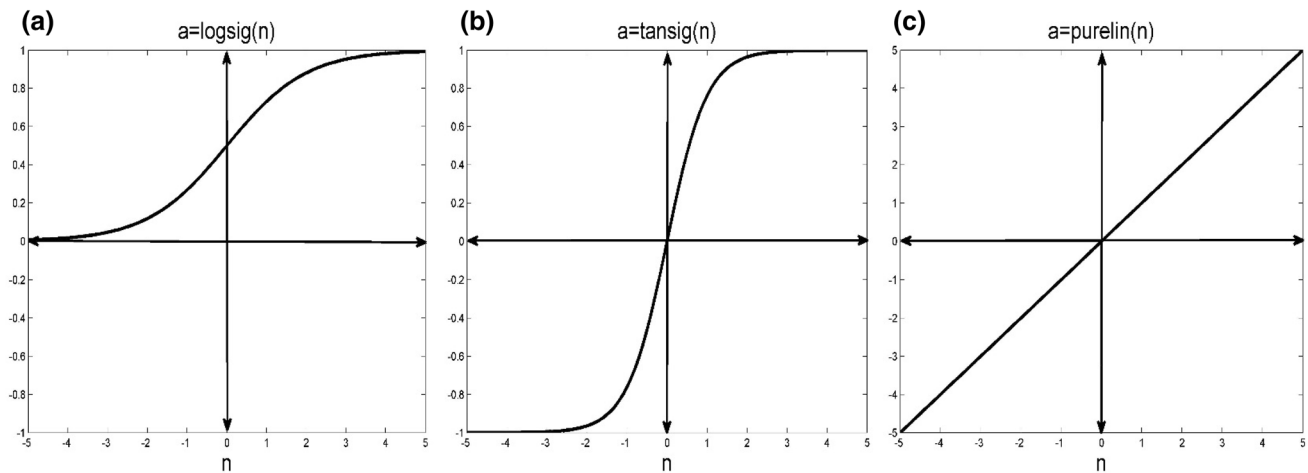


Fig. 3 Transfer functions **a** LOGSIG, **b** TANSIG, **c** PURELIN

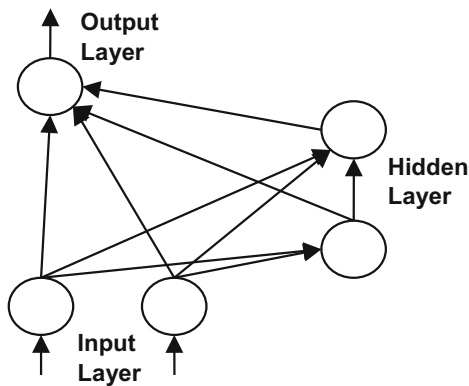


Fig. 4 Schematic diagram of CFNN

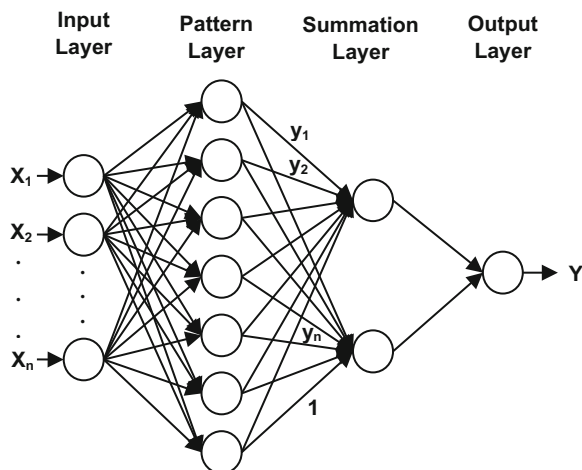


Fig. 5 Schematic diagram of GRNN

Input, pattern, summation and output layers form a GRNN as shown in Fig. 5. In the input layer, each input corresponds to individual process parameter. Input layer is fully

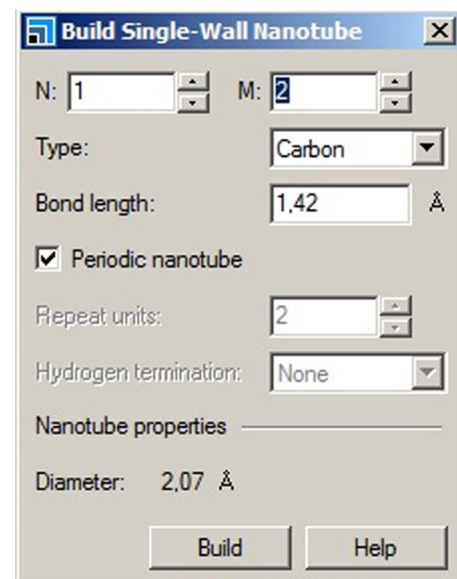


Fig. 6 Building single-wall nanotube in CASTEP

connected to pattern layer, where each unit represents a training pattern and its output is a measure of the distance of the input from the stored patterns. In the summation layer there are two neurons named as S- and D-summation neurons. Each pattern layer unit is connected to these two neurons. Sum of weighted outputs of pattern layer is computed by S-summation neuron. On the other hand, sum of unweighted outputs of pattern layer is computed by D-summation neuron. In the output layer just output of each S-summation neuron is divided by each D-summation neuron, yielding the predicted value Y'_i to an unknown input vector x as:

$$Y'_i = \frac{\sum_{i=1}^n y_i \cdot \exp[-D(x, x_i)]}{\sum_{i=1}^n \exp[-D(x, x_i)]} \quad (3)$$

