**ÇUKUROVA UNIVERSITY**
**INSTITUTE OF NATURAL AND APPLIED SCIENCES**

**PhD THESIS**

**Mehmet ACI**

# PREDICTION OF CARBON NANOTUBE ATOMIC COORDINATES BASED ON MACHINE LEARNING ALGORITHMS

**DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING**

**ADANA, 2016**

**ÇUKUROVA UNIVERSITY**
**INSTITUTE OF NATURAL AND APPLIED SCIENCES**

# PREDICTION OF CARBON NANOTUBE ATOMIC COORDINATES BASED ON MACHINE LEARNING ALGORITHMS

**Mehmet ACI**

**PhD THESIS**

**DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING**

We certify that the thesis titled above was reviewed and approved for the award of degree of the Doctor of Philosophy by the board of jury on 25/11/2016.

| | | |
|---|---|---|
| ……..... | ..... | ..... |
| Assoc.Prof.Dr. Mutlu AVCI | Prof.Dr. Tülay YILDIRIM | Prof.Dr. Mustafa GÖK |
| SUPERVISOR | MEMBER | MEMBER |

| | |
|---|---|
| ……..... | …… |
| Prof.Dr. Faruk KARADAĞ | Assoc.Prof.Dr. Zeki YETGİN |
| MEMBER | MEMBER |

This PhD Thesis is performed in Department of Electrical and Electronics Engineering of Institute of Natural and Applied Sciences of Çukurova University.
**Registration Number**:

**Prof. Dr. Mustafa GÖK**
**Director**
**Institute of Natural and Applied Sciences**

**This study was supported by Çukurova University Research Fund.**
**Project Number: FDK-2015-3170**

**Note:** The usage of the presented specific declarations, tables, figures, and photographs either in this thesis or in any other reference without citation is subject to "The law of Arts and Intellectual Products" number of 5846 of Turkish Republic.

***To Çiğdem...***

## ABSTRACT

## PhD THESIS

**PREDICTION OF CARBON NANOTUBE ATOMIC COORDINATES BASED ON MACHINE LEARNING ALGORITHMS**

**Mehmet ACI**

**ÇUKUROVA UNIVERSITY**
**INSTITUTE OF NATURAL AND APPLIED SCIENCES**
**DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING**

Supervisor : Assoc. Prof. Dr. Mutlu AVCI
Year: 2016, Pages: 101
Jury : Assoc. Prof. Dr. Mutlu AVCI
: Prof. Dr. Tülay YILDIRIM
: Prof. Dr. Mustafa GÖK
: Prof. Dr. Faruk KARADAĞ
: Assoc. Prof. Dr. Zeki YETGİN

In this thesis, seven prediction models (i.e. Feed Forward Neural Network (FFNN), Function Fitting Neural Network (FITNET), Cascade-Forward Neural Network (CFNN), Generalized Regression Neural Network (GRNN), Support Vector Regression (SVR), Classification and Regression Tree (CART) and Multiple Linear Regression (MLR)) have been developed for atomic coordinate prediction of carbon nanotubes. The main aim of this research is to reduce the calculation time for atomic coordinates from days to minutes using developed prediction models. The dataset was created by combining the atomic coordinates of elements and chiral vectors using CASTEP software. The accuracy of the proposed models is evaluated with Mean Square Error (MSE), Mean Absolute Error (MAE), Standard Error of the Estimate and Correlation Coefficient metrics. The dataset is studied separately with and without using 10-fold cross-validation. FFNN, CFNN and FITNET prediction models yielded very high performance by means of MSE and MAE. These models are followed by CART, SVR, GRNN and MLR, respectively. The results obtained from this study can be used in two ways: i) The atomic coordinates can be predicted with a low-error without using a simulation program, ii) The estimated results can be used as an initial value of simulation software for reducing duration of the atomic coordinate simulation seriously.



**Key Words**: Atomic Coordinate, Carbon Nanotube, Prediction, Machine Learning

# ÖZ

## DOKTORA TEZİ

**KARBON NANOTÜP ATOMİK KOORDİNATLARININ MAKİNE ÖĞRENMESİ ALGORİTMALARI TABANLI TAHMİNİ**

**Mehmet ACI**

**ÇUKUROVA ÜNİVERSİTESİ**
**FEN BİLİMLERİ ENSTİTÜSÜ**
**ELEKTRİK VE ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI**

Danışman : Doç. Dr. Mutlu AVCI
Yıl: 2016, Sayfa: 101
Jüri : Doç. Dr. Mutlu AVCI
: Prof. Dr. Tülay YILDIRIM
: Prof. Dr. Mustafa GÖK
: Prof. Dr. Faruk KARADAĞ
: Doç. Dr. Zeki YETGİN

Bu tezde, karbon nanotüplerin atomik koordinatlarını tahmin etmek için yedi adet tahmin modeli (İleri Beslemeli (İBYSA), Fonksiyon Uydurma (FUYSA), Kaskat Bağlantılı (KBYSA) ve Genelleştirilmiş Regresyon Yapay Sinir Ağları (GRYSA) ile Destek Vektör Regresyonu (DVR), Sınıflama ve Regresyon Ağacı (SRA) Çoklu Regresyon Analizi (ÇRA)) geliştirilmiştir. Çalışmanın temel amacı, önerilen modeller ile atomik koordinatların hesaplanma süresini günlerden dakikalara düşürmektir. Elementlerin atomik koordinatları ve kiral vektörleri ile CASTEP yazılımı kullanılarak bir veri kümesi oluşturulmuştur. Ortalama Karesel Hata (OKH), Ortalama Mutlak Hata (OMH), Standart Tahmin Hatası ve Korelasyon Katsayısı ölçütleri ile önerilen modellerin doğrulukları değerlendirilmiştir. Veri kümesi, 10-katlı çapraz doğrulama kullanılarak ve kullanılmadan ayrı ayrı çalışılmıştır. İBYSA, KBYSA ve FUYSA tahmin modelleri OKH ve OMH ölçütleri için çok yüksek başarım göstermişlerdir. Bu modelleri sırasıyla SRA, DVR, GRYSA ve ÇRA izlemektedir. Bu çalışmadan elde edilen sonuçlar iki şekilde kullanılabilir: i) Atomik koordinatlar hiçbir benzetim programı kullanmadan düşük bir hata oranı ile tahmin edilebilir, ii) Tahmin edilen koordinatlar benzetim programlarına başlangıç değeri olarak verilip benzetimin süresi ciddi oranda kısaltılabilir.