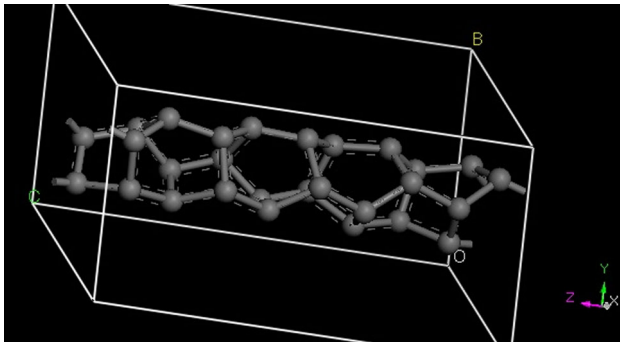


Fig. 7 A screenshot of simulated CNT in CASTEP

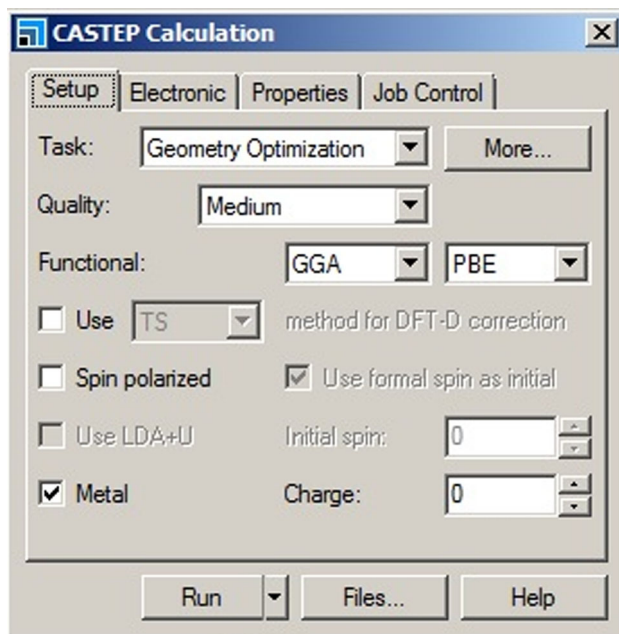


Fig. 8 Calculation interface of CASTEP

$$D(x, x_i) = \sum_{k=1}^m \left(\frac{x_i - x_{ik}}{\sigma} \right)^2 \quad (4)$$

y_i represents the weight connection between i th neuron in pattern layer and S-summation neuron and n represents the number of training patterns. Gaussian function and number of elements of an input vector are represented as D and m , respectively. x_k is j th element of x , and x_{ik} is j th element of x_i . Optimal value of spread parameter is determined experimentally, and it is represented as σ . All units in the pattern layer have the same single spread in conventional GRNN applications [33, 34].

4 Dataset generation and ANN prediction models

4.1 Dataset generation

The dataset used in this study is generated with CASTEP using CNT geometry optimization. Many CNTs are simulated in CASTEP, then geometry optimizations are calculated, and the calculated data is saved in distinct files by CASTEP. Initial coordinates of all carbon atoms are generated randomly. Figure 6 shows building step of a simulation where chiral vectors are selected before calculation (Zigzag and armchair CNTs were not used) by user. Different chiral vectors are used for each CNT simulation. The atom type is selected as carbon, bond length is used as 1.42 Å (default value), and then the nanotube is built by CASTEP. A screenshot of simulated CNT in CASTEP is shown in Fig. 7. CNT calculation parameters are used as default parameters to get general data. Figure 8 depicts a screenshot of CASTEP calculation interface.