**Anahtar Kelimeler:** Atomik Koordinat, Karbon Nanotüp, Tahmin, Makine Öğrenmesi

## GENİŞLETİLMİŞ ÖZET

Günümüzde yarıiletken teknolojisinin IVA grubu elementer yarıiletken olan karbon ve karbon türevli yapılara yönelmesi sentezlenecek yapıların benzetimlerini önemli kılmıştır. Bu benzetimler için ilkten gelim (ab initio) yöntemleri her geçen gün daha da ilerletilerek uygulanmaktadır. Bu yöntemlerin karmaşık hesaplamaları karşısında işlemleri kolaylaştıracak alternatif yaklaşımlara ihtiyaç vardır.

Bu tezde, karbon nanotüplerin (KNT) atomik koordinatlarını tahmin etmek için yapay zeka tabanlı yedi adet tahmin modeli (İleri Beslemeli Yapay Sinir Ağı (İBYSA), Fonksiyon Uydurma Yapay Sinir Ağı (FUYSA), Kaskat Bağlantılı Yapay Sinir Ağı (KBYSA), Genelleştirilmiş Regresyon Yapay Sinir Ağı (GRYSA), Destek Vektör Regresyonu (DVR), Sınıflama ve Regresyon Ağacı (SRA) ve Çoklu Regresyon Analizi (ÇRA)) geliştirilmiştir. Geliştirilen tahmin modelleri ile klasik matematiksel yöntemlerde yaşanan zaman kaybının en düşük seviyeye düşürülmesi ve matematiksel yöntemlerle ulaşılan kesin sonuçlara en yakın tahminlerin yapılması amaçlanmaktadır.

Yoğunluk fonksiyonel teorisi (YFT) atomik koordinatların hesaplanmasında diğer matematiksel yaklaşımlara kıyasla daha hızlı ve kesin sonuçlar üreten başarılı bir metottur. Atomik koordinatların belirlenmesi için YFT kullanılan BIOVIA Materials Studio CASTEP (CASTEP) yazılımı ile yapılan hesaplamalar sonucunda diğer yaklaşımlara göre düşük olan ancak günler süren hesaplama süreleri ortaya çıkabilmektedir ve bu sürelerinin azaltılması için yüksek kapasiteli iş istasyonları ve paralel iş istasyonları şebekeleri (grid) gibi pahalı çözümlerin kullanılması gerekmektedir.

Uzun hesaplama süresi birçok araştırmacının çalışmaları sırasında karşılaşılan bir sorun olmuş ve bu probleme araştırmacılar tarafından dikkat çekilmiştir. Literatürde sıklıkla "YFT hesaplamaları zaman tüketicidir, YFT hesaplamaları zaman almaktadır" nitelemeleri yapılmıştır. Hesaplamaların daha hızlı

yapılması için araştırmacılar tarafından çeşitli yaklaşımlar öne sürülmüş ancak önerilen yaklaşımlar matematiksel yöntemleri kullanmaya devam ettiği için etkileri sınırlı olmuştur. Bu tezde yaptığımız çalışmalarda, matematiksel hesaplamaların zaman alan işlemlerine karşın, matematiksel hesaplamaları kullanmayan ve tahmin yoluyla KNTlerin atomik koordinatlarının çok kısa sürede ve en doğru şekilde belirlenmesini amaçlayan yapay zeka tabanlı tahmin modelleri geliştirilmiştir. Yapılan literatür taramasında atomik koordinatların tahmin yoluyla belirlenmesine yönelik herhangi bir çalışmaya rastlanmamıştır. Bu tezde KNTlerin atomik koordinatlarının tahmini için yapay zeka kullanımı literatür açısından yeni bir uygulama alanı olmaktadır.

Geliştirilen tahmin modelleri ile iki temel amaç hedeflenmiştir. İlk olarak regresyon tabanlı bu modeller ile KNTlerin atomik koordinatları çok kısa sürede (dakikalar içerisinde) tahmin edilecektir. Bu sayede araştırmacıların benzetim yazılımı kullanmadan çok kısa sürede çok yakın sonuçlara ulaşması sağlanmaya çalışılmıştır. İkinci amaç ise elde edilen bu atomik koordinatların, benzetim yazılımı olarak kullanılan CASTEP'de hesaplamalar sırasında oluşan yineleme (iterasyon) sayısının azaltılmasını sağlamak için başlangıç atomik koordinatları olarak kullanılmasını sağlamaktır. Bu durumda araştırmacılar, önerilen tahmin modellerini bir benzetim programı ile birlikte kullanarak günlerce sürebilecek hesaplamalar yerine saatler içerisinde kesin sonuçlara ulaşabileceklerdir.

Çalışmada, geliştirilen tahmin modellerinde kullanılmak üzere CASTEP yazılımı ile 10721 veri örneği içeren girdi ve çıktı veri kümeleri hazırlanmıştır. Girdi veri kümesi, başlangıç atomik koordinatları (u, v, w) ve kiral vektörleri (m, n) içeren 5 parametreden oluşmaktadır. Çıktı veri kümesi ise hesaplanmış atomik koordinatlardan (u', v', w') oluşan 3 parametreyi içermektedir. CASTEP yazılımı KNT benzetimi, geometri optimizasyon hesaplaması ve ilgili kiral vektöre göre başlangıç ve hesaplanmış atomik koordinatların bulunduğu çıktı dosyasının oluşturulması amacıyla kullanılmıştır.

Önerilen atomik koordinat tahmin modelleri MATLAB yazılımında tasarlanmış ve modellerin doğruluklarını değerlendirebilmek için Ortalama Karesel Hata (OKH), Ortalama Mutlak Hata (OMH), Standart Tahmin Hatası (STH) ve Korelasyon Katsayısı (R) ölçütleri kullanılmıştır.

Veri kümesi, 10-katlı çapraz doğrulama kullanılarak ve kullanılmadan ayrı ayrı çalışılmıştır. 10-katlı çapraz doğrulamada verilerin %90'ı eğitim verisi olarak, %10'u ise test verisi olarak kullanılmıştır. Çapraz doğrulama kullanılmadığı durumlarda ise verilerin %70'i eğitim, %15'i doğrulama, kalan %15'i ise test verisi olarak ayrılmıştır.