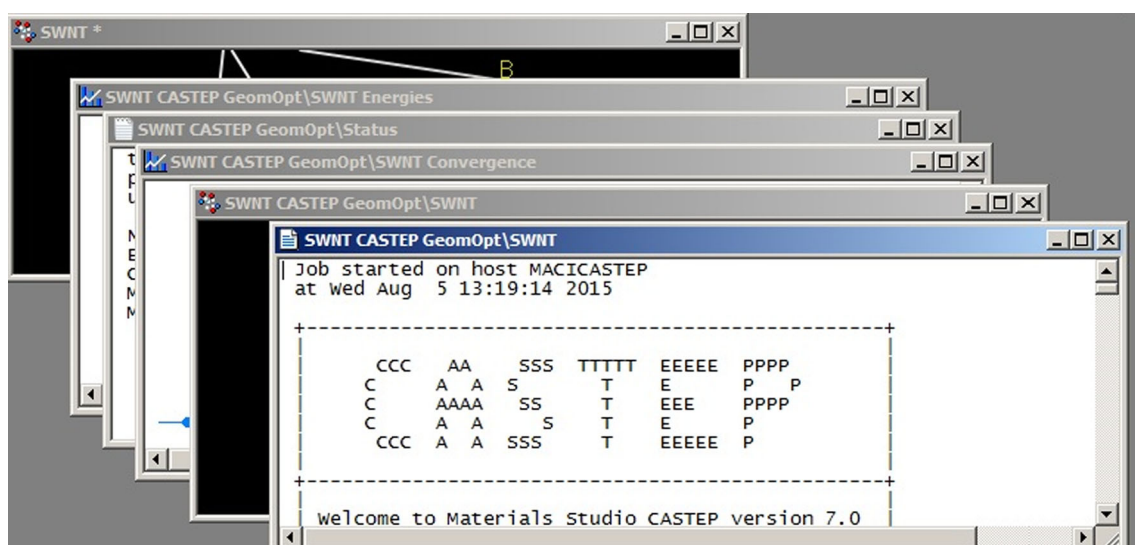


Fig. 9 Data files created by CASTEP after calculation

To finalize the computation, CASTEP uses a parameter named as `elec_energy_tol` (electrical energy tolerance) which represents that the change in the total energy from one iteration to the next remains below some tolerance value per atom for a few self-consistent field steps. This parameter also determines the calculation level of inputs and outputs. The default value of the parameter is 1×10^{-5} eV per atom and is usually suitable [6].

After the calculation step, CASTEP creates many files and saves all calculated data in these files for each CNT simulation (Fig. 9). `SWNT.castep` file is used as output file to create the input and the output datasets.

Initial atomic coordinates u , v , w and chiral vector n , m are obtained from the output files to form the input dataset. Also calculated atomic coordinates u' , v' , w' are extracted from same files to form the output dataset. The dataset consisting of 10,721 data samples firstly was divided randomly into training, validating and testing data randomly, in which there are 70–15–15 % training, validating and testing sets. After that, to determine the degree of accuracy, we used cross-validation techniques that compared the actual values with the estimated values. The datasets are evaluated by means of tenfold cross-validation. Cross-validation predicts the average error of estimates by using the dataset with one individual removed. Cross-validated datasets consist of 80 % of the training data and 20 % of the test data. The generated prediction models are evaluated by whether using tenfold cross-validation or without using cross-validation.

The calculations are performed on a server that has 2.00 Ghz CPU power on 2 processors with 4 cores and 8 GB RAM. The calculation time according to these calculations are given in seconds in Table 1. A summary of the descriptive statistics for the dataset is given in Table 2.

4.2 ANN prediction models

FFNN, FITNET, CFNN and GRNN models were trained and tested with the dataset of atomic coordinates for CNTs. The FFNN, FITNET and CFNN models have three layers (input, hidden and output), and the input and output layers have 5 and 3 neurons, respectively (Figs. 10, 11, 12). The hidden layers of FITNET and CFNN models have 10 neurons, and FFNN model's hidden layer has 20 neurons. A LOGSIG activation function is used in FFNN model, and the TANSIG activation function is used in FITNET and CFNN models in the hidden layer. A pure linear activation function is used in all models in the output layer, and Levenberg–Marquardt algorithm [35] is utilized for training the networks. Weights and biases were randomly initialized. The other important parameters of the FFNN and FITNET models are the number of epochs (1000), the learning rate (0.02) and momentum (0.5). Different

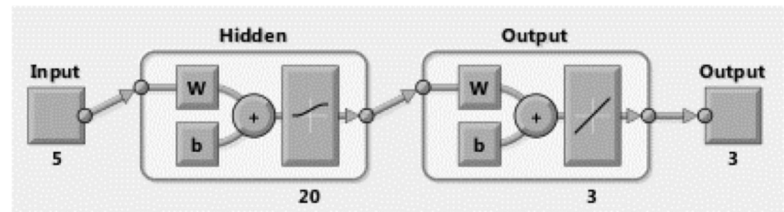
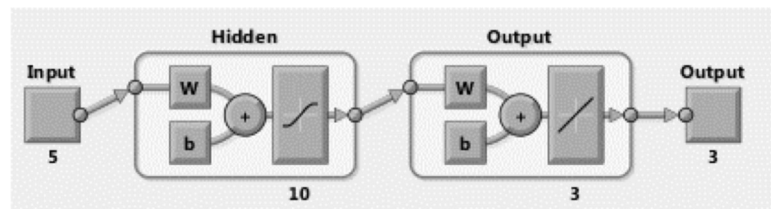
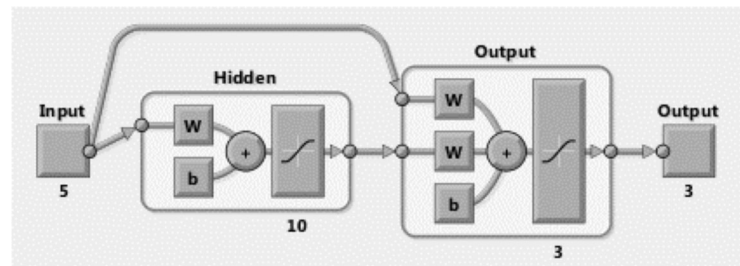
Table 1 Calculation time for CNT simulations in seconds