Yapılan testlerin sonuçlarına göre tüm tahmin modellerinde başarılı sonuçlara ulaşılmıştır. İBYSA, FUYSA ve KBYSA tahmin modelleri ile diğer modellere kıyasla daha başarılı sonuçlar elde edilmiştir. Aynı zamanda bu sonuçlar ile yapay sinir ağları tabanlı modellerin, KNTlerin atomik koordinatlarını çok yakın tahmin ettiği ortaya çıkmıştır. SRA ve DVR tahmin modelleri ile ortalama performans sonuçlarına ulaşılmıştır.

İBYSA, FUYSA ve KBYSA tahmin modelleri, takviyeli öğrenme yöntemi ile eğitilmeleri nedeniyle uzun yineleme süreçleri boyunca çalışmışlardır. Bu nedenle, çalışma zamanı açısından diğer modellerden daha fazla zaman harcamışlardır. GRYSA ve MLR tahmin modelleri ise tek geçişli öğrenme yeteneği ile diğer modellerden çalışma zamanlarına göre kıyaslandığında daha hızlı olmalarına rağmen, İBYSA, FUYSA ve KBYSA tahmin modellerinin yanı sıra DVR ve SRA modellerinden de daha kötü sonuçlar üretmişlerdir.

İBYSA ve KBYSA tahmin modelleri OKH ve OMH ölçütleri için çok yüksek başarım göstermişlerdir. FUYSA tahmin modeli ise gösterdiği hesaplama performansı ile bu iki modele çok yakın sonuçlar elde etmiştir. GRYSA ve ÇRA tahmin modelleri hariç, diğer modellere ait STH ve R sonuçları, koordinatlardan bağımsız olarak birbirine çok yakındır. İBYSA, KBYSA ve FUYSA tahmin modelleri haricindeki modellerin başarısı ise sırasıyla SRA, DVR, GRYSA ve

ÇRA'dır. Sonuçlar koordinatlar açısından ele alındığında ise çapraz doğrulama kullanılmadan elde edilen sonuçlara göre KBYSA tahmin modeli u' ve w' koordinatları için iyi bir tahmin modeli olmuştur. İBYSA tahmin modeli ise v' koordinatının tahmininde en başarılı tahmin modeli olmuştur. En düşük OKH ve OMH değerleri ise çapraz doğrulama kullanılmadan w' koordinatının tahmininde elde edilmiştir.

Çalışmanın bir diğer amacı olan, tahmin edilen atomik koordinatların, benzetim yazılımı olarak kullanılan CASTEP programında, yineleme sayısının azaltılmasını sağlamak için başlangıç atomik koordinatları olarak kullanılması işleminden sonra yineleme sayılarında yaklaşık %85'e varan azalmalar gözlenmiştir. Tahmin modellerinin, CASTEP yazılımındaki yineleme sayılarını azaltma oranlarındaki başarı sırası önerilen modellerin tahmin başarılarına paralel bir şekilde gerçekleşmiştir. FUYSA tahmin modeli en yüksek yineleme sayısı azalışına ulaşırken, ÇRA tahmin modelindeki iyileşme en düşük seviyede gerçekleşmiştir. Yineleme sayılarındaki bu azalmalar, CASTEP yazılımında yapılan benzetimlerin günler ile ifade edilen hesaplama sürelerini saatlerle ölçülebilecek hale getirmektedir.

Bu çalışmadan elde edilen sonuçların iki şekilde kullanılmasının mümkün olduğu görülmüştür: i) Atomik koordinatlar hiçbir benzetim programına ihtiyaç duyulmadan, yapay zeka yöntemleri kullanılarak düşük bir hata oranı ile tahmin edilebilir, ii) Önerilen modeller tarafından tahmin edilen koordinatlar, benzetim programlarına başlangıç değeri olarak verilip, benzetimin süresi ciddi oranda kısaltılabilir.

Bu çalışmanın sonucunda yazılımlara entegre edilebilecek, atomik koordinatları tahmin eden yapay zeka çözümlerinin etkinliği gözlenmiş ve nanoteknoloji alanında benzetim süreçlerini önemli miktarda kısaltan sonuçlar, bilimsel veya ticari yazılımlara entegre edilebilecek halde araştırmacıların kullanımına sunulmuştur.

**ACKNOWLEDGEMENTS**

I would like to thank and express my gratitude to my advisor *Assoc. Prof. Dr. Mutlu AVCI* for his great guidance, endeavor and patience. This thesis was finally over with his help and motivation. I wish he will not deprive his support for the rest of my academic life.

I would like to acknowledge my committee members *Prof. Dr. Tülay YILDIRIM, Prof. Dr. Mustafa GÖK, Prof. Dr. Faruk KARADAĞ* and *Assoc. Prof. Dr. Zeki YETGİN* for their time, suggestions and patience.

I would also like to thank *Prof. Dr. Faruk KARADAĞ* for his guidance about physics aspect of the thesis.

I would like to thank *Scientific Research Project Unit of Çukurova University* for the financial support during my PhD thesis.

Special thanks to my mother *Kamet*, father *Turan*, sister *Havva* and brother *Can* for their love and support for all my life.

Finally, I should say that this thesis would never end without my beloved wife *Çiğdem* and her support, help and patience. I would like to thank and express my love to *Çiğdem* for the last eight years with her and the rest of my life with our twins *Defne* and *Nehir*.

**CONTENTS** **PAGE**

ABSTRACT ..... I
ÖZ ..... II
GENİŞLETİLMİŞ ÖZET ..... III
ACKNOWLEDGEMENTS ..... VII
CONTENTS ..... VIII
LIST OF TABLES ..... X
LIST OF FIGURES ..... XIV
LIST OF ABBREVIATIONS ..... XVI
1. INTRODUCTION ..... 1
1.1. Review of Carbon Nanotubes ..... 4
1.2. Ab Initio Method and Density Functional Theory ..... 10
1.3. Machine Learning ..... 14
1.3.1. Unsupervised and Supervised Learning ..... 14
1.3.2. Regression-Based Supervised Methods ..... 15
2. RELATED WORK ..... 19
2.1. Previous Studies on Density Functional Theory and Carbon Nanotube Geometry Optimizations ..... 19
2.2. Previous Studies on Artificial Intelligence in Carbon Nanotube Calculations ..... 23
3. MATERIALS AND METHODS ..... 31
3.1. Electronic Structure Calculations of Carbon Nanotubes using CASTEP ..... 31
3.2. Artificial Neural Networks ..... 33
3.2.1. Feed-Forward Neural Network and Function Fitting Neural Network ..... 35
3.2.2. Cascade Forward Neural Network ..... 36
3.2.3. Generalized Regression Neural Network ..... 37

3.3. Support Vector Regression ..... 39
3.4. Classification and Regression Tree ..... 41
3.5. Multiple Linear Regression ..... 44
3.6. Cross-Validation Method for Assessing Performance ..... 46
4. RESULTS AND DISCUSSION ..... 49
4.1. Dataset ..... 49
4.2. Performance Measures ..... 55
4.3. Performance Results of Artificial Neural Network Prediction Models ..... 56
4.4. Performance Results of Support Vector Regression Prediction Model ..... 70
4.5. Performance Results of Classification and Regression Tree Prediction Model ..... 74
4.6. Performance Results of Multiple Linear Regression Prediction Model ..... 77
4.7. Overall Comparison of the Prediction Models ..... 79
4.8. Improvement on CASTEP Simulation Time Consumption ..... 84
5. CONCLUSIONS ..... 89
REFERENCES ..... 91
BIOGRAPHY ..... 101

**LIST OF TABLES** **PAGE**


Table 1.1. Comparison between SWNT and MWNT ..... 8
Table 1.2. Physical properties of carbon nanotubes ..... 9
Table 4.1. The calculation time for CNT simulations in seconds ..... 54
Table 4.2. Descriptive statistics for the dataset ..... 55
Table 4.3. Performance results of u' coordinate prediction using ANN prediction models with the non-cross-validated dataset ..... 611
Table 4.4. Performance results of v' coordinate prediction using ANN prediction models with the non-cross-validated dataset ..... 61
Table 4.5. Performance results of w' coordinate prediction using ANN prediction models with the non-cross-validated dataset ..... 61
Table 4.6. Performance results of u' coordinate prediction using FFNN prediction model by means of 10-fold cross-validation ..... 65
Table 4.7. Performance results of v' coordinate prediction using FFNN prediction model by means of 10-fold cross-validation ..... 65
Table 4.8. Performance results of w' coordinate prediction using FFNN prediction model by means of 10-fold cross-validation ..... 66
Table 4.9. Performance results of u' coordinate prediction using FITNET prediction model by means of 10-fold cross-validation ..... 66
Table 4.10. Performance results of v' coordinate prediction using FITNET prediction model by means of 10-fold cross-validation ..... 67
Table 4.11. Performance results of w' coordinate prediction using FITNET prediction model by means of 10-fold cross-validation ..... 67
Table 4.12. Performance results of u' coordinate prediction using CFNN prediction model by means of 10-fold cross-validation ..... 68
Table 4.13. Performance results of v' coordinate prediction using CFNN prediction model by means of 10-fold cross-validation ..... 68

Table 4.14. Performance results of w’ coordinate prediction using CFNN prediction model by means of 10-fold cross-validation ..... 69
Table 4.15. Performance results of u’ coordinate prediction using GRNN prediction model by means of 10-fold cross-validation ..... 69
Table 4.16. Performance results of v’ coordinate prediction using GRNN prediction model by means of 10-fold cross-validation ..... 70
Table 4.17. Performance results of w’ coordinate prediction using GRNN prediction model by means of 10-fold cross-validation ..... 70
Table 4.18. Performance results of u’, v’ and w’ coordinates prediction using SVR prediction model with the non-cross-validated dataset..... 71
Table 4.19. The performance results of u’ coordinate prediction with 10-fold cross validation using SVR prediction model ..... 72
Table 4.20. The performance results of v’ coordinate prediction with 10-fold cross validation using SVR prediction model ..... 73
Table 4.21. The performance results of w’ coordinate prediction with 10-fold cross validation using SVR prediction model ..... 73
Table 4.22. Performance results of u’, v’ and w’ coordinates prediction using CART prediction model with the non-cross-validated dataset..... 75
Table 4.23. The performance results of u’ coordinate prediction with 10-fold cross validation using CART prediction model ..... 76
Table 4.24. The performance results of v’ coordinate prediction with 10-fold cross validation using CART prediction model ..... 76
Table 4.25. The performance results of w’ coordinate prediction with 10-fold cross validation using CART prediction model ..... 77
Table 4.26. Performance results of u’, v’ and w’ coordinates prediction using MLR prediction model with the non-cross-validated dataset..... 78
Table 4.27. The performance results of u’ coordinate prediction with 10-fold cross validation using MLR prediction model ..... 78

Table 4.28. The performance results of v' coordinate prediction with 10-fold cross validation using MLR prediction model ..... 79
Table 4.29. The performance results of w' coordinate prediction with 10-fold cross validation using MLR prediction model ..... 79
Table 4.30. Overall comparison of the u' coordinate prediction models with non-cross validated dataset ..... 81
Table 4.31. Overall comparison of the v' coordinate prediction models with non-cross validated dataset ..... 82
Table 4.32. Overall comparison of the w' coordinate prediction models with non-cross validated dataset ..... 82
Table 4.33. Overall comparison of the best values of u' coordinate prediction models with 10-fold cross-validated dataset ..... 82
Table 4.34. Overall comparison of the best values of v' coordinate prediction models with 10-fold cross-validated dataset ..... 83
Table 4.35. Overall comparison of the best values of w' coordinate prediction models with 10-fold cross-validated dataset ..... 83
Table 4.36. Overall comparison of the average values of u' coordinate prediction models with 10-fold cross-validated dataset ..... 83
Table 4.37. Overall comparison of the average values of v' coordinate prediction models with 10-fold cross-validated dataset ..... 84
Table 4.38. Overall comparison of the average values of w' coordinate prediction models with 10-fold cross-validated dataset ..... 84
Table 4.39. Number of iterations before and after the two-staged CASTEP simulation ..... 86
Table 4.40. Percentage of declines in number of iterations ..... 87