ID of CNT	Number of atoms	Calculation time (s)
1	28	1050.14
2	52	1256.4
3	76	2254.18
4	84	2816.61
5	56	834.04
6	148	18,496.07
7	124	33,280.8
8	156	9604.28
9	196	59,998.78
10	244	19,389.07
11	172	64,102.59
12	104	7207.22
13	84	6842.67
14	152	7715.19
15	364	78,421.39
16	228	13,513.07
17	268	27,799.53
18	316	35,982.11
19	372	94,767.02
20	436	266,027.27
21	508	233,190.94
22	292	27,792.55
23	168	13,292.91
24	388	106,542.26
25	112	6838.91
26	516	201,214.72
27	296	34,136.32
28	364	103,759.38
29	412	95,275.6
30	156	9558.96
31	532	427,956.63
32	228	27,392.57
33	444	159,893.39
34	248	74,997.74
35	312	40,999.97
36	140	10,014.92
37	392	83,173.05
38	588	397,369.36
39	344	89,653.65
40	252	26,127.05
41	208	23,123.48
42	168	11,826.42

network parameters during the training have been tried to get the overall best performance (average accuracy) over test sets. These numbers have been obtained by trial and error. All ANN-based prediction models are coded in MATLAB [29].

Table 2 Descriptive statistics

	u Input	v Input	w Input	m	n	u' Output	v' Output	w' Output
Min.	0.0451	0.0451	0	2	1	0.0385	0.0389	0
Max.	0.9548	0.9548	0.9999	12	6	0.9614	0.9610	0.9996
Mean	0.5000	0.5000	0.4994	8.2252	0.3378	0.5000	0.4999	0.4993
SD	0.2900	0.2900	0.2900	2.1400	1.6800	0.2909	0.2910	0.2884

Fig. 10 FFNN model with 5, 20 and 3 neurons**Fig. 11** FITNET model with 5, 10 and 3 neurons**Fig. 12** CFNN model with 5, 10 and 3 neurons

The GRNN model's only parameter σ (also called 'spread' in MATLAB software) determines the generalization capability of the GRNN. The spread parameter is adopted as 2.

5 Results and discussion

This section presents performance comparisons of the ANN prediction models and performance improvement of CASTEP simulation after using predicted data. All prediction models are evaluated in terms of four performances measured; (1) R value (coefficient of correlation) is used for measuring correlation between target and predicted values, (2) MSE measures the average of the squares of the errors, (3) MAE measures how close predictions are to the target values, and (4) SSE calculates the sum of the squared errors of the prediction models. Summaries of

mathematical equations of these performance measures are given in Eqs. (5)–(8), respectively.

$$R = \sqrt{1 - \frac{\sum_{i=1}^n (O_i - P_i)^2}{\sum_{i=1}^n (O_i - O_m)^2}} \quad (5)$$

$$\text{MSE} = \left[\sum_{i=1}^n \left(\frac{O_i - P_i}{n} \right)^2 \right] \quad (6)$$

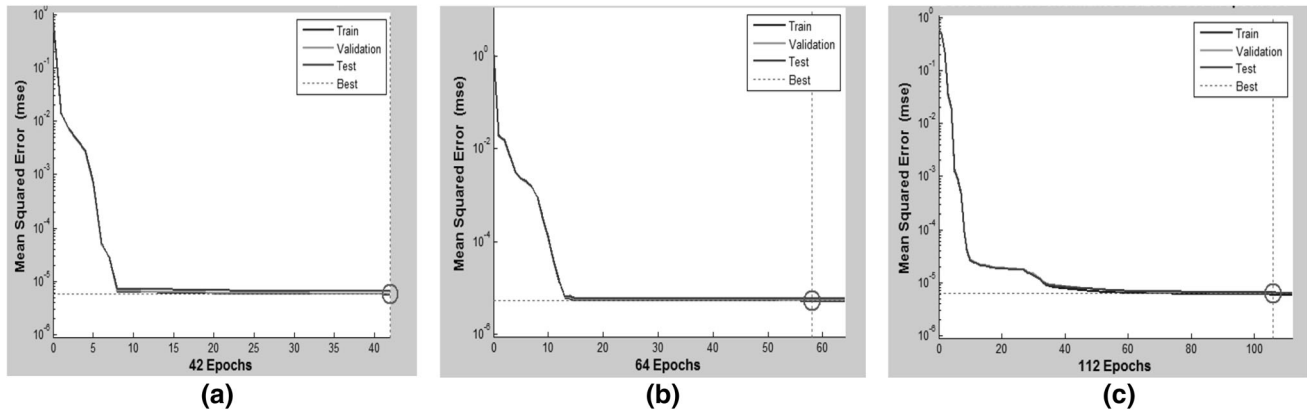
$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |O_i - P_i| \quad (7)$$

$$\text{SSE} = \sum_{i=1}^n (O_i - P_i)^2 \quad (8)$$

where n is the number of data points used for testing, P_i is the predicted value, O_i is the observed value, and O_m is the average of the observed values.

Table 3 Performance results of all prediction models on the non-cross-validated dataset

	FFNN	FITNET	CFNN	GRNN
MSE	6.049476e−006	6.048698e−006	6.497111e−006	7.839317e−002
MAE	1.376493e−003	1.393361e−003	1.576503e−003	2.482294e−001
SSE	2.593410e−002	2.593077e−002	2.785311e−002	7.360715e+001
R	9.999638e−001	9.999635e−001	9.999608e−001	9.096695e−001

**Fig. 13** Learning curve of the prediction models and its relation with validation and test sets. **a** FFNN, **b** FITNET, **c** CFNN

5.1 Performance results of ANN prediction models

Table 3 shows prediction performance of the models in terms of MSE, MAE, SSE and R on non-cross-validated dataset. The results of the prediction models are compared with each other. Figure 13 shows the training curves of FFNN-, FITNET- and CFNN-based prediction models. There is not any training curve for GRNN-based prediction model because the model does not have training phase.

Tables 4, 5, 6 and 7 show the performance of all prediction models using tenfold cross-validation. In Table 8, the overall comparison of the prediction models on tenfold cross-validated dataset is shown.

When the cross-validation is not considered, the following inferences can be made from Table 3 and Fig. 13.

- The results of FFNN-based and FITNET-based models are nearly the same for all performance measures. This result can be interpreted as saying that FITNET-based prediction model is a kind of FFNN-based prediction model. Also, the best results belong to these two models.
- CFNN-based prediction model gives better results than GRNN-based prediction model for all performance measures.
- Figure 13 shows that the best validation performance is achieved from FFNN-, FITNET- and CFNN-based

Table 4 Results of FFNN-based prediction model by means of tenfold cross-validation

Fold number	MSE	MAE	SSE	R
1	6.774062e−006	1.482343e−003	2.032219e−002	9.999590e−001
2	7.680725e−006	1.602885e−003	2.304217e−002	9.999532e−001
3	5.318196e−006	1.357533e−003	1.595459e−002	9.999686e−001
4	6.152468e−006	1.409395e−003	1.845740e−002	9.999628e−001
5	5.346127e−006	1.337258e−003	1.603838e−002	9.999679e−001
6	8.436487e−006	1.892817e−003	2.530946e−002	9.999481e−001
7	6.265620e−006	1.494962e−003	1.879686e−002	9.999623e−001
8	7.929273e−006	1.842293e−003	2.378782e−002	9.999526e−001
9	6.569607e−006	1.437812e−003	1.970882e−002	9.999606e−001
10	6.325530e−006	1.508314e−003	1.897659e−002	9.999618e−001
Average	6.679809e−006	1.536561e−003	2.003943e−002	9.999597e−001

The best and average results are outlined in bold

Table 5 Results of FITNET-based prediction model by means of tenfold cross-validation

Fold number	MSE	MAE	SSE	R
1	6.589269e−006	1.557503e−003	1.976781e−002	9.999608e−001
2	5.861648e−006	1.429010e−003	1.758494e−002	9.999648e−001
3	7.277767e−006	1.548961e−003	2.183330e−002	9.999561e−001
4	6.709909e−006	1.471574e−003	2.012973e−002	9.999589e−001
5	6.301777e−006	1.396791e−003	1.890533e−002	9.999622e−001
6	7.128428e−006	1.514411e−003	2.138528e−002	9.999572e−001
7	7.168583e−006	1.537053e−003	2.150575e−002	9.999571e−001
8	6.216803e−006	1.422944e−003	1.865041e−002	9.999626e−001
9	6.011725e−006	1.449355e−003	1.803517e−002	9.999642e−001
10	6.991452e−006	1.460794e−003	2.097436e−002	9.999582e−001
Average	6.625736e−006	1.478840e−003	1.987721e−002	9.999602e−001

The best and average results are outlined in bold

Table 6 Results of CFNN-based prediction model by means of tenfold cross-validation

Fold number	MSE	MAE	SSE	R
1	8.374742e−006	1.851975e−003	2.512423e−002	9.999499e−001
2	9.096778e−006	1.998724e−003	2.729033e−002	9.999444e−001
3	8.724822e−006	1.655123e−003	2.617447e−002	9.999468e−001
4	7.400171e−006	1.796050e−003	2.220051e−002	9.999558e−001
5	1.584400e−005	2.843949e−003	4.753199e−002	9.999038e−001
6	1.966750e−005	3.020017e−003	5.900249e−002	9.998831e−001
7	1.295598e−005	2.512414e−003	3.886793e−002	9.999229e−001
8	9.541929e−006	1.978344e−003	2.862579e−002	9.999430e−001
9	8.099411e−006	1.702325e−003	2.429823e−002	9.999521e−001
10	7.964141e−006	1.948631e−003	2.389242e−002	9.999525e−001
Average	1.076695e−005	2.130755e−003	3.230084e−002	9.999354e−001

The best and average results are outlined in bold

Table 7 Results of GRNN-based prediction model by means of tenfold cross-validation