**LIST OF FIGURES** **PAGE**

Figure 1.1. Prediction model addition to the CASTEP simulation ..... 3
Figure 1.2. Schematic structure of SWCNT (A) and MWCNT (B) ..... 5
Figure 1.3. Models of three atomically perfect SWCNT structures..... 6
Figure 1.4. The chiral vector C ..... 7
Figure 3.1. Schematic diagram of FFNN ..... 35
Figure 3.2. Transfer functions (a) LOGSIG (b) TANSIG (c) PURELIN ..... 36
Figure 3.3. Schematic diagram of CFNN ..... 37
Figure 3.4. Schematic diagram of GRNN ..... 38
Figure 3.5. $\epsilon$ -insensitive loss function for a linear SVM ..... 40
Figure 4.1. Building single-wall nanotube in CASTEP ..... 50
Figure 4.2. A screenshot of simulated CNT in CASTEP..... 50
Figure 4.3. Calculation interface of CASTEP..... 51
Figure 4.4. The chart of energy change during the iterations ..... 52
Figure 4.5. The chart of convergence change during the iterations ..... 52
Figure 4.6. Data files created by CASTEP after calculation..... 53
Figure 4.7. Calculation time by CASTEP ..... 53
Figure 4.8. FFNN Model with 5, 20 and 3 neurons ..... 57
Figure 4.9. FITNET Model with 5, 10 and 3 neurons..... 57
Figure 4.10. CFNN Model with 5, 10 and 3 neurons..... 58
Figure 4.11. R values of u' coordinate prediction with the non-cross-validated dataset (a) FFNN (b) FITNET (c) CFNN (d) GRNN ..... 62
Figure 4.12. R values of v' coordinate prediction with the non-cross-validated dataset (a) FFNN (b) FITNET (c) CFNN (d) GRNN ..... 63
Figure 4.13. R values of w' coordinate prediction with the non-cross-validated dataset (a) FFNN (b) FITNET (c) CFNN (d) GRNN ..... 64

Figure 4.14. R values of u', v' and w' coordinates prediction with the non-cross-validated dataset using SVR prediction model (a) u' coordinate (b) v' coordinate (c) w' coordinate ..... 72
Figure 4.15. A sample CART prediction model for atomic coordinate prediction ..... 75
Figure 4.16. R values of u', v' and w' coordinates prediction with the non-cross-validated dataset using CART prediction model (a) u' coordinate (b) v' coordinate (c) w' coordinate ..... 75
Figure 4.17. Steps of the two-staged CASTEP simulation ..... 85

## LIST OF ABBREVIATIONS

| | | |
|---|---|---|
| 2D | : | Two-dimensional |
| a | : | Ampere |
| Å | : | Angstrom |
| Ag | : | Silver |
| ANN | : | Artificial neural network |
| C | : | Carbon |
| CART | : | Classification and regression tree |
| CASTEP | : | BIOVIA Materials Studio CASTEP |
| CFNN | : | Cascade forward neural network |
| $CH_3$ | : | Methyl |
| cm | : | Centimeter |
| $cm^3$ | : | Cubiccentimeter |
| CNT | : | Carbon nanotube |
| Co | : | Cobalt |
| Cu | : | Copper |
| DFT | : | Density functional theory |
| eV | : | Electron volt |
| Fe | : | Iron |
| FFNN | : | Feed forward neural network |
| FITNET | : | Function fitting neural network |
| g | : | Gram |
| GA | : | Genetic algorithm |
| Ghz | : | Gigahertz |
| GPa | : | Gigapascal |
| GRNN | : | Generalized regression neural network |
| H | : | Hydrogen |

| | | |
|---|---|---|
| $H_2$ | : | Dihydrogen |
| K | : | Kelvin |
| Li | : | Lithium |
| LOGSIG | : | Log-sigmoid transfer function |
| m | : | Meter |
| $m^2$ | : | Squaremeter |
| MAE | : | Mean absolute error |
| MLR | : | Multiple linear regression |
| MSE | : | Mean squared normalized error |
| MWCNT | : | Multi-wall carbon nanotube |
| N | : | Nitrogen |
| Ni | : | Nickel |
| nm | : | Nanometer |
| O-G | : | Graphene with epoxy |
| P-G | : | Pristine grapheme |
| PURELIN | : | Pure linear transfer function |
| R | : | Multiple correlation coefficient |
| RAM | : | Random access memory |
| s | : | Second |
| SEE | : | Standard error of estimation |
| SiCNT | : | Silicon carbide nanotube |
| SVM | : | Support vector machine |
| SVR | : | Support vector regression |
| SWCNT | : | Single-wall carbon nanotube |
| TANSIG | : | Hyperbolic tangent transfer function |
| $TiO_2$ | : | Titanium dioxide |
| TPa | : | Terapascal |
| V | : | Volt |

| | | |
|---|---|---|
| V-G | : | Graphene with monovacancy |
| W | : | Watt |
| μm | : | Micrometer |

# 1. INTRODUCTION

Nowadays, semiconductor technology is directed to carbon and carbon-derived structures which are IVA group of semiconducting elements and this situation makes simulations of structures to be synthesized important. For these simulations, the methods of preterm delivery (ab initio) are being implemented with each passing day. There is a need for alternative approaches that will facilitate operations in the face of complex computations of these methods.

In this thesis some alternative Machine Learning based methods, like artificial neural network (ANN), support vector regression (SVR), classification and regression tree (CART) and multiple linear regression (MLR), are proposed to predict the atomic coordinates of carbon nanotubes (CNT).

Density functional theory (DFT) is the most successful method that calculates atomic coordinates faster than other mathematical approaches and it also reaches more accurate results. However, the elapsed time for calculation of high number of atoms is quite long as measured in this study using BIOVIA Materials Studio CASTEP (CASTEP). Calculation may even take several days due to calculation capability limits of workstation computers. On the other hand, users need to use more powerful workstations and parallel computer grids which are too expensive to buy easily for reducing the calculation time. In literature, many researchers remark this calculation time problem in their studies. General view of the researchers who studied on geometry optimization using DFT can be summarized as “DFT calculations are time consuming” (Bauschlicher and Ricca, 2004, Geng et al., 2013, Hong-Xia et al., 2010, Hong-Xia et al., 2009, Krasnenko et al., 2014, Mao and Zhong, 2009, Tao et al., 2014, Van Regemorter and Larsson, 2008, Wang et al., 2013). Some researchers have tried to overcome this problem by various approaches given in Section 2.1.

The motivation of this research is to reduce 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. In this work, the problem is investigated in another perspective. Instead of calculation; the atomic coordinates are predicted as accurately as possible in a short time. These predicted atomic coordinates can be used as initial coordinates for the simulation software. Thus, the exact atomic coordinates can be calculated within minutes or hours instead of days utilizing the proposed approach. In some researches predicted atomic coordinates may be enough in accuracy. In that cases the proposed approach with direct result outcomes may be the fastest solution. The prediction mechanism to reach faster results with limited accuracy may be obtained by using machine learning methods.