Fold number	MSE	MAE	SSE	R
1	7.680943e−002	2.227795e−001	7.680943e+001	9.427723e−001
2	7.855592e−002	2.237748e−001	7.680943e+001	9.405304e−001
3	7.748048e−002	2.227352e−001	7.748048e+001	9.418140e−001
4	7.993998e−002	2.261713e−001	7.993998e+001	9.422623e−001
5	8.096245e−002	2.266645e−001	8.096245e+001	9.409418e−001
6	7.749710e−002	2.226202e−001	7.749710e+001	9.420573e−001
7	7.781022e−002	2.240045e−001	7.781022e+001	9.396740e−001
8	7.920970e−002	2.236537e−001	7.920970e+001	9.411874e−001
9	7.791531e−002	2.255187e−001	7.791531e+001	9.408844e−001
10	7.617100e−002	2.224810e−001	7.617100e+001	9.403550e−001
Average	7.823516e−002	2.240403e−001	7.773600e+001	9.412479e−001

The best and average results are outlined in bold

Table 8 Overall comparison of the cross-validated prediction models

	FFNN	FITNET	CFNN	GRNN
Average MSE	6.679809e−006	6.625736e−006	1.076695e−005	7.823516e−002
Average MAE	1.536561e−003	1.478840e−003	2.130755e−003	2.240403e−001
Average SSE	2.003943e−002	1.987721e−002	3.230084e−002	7.773600e+001
Average R	9.999597e−001	9.999602e−001	9.999354e−001	9.412479e−001

prediction models at epoch 42, 64 and 112, respectively. It shows that these prediction models can essentially estimate close to the target values. The best performances are not obtained very early in the training phase; therefore, the prediction models can be generalized.

- All prediction models show relevant results to the models which are tested by tenfold cross-validation (Tables 4, 5, 6, 7).

Based on the results, the following points can be made as results of the average values in Table 8:

- For all MSE, MAE, SEE and R performance measures, FITNET-based prediction model performs better than FFNN-, CFNN- and GRNN-based prediction models.
- With a general view FITNET-, FFNN- and CFNN-based prediction models perform good levels of successes.
- When we look at all results, FITNET-based prediction model performs best and FFNN-based prediction model follows it with a small margin.
- GRNN-based prediction model performs with high errors than others and is not a good predictor for this problem.

As an important point, if we look through the ten folds for each model in detail, the following points can be made:

- The best MSE, SSE and R values are achieved with FFNN-based prediction model as $5.318196\text{e}-006$, $1.595459\text{e}-002$ and $9.999686\text{e}-001$, respectively, at third fold.
- The best MAE value is achieved with FFNN-based prediction model as $1.337258\text{e}-003$ at fifth fold.
- Above points show that FFNN-based prediction model achieves best performance measures in all folds.

Also the following points can be made for each model:

- FFNN model achieves best results at fifth fold for MAE and at third fold for MSE, SSE and R performance measures.
- FITNET model achieves best results at fifth fold for MAE and at second fold for MSE, SSE and R performance measures.
- CFNN model achieves best results at third fold for MAE and at fourth fold for MSE, SSE and R performance measures.
- At first fold for *R* and at tenth fold for MSE, MAE and SSE performance measures' best results are achieved by GRNN model.
- According to the above points, generally best performance measures are achieved at same folds.

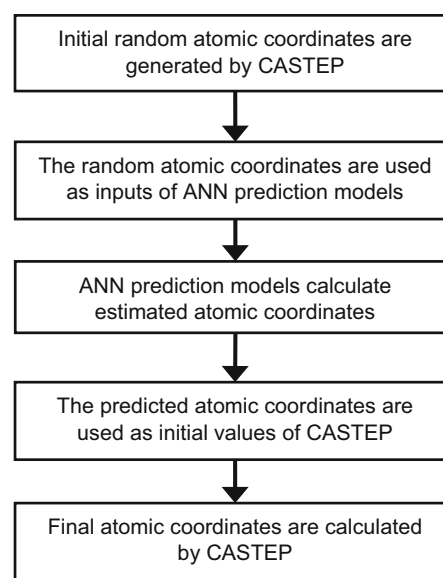


Fig. 14 Steps of the two-staged CASTEP simulation

5.2 Performance improvement of CASTEP simulation

Another usage of these predicted data is to decrease number of iterations in calculation process in CASTEP. If CASTEP simulation starts with random atomic coordinates, the calculation may take several days. The number of iterations in CASTEP can be decreased from days to minutes by using prediction results of ANN models as initial atomic coordinates. The design of this approach is depicted in Fig. 14. The random values of atomic coordinates are used as inputs of ANN prediction models, and after that the predicted values are used to calculate final atomic coordinates by CASTEP (two-staged simulation). Table 9 shows the number of iterations before and after using the initial values that are predicted by FITNET, FFNN, CFNN and GRNN models. The rate of decline in iteration number is related to performances of ANN prediction models. The best decline is achieved by FITNET-based model, and the worst is GRNN-based model. Table 10 shows percentage of declines in number of iterations. As expected, the highest percentage of decline reached up to $\approx 85\%$ by FITNET-based model.

6 Conclusions

In this study, ANN-based models are proposed to predict the atomic coordinates of CNTs. The predicted atomic coordinates can be used instead of simulated ones, or they can be the initial values of the simulation. In order to carry out these prediction models, a dataset is created using

Table 9 Number of iterations before and after the two-staged CASTEP simulation

ID of CNT	Number of atoms	Number of iteration before ANN	Number of iteration after FITNET	Number of iteration after FFNN	Number of iteration after CFNN	Number of iteration after GRNN
1	28	26	4	5	7	9
2	52	18	3	4	5	7
3	76	13	3	3	4	6
4	84	13	3	4	5	7
5	56	6	2	2	3	4
6	148	19	4	5	5	8
7	124	57	12	13	15	21
8	156	8	2	2	2	4
9	196	23	5	5	6	8
10	244	11	3	3	4	6
11	172	32	7	8	8	14
12	104	11	4	4	5	7
13	84	34	7	7	9	16
14	152	15	4	5	5	9
15	364	12	4	4	4	6
16	228	10	3	4	4	6
17	268	14	3	3	5	7
18	316	11	3	3	3	6
19	372	18	4	5	6	7
20	436	31	6	7	9	12
21	508	17	4	4	5	9
22	292	10	3	3	3	5
23	168	22	6	5	7	13
24	388	17	5	6	6	8
25	112	18	5	5	6	8
26	516	15	5	5	6	8
27	296	12	4	3	4	7
28	364	23	7	7	8	10
29	412	14	3	3	3	5
30	156	22	5	5	7	10
31	532	33	8	9	10	17
32	228	21	5	6	7	14
33	444	19	5	5	7	10
34	248	52	14	16	17	28
35	312	13	4	4	4	6
36	140	15	5	6	6	8
37	392	13	5	5	5	6
38	588	16	5	5	6	7
39	344	18	6	5	7	9
40	252	11	3	4	4	7
41	208	18	5	5	6	8
42	168	32	7	7	9	14

CASTEP. The dataset contains five inputs (initial atomic coordinates u , v , w and the chiral vector n , m) and three outputs (calculated atomic coordinates u' , v' , w'). MSE, MAE, SSE and R values of the developed models have

been calculated for evaluation of the models. FITNET-, FFNN- and CFNN-based prediction models perform good levels of successes. For all average MSE, MAE, SSE and R performance measures, FITNET-based prediction model

Table 10 Percentage of declines in number of iterations

ID of CNT	Number of atoms	Number of iteration before ANN	Percentage of decline after FITNET (%)	Percentage of decline after FFNN (%)	Percentage of decline after CFNN (%)	Percentage of decline after GRNN (%)
1	28	26	84.62	80.77	73.08	65.38
2	52	18	83.33	77.78	72.22	61.11
3	76	13	76.92	76.92	69.23	53.85
4	84	13	76.92	69.23	61.54	46.15
5	56	6	66.67	66.67	50.00	33.33
6	148	19	78.95	73.68	73.68	57.89
7	124	57	78.95	77.19	73.68	63.16
8	156	8	75.00	75.00	75.00	50.00
9	196	23	78.26	78.26	73.91	65.22
10	244	11	72.73	72.73	63.64	45.45
11	172	32	78.13	75.00	75.00	56.25
12	104	11	63.64	63.64	54.55	36.36
13	84	34	79.41	79.41	73.53	52.94
14	152	15	73.33	66.67	66.67	40.00
15	364	12	66.67	66.67	66.67	50.00
16	228	10	70.00	60.00	60.00	40.00
17	268	14	78.57	78.57	64.29	50.00
18	316	11	72.73	72.73	72.73	45.45
19	372	18	77.78	72.22	66.67	61.11
20	436	31	80.65	77.42	70.97	61.29
21	508	17	76.47	76.47	70.59	47.06
22	292	10	70.00	70.00	70.00	50.00
23	168	22	72.73	77.27	68.18	40.91
24	388	17	70.59	64.71	64.71	52.94
25	112	18	72.22	72.22	66.67	55.56
26	516	15	66.67	66.67	60.00	46.67
27	296	12	66.67	75.00	66.67	41.67
28	364	23	69.57	69.57	65.22	56.52
29	412	14	78.57	78.57	78.57	64.29
30	156	22	77.27	77.27	68.18	54.55
31	532	33	75.76	72.73	69.70	48.48
32	228	21	76.19	71.43	66.67	33.33
33	444	19	73.68	73.68	63.16	47.37
34	248	52	73.08	69.23	67.31	46.15
35	312	13	69.23	69.23	69.23	53.85
36	140	15	66.67	60.00	60.00	46.67
37	392	13	61.54	61.54	61.54	53.85
38	588	16	68.75	68.75	62.50	56.25
39	344	18	66.67	72.22	61.11	50.00
40	252	11	72.73	63.64	63.64	36.36
41	208	18	72.22	72.22	66.67	55.56
42	168	32	78.13	78.13	71.88	56.25

performs better than the other prediction models. The best performance measure results are achieved with FFNN-based prediction model which tested tenfold cross-validated dataset. If the cross-validation is not used, FFNN-

based and FITNET-based prediction models show nearly the same performance. When the predicted results are used as preliminary values to calculate exact results in CASTEP, the number of iterations is decreased up to $\approx 85\%$ of their

first value. The use of ANN ensures that the exact results of atomic coordinates can be obtained in a short time. Future research can be performed in two areas: The first one would be expanding the number of parameters in the dataset. The second area would be using new models like support vector machine and regression trees to predict the atomic coordinates.