There are two main objectives aimed in this thesis. First one is to develop prediction models using regression-based supervised machine learning methods such as ANN, CART, MLR and SVR to estimate the atomic coordinates of CNTs. Second one is to obtain initial atomic coordinates for decreasing number of iterations in the simulation software by using prediction models. Two distinct datasets named as input dataset and output dataset are prepared to be used in prediction models. Initial atomic coordinates u, v, w and the pair of integers (n, m) used for specifying the chiral vector are the 5 parameters of input dataset. On the other hand, calculated atomic coordinates u', v', w' are the 3 parameters of output dataset. CASTEP software is used for three processes: the CNTs simulation, the geometry optimization calculation and output file creation for the initial and calculated atomic coordinates of related chiral vector. Mean Squared Normalized Error (MSE), Mean Absolute Error (MAE), Standard Error of Estimation (SEE) and Multiple Correlation Coefficient (R) are calculated to evaluate the performance of prediction models.

According to the test results all prediction models performed estimation tasks successfully. Feed Forward Neural Network (FFNN), Function Fitting Neural Network (FITNET) and Cascade Forward Neural Network (CFNN) prediction

models achieved better results than other tested prediction models. In other words, atomic coordinates of CNTs are closely estimated by ANN based models. However, average performance results are achieved by CART and SVR prediction models.

According to training times consumption, FFNN, FITNET and CFNN prediction models last longer than others. Generalized Regression Neural Network (GRNN) and MLR prediction models are faster than the others with one pass learning ability. At the same time, they cannot predict more accurate results than the existing.

Prediction results of the proposed models are used in CASTEP software for initial atomic coordinate estimation. If CASTEP simulation started with random atomic coordinates, the calculation may takes several days. The number of iterations in CASTEP can be decreased thus the calculation time decreases from days to minutes by using prediction results of proposed prediction models as initial atomic coordinates. With a two-staged simulation, the steps of prediction models are processed between CASTEP steps as in Figure 1.1. There is a consistency between decrease in iteration number and the prediction accuracy. FITNET and MLR based models achieve the best and the worst iteration number decrement, respectively. FITNET based model reached up to approximately 85% of decline.

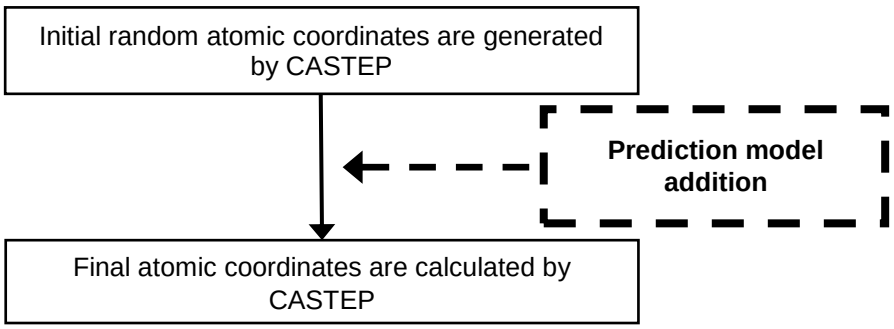



Figure 1.1. Prediction model addition to the CASTEP simulation

FFNN, FITNET and CFNN prediction models with very low MSE and MAE values on the cross validated dataset reached the first aim of the thesis by estimating high accuracy atomic coordinates. Also all prediction models achieve high R values. On the other hand CFNN (for u' and w' coordinates) and FFNN (for v' coordinate) prediction models perform successful results with the non-cross validated dataset. The lowest MSE and MAE values of this study are obtained from w' coordinate prediction when the cross validation is not considered.

The usage of predicted atomic coordinates as initial values to calculate exact results in CASTEP is the second aim of this thesis and it is also reached with an 85% of decrease in the number of iterations. It is also observed that the exact results of atomic coordinates can be achieved in a short time by using Machine Learning methods.

The rest of the thesis is organized as follows: This section covers short reviews of Carbon Nanotubes, Ab Initio Method, Density Functional Theory and Machine Learning algorithms. A literature review about CNT geometry optimization and artificial intelligence based CNT studies is summarized in section 2. The materials used in this study and the proposed methods are explained in section 3. In section 4, the dataset, performance measures, performance results of the prediction models and performance improvement of CASTEP simulation after using predicted data are presented. Finally, the study is concluded in section 5.

### 1.1. Review of Carbon Nanotubes

The chemical element carbon (C) has six electrons which exist on $1s^2$, $2s^2$ and $p^2$ atomic orbital and its atomic number is 6. Carbon can hybridize in either sp, $sp^2$ or $sp^3$ forms. Fullerenes (Kim et al., 2003), graphene (Ouyang et al., 2001) and CNTs (Chico et al., 1996) are very constant $sp^2$ carbon bonded materials at nanometer size and their discoveries motivated to make investigations in this field.

Graphene and CNTs have many common physical properties which are derived from graphene. Carbon atoms of graphene are densely arranged in a regular $sp^2$-bonded atomic-scale honeycomb (hexagonal) pattern. For other $sp^2$ carbon bonded materials (allotropes) like fullerenes and CNTs, this pattern is a basic structure. The theoretical difference of CNT is being a cylinder fabricated of rolled up graphene sheet. It has two types according to wall types. Single-wall carbon nanotubes (SWCNTs) were reported in 1993 (Iijima and Ichihashi, 1993) and have only one wall. (Iijima, 1991) were first discovered multi-wall carbon nanotubes (MWCNTs) in 1991 and these nanotubes have more than one wall (Figure 1.2) (Eatemadi et al., 2014).

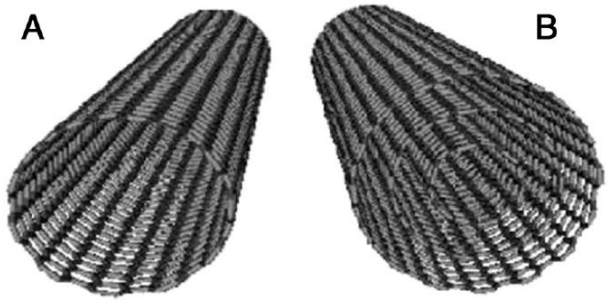



Figure 1.2. Schematic structure of SWCNT (A) and MWCNT (B) (Eatemadi et al., 2014)

Many structures with completely different properties are constructed according to various carbon bonding. A layered construction with weak out-ofplane bonding of the van der Waals form and strong inplane bounds is built by the $sp^2$ hybridization of carbon. MWCNTs is formed by a few to a few tens of concentric cylinders with the regular periodic interlayer spacing locate around ordinary central hollow. A range of interlayer spacing (0.34 to 0.39 nm) is calculated according to the real space analysis of multiwall nanotube images (Ajayan and Ebbesen, 1997).