Acknowledgments We would like to thank Cukurova University Scientific Research Projects Center for supporting this work (Project code: FDK-2015-3170).

References

1. S. Yamacli, M. Avci, Phys. Lett. A **374**, 297 (2009)
2. G. Kresse, J. Furthmüller, Comput. Mater. Sci. **6**, 15 (1996)
3. W. Kohn, L.J. Sham, Phys. Rev. **140**, A1133 (1965)
4. C. Pisani (ed.), *Quantum-Mechanical Ab-Initio Calculation of the Properties of Crystalline Materials (Lecture Notes in Chemistry)*, 1st edn. (Springer, Berlin, 1996)
5. W. Kohn, A.D. Becke, R.G. Parr, J. Phys. Chem. **100**, 12974 (1996)
6. BIOVIA Materials Studio. <http://accelrys.com/products/data-sheets/castep.pdf>
7. M. Aci, C. Inan, M. Avci, Exp. Syst. Appl. **37**, 5061 (2010)
8. G. Cheng, Q. Ji, H. Wu, Q. Zhao, X. Qiang, in *International Conference on Mechatronics, Electronic Industrial and Control Engineering* (2015), pp. 1479–1483
9. W.D. Cheng, C.Z. Cai, Y. Luo, Y.H. Li, C.J. Zhao, Mod. Phys. Lett. B **29**, 1550016 (2015)
10. E.I.M. Zayid, M.F. Akay, Neural Comput. Appl. **23**, 2481 (2012)
11. A. Mellit, A.M. Pavan, Sol. Energy **84**, 807 (2010)
12. A. Sharma, P.K. Sahoo, R.K. Tripathi, L.C. Meher, Int. J. Ambient Energy **36**, 1 (2015)
13. J. Xiong, G. Zhang, J. Hu, L. Wu, J. Intell. Manuf. **25**, 157 (2012)
14. J. Liu, W. Bao, L. Shi, B. Zuo, W. Gao, Appl. Acoust. **76**, 128 (2014)
15. L. Ekonomou, Energy **35**, 512 (2010)
16. S. Ferlito, M. Atrigna, G. Graditi, S. De Vito, M. Salvato, A. Buonanno, G. Di Francia, in *2015 XVIII AISEM Annual Conference* (IEEE, 2015), pp. 1–4
17. İ. Asiltürk, M. Çunkaş, Exp. Syst. Appl. **38**, 5826 (2011)
18. H. Yip, H. Fan, Y. Chiang, Autom. Constr. **38**, 30 (2014)
19. J. Sobhani, M. Najimi, A.R. Pourkhorshidi, T. Parhizkar, Constr. Build. Mater. **24**, 709 (2010)
20. U. Atici, Exp. Syst. Appl. **38**, 9609 (2011)
21. S. Dehghan, G. Sattari, S. Chehreh Chelgani, M. Aliabadi, Min. Sci. Technol. **20**, 41 (2010)
22. M.S. Dresselhaus, G. Dresselhaus, P. Avouris, *Carbon Nanotubes: Synthesis, Structure, Properties and Applications* (Springer, Berlin, 2001)
23. M.J. O'Connell, *Carbon Nanotubes: Properties and Applications*, 1st edn. (Taylor & Francis, New York, 2006)
24. S. Iijima, Nature **354**, 56 (1991)
25. A. Kis, A. Zettl, Philos. Trans. A. Math. Phys. Eng. Sci. **366**, 1591 (2008)
26. S. Hakim, *Development of Artificial Neural Network with Application to Some Civil Engineering Problems*, University Putra Malaysia, 2006
27. J. Ghaboussi, J.H. Garrett Jr., X. Wu, J. Eng. Mech. **117**, 132 (1991)
28. Z. Sen, in *Water Found. Publ.* (Istanbul, 2004)
29. M.H. Beale, M.T. Hagan, H.B. Demuth, R2012a, MathWorks, Inc., 3 Apple Hill Drive Natick, MA 01760-2098. www.Mathworks.Com (2012)
30. S.R. Patil, *Regionalization of an Event Based Nash Cascade Model for Flood Predictions in Ungauged Basins*, Universität Stuttgart (2008)
31. M. Dorofki, A.H. Elshafie, O. Jaafar, O. Karim, in *International Proceedings of Chemical, Biological and Environ* (2012), p. 39
32. A. Hedayat, H. Davilu, A.A. Barfrosh, K. Sepanloo, Prog. Nucl. Energy **51**, 709 (2009)
33. B. Kim, D.W. Lee, K.Y. Park, S.R. Choi, S. Choi, Vacuum **76**, 37 (2004)
34. J.-S.R. Jang, C.-T. Sun, *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*, 1st edn. (Prentice Hal, New Jersey, 1997)
35. J.J. More, Numer. Anal. Lect. Notes Math. **630**, 105 (1978)