The inner and outer diameter of MWCNTs varies from 0.4 nm up to a few nanometers and 2 nm up to 30 nm, respectively and depend on the number of layers. The tips of MWCNTs are capped by dome shaped half-fullerene molecules

(pentagonal defects). Axial size changes between 1μm and a few centimeters. The half-fullerene molecules at the two ends help in closing of the tube (Eatemadi et al., 2014).

The diameter of a SWCNT differ from 0.4 to 3 nm and its length is generally of the micrometer range. Usually they come together to form bundles (ropes). They organized hexagonally in a bundle structure and form a crystal-like construction (Chico et al., 1996).

SWCNTs can exists in three forms named as armchair, chiral, and zigzag according to wrapping to a cylinder way (Figure 1.3). Chiral vector is a pair of indices (n, m) that directly effects electrical properties of nanotubes and characterizes the structure of SWCNTs. The integers, n and m, determine the number of unit vectors along two directions in the honeycomb crystal lattice of graphene. According to the general opinions, the nanotubes are called as zigzag nanotubes when m=0. If n=m, then they are called as armchair nanotubes and for all other states they are named as chiral (Eatemadi et al., 2014).

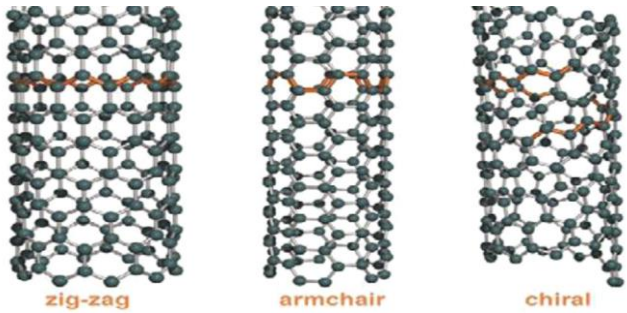



Figure 1.3. Models of three atomically perfect SWCNT structures (Eatemadi et al., 2014)

The tube diameter $d$ [4, 5] and the direction of rolling a graphene sheet are determined by the chiral vector $C = na1 + ma2$, where $a1$ and $a2$ are the base cell vectors of graphite (Figure 1.4). Thus, the carbon tube diameter (equation 1.1) can be calculated by

$$d = \frac{a\sqrt{m^2 + mn + n^2}}{\pi} \quad (1.1.)$$

where $a$ is the lattice constant in the graphite sheet (Eatemadi et al., 2014).

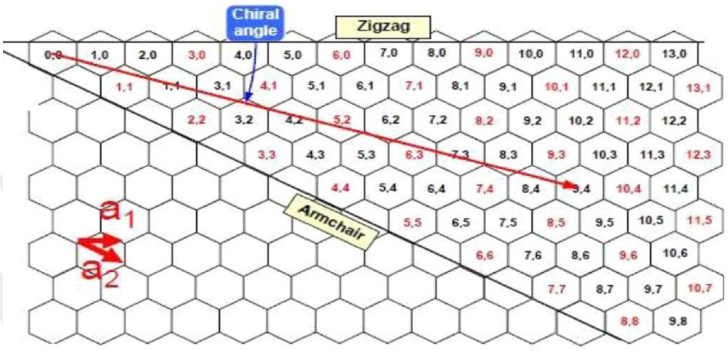



Figure 1.4. The chiral vector C (Eatemadi et al., 2014)

A nanotube is classified as 'metallic' or highly conducting nanotubes if $n - m$ is a multiple of 3, otherwise it is a semimetallic or semiconductor. However, an armchair formed nanotube is always classified as metallic. Nanotube structures are affected from many parameters and vectors such as the following (O'connell, 2006):

Translational vector $= T = t1a1 + t2a2 \gg (t1, t2)$

Chiral vector $= \mathrm{Ch} = na1 + na2 \gg (n, m)$

Length of chiral vector $= L = a\sqrt{(n^2 + m^2 + n * m)}$,

where $a$ is the lattice constant

Chiral angle $= cos\theta = (2n + m)/(2 * \sqrt{(n^2 + m^2 + n * m)})$

Number of hexagons in the unit cell $= N = (2 * (n^2 + m^2 + n * m)/dR)$

Diameter $= dt = L/\pi$

Rotation angle of the symmetry vector $= \psi = 2\pi/N$ (in radians)

Symmetry vector $= R = pa1 + qa2 \gg (p, q)$

Pitch of the symmetry vector $= \tau = ((m * p - n * q) * T)/N$

Russian Doll and Parchment models are the two structural models that MWCNTs can be formed in. For Russian Doll model, there is a CNT that has a greater diameter and contains another nanotube inside it which has a smaller diameter. However, in Parchment model, like a rolled up scroll of paper, a single graphene sheet is wrapped around itself manifold times. Single and multi-walled CNTs have different properties as the common features. Unlike SWCNTs, the outer walls of MWCNTs shield the inner CNTs from chemical interactions with outside substances and present high tensile strength properties (Vander Wal et al., 2003) (Table 1.1).

Table 1.1. Comparison between SWNT and MWNT (Iijima and Ichihashi, 1993)

| **SWNT** | **MWNT** |
|---|---|
| Single layer of graphene | Multiple layers of graphene |
| Catalyst is required for synthesis | Can be produced without catalyst |
| Bulk synthesis is difficult as it requires proper control over growth and atmospheric condition | Bulk synthesis is easy |
| Purity is poor | Purity is high |
| A chance of defect is more during functionalization | A chance of defect is less however once occurred it is difficult to improve |
| Less accumulation in the body | More accumulation in the body |
| Characterization and evaluation is easy | It has very complex structure |
| It can be easily twisted and is more pliable | It cannot be easily twisted |

Compared to steel and Kevlar, CNTs have a higher tensile because of the $sp^2$ bonds that they have between the individual carbon atoms. The $sp^3$ bond found in diamond is even weaker than the $sp^2$ bond. Theoretically, the tensile strength of SWCNTs is hundreds of times stronger than steel. In Table 2.1, the main physical properties of CNTs are listed.

Table 1.2. Physical properties of carbon nanotubes (Eatemadi et al., 2014)

| **Physical properties** | | **Values** |
|---|---|---|
| **Equilibrium structure** | Average diameter of SWNTs | 1.2 to 1.4 nm |
| | Distance from opposite carbon atoms (line 1) | 2.83 Å |
| | Analogous carbon separation (line 2) | 2.456 Å |
| | Parallel carbon bond separation (line 3) | 2.45 Å |
| | Carbon bond length (line 4) | 1.42 Å |
| | C-C tight bonding overlap energy | Approximately 2.5 eV |
| | Group symmetry (10,10) | CSV |
| | Lattice: bundles of ropes of nanotubes | Triangular lattice (2D) |
| **Lattice constant** | | 17 Å |
| **Lattice parameter** | (10,10) Armchair | 16.78 Å |
| | (17,0) Zigzag | 16.52 Å |
| | (12,6) Chiral | 16.52 Å |
| **Density** | (10,10) Armchair | 1.33 g/cm$^3$ |
| | (17,0) Zigzag | 1.34 g/cm$^3$ |
| | (12,6) Chiral | 1.40 g/cm$^3$ |
| **Interlayer spacing** | (n,n) Armchair | 3.38 Å |
| | (n,0) Zigzag | 3.41 Å |
| | (2n,n) Chiral | 3.39 Å |
| **Optical properties** | | |
| **Fundamental gap** | For (n,m); n-m is divisible by 3 [metallic] | 0 eV |
| | For (n,m); n-m is not divisible by 3 [semiconducting] | Approximately 0.5 eV |
| **Electrical transport** | | |
| | Conductance quantization | (12.9 k O)-1 |
| | Resistivity | 10-4 O -cm |
| | Maximum current density | 1,013 a/m$^2$ |
| **Thermal transport** | | |
| | Thermal conductivity | Approximately 2,000 W/m/K |
| | Phonon mean free path | Approximately 100 nm |
| | Relaxation time | Approximately 10 to 11 s |
| **Elastic behavior** | | |
| | Young's modulus (SWNT) | Approximately 1 TPa |
| | Young's modulus (MWNT) | 1.28 TPa |
| | Maximum tensile strength | Approximately 100 GPa |

### 1.2. Ab Initio Method and Density Functional Theory

The atomistic viewpoint becomes a very critical issue for both scientists and engineers after the materials behavior at nanoscale are dominated by the effects of single atoms. Nevertheless, to achieve the straightforward manipulation of the nanostructure is rather complex and for this reason it is difficult to measure the experimental results of mechanical properties at nanoscale. At present, proper measuring techniques at the nanoscale are not enough and to get some dependable data from indirect measurements is really difficult. So, to investigate the mechanical properties of nanostructures, the role of the computational techniques become very important. From this point of view, the importance of computer modeling becomes more important every day for new technologies development. The models are grouped under two main categories named as dynamic and static. The interaction between atoms is considered dynamically and statically with dynamic and static models, respectively. Ab initio methods, tight-binding methods, and classical molecular dynamics simulations are three types of the dynamic methods (Tserpes and Silvestre, 2014).

Ab initio method needs a wave function description for the electrons because it treats separately the electronic and ionic degrees of freedom. The ab initio is an expensive and time-consuming method and for this reason some approximations like the DFT are incorporated instead of many-body interacting wave functions. DFT is based on the fact that the ground-state electronic energy is a unique function of the electronic density (Tserpes and Silvestre, 2014).

(Hohenberg and Kohn, 1964) and (Kohn and Sham, 1965) developed a simple method, DFT, that describes the effects of exchange and correlation in an electron gas. The total energy, that contains exchange and correlation, of an electron gas is a unique functional of the electron density and this fact proved by (Hohenberg and Kohn, 1964). The ground-state energy of the system is the minimum value of the total-energy functional. The exact single-particle ground-state density is this

minimum value which is obtained from the density. It is showed that an exactly equivalent set of self-consistent one-electron equations is used instead of the many-electron problem by (Kohn and Sham, 1965).

The fundamentals of DFT can be summarized as follows. The simplest class of systems, $N$ nonrelativistic, interacting electrons in a nonmagnetic state with Hamiltonian (equation 1.2)

$$H \equiv T + V + U \tag{1.2.}$$

where (in atomic units)

$$T \equiv -\frac{1}{2}\sum_j \nabla_j^2 \ , \quad V \equiv \sum_j v(r_j) \ , \quad U = \frac{1}{2}\sum_{i \neq j} \frac{1}{|r_i - r_j|} \tag{1.3.}$$

For mathematical reasons, a broad class of Hamiltonians is considered with electrons moving in an arbitrary external potential $v(r)$, not only the physically relevant Coulomb potentials due to point nuclei (Tserpes and Silvestre, 2014).

DFT is started with Hohenberg and Kohn where is the rigorous and simple lemma. The external potential $v(r)$ is uniquely determined with the ground state density, $n(r)$,

$$n(r) \rightarrow v(r) \ \text{unique} \tag{1.4.}$$

Since $N$ is determined by $n(r)$ using integration, the full Hamiltonian $H$ is determined by $N$ and thence $H$ determines all properties.

A minimal principle for the energy as functional of $n(r)$ can be derived using this lemma. The following energy functional of $n(r)$ is defined in equation 1.5 for given $v(r)$ (Kohn et al., 1996).

$$E_{v(r)}[n(r)] \equiv \int v(r)n(r)\mathrm{d}r + F[n(r)] \tag{1.5.}$$

where

$$F[n(r)] \equiv (\Psi[n(r)], (T+U)\Psi[n(r)]) \tag{1.6.}$$

is a functional of $n(r)$, since Ψ is. The minimal principle (equation 1.7) is

$$E_{v(r)}[n(r)] \geq E_{v(r)}[n_0(r)] \equiv E \tag{1.7.}$$

where $n_0(r)$ is the density and $E$ is energy of the ground state when $n(r) = n_0(r)$.

The Thomas-Fermi approximation (Myers and Swiatecki, 1996) and its refinements can be rederived after making simple approximations for $F[n(r)]$. Even so, being more useful, a different approach has proved for most purposes. Largest and elementary contributions of $F[n]$ are extracted from itself, by writing equation 1.8 (Kohn et al., 1996)

$$F[n(r)] = T_s[n(r)] + \frac{1}{2}\int \frac{n(r)n(r')}{|r-r'|}\mathrm{d}r\mathrm{d}r' + E_{xc}[n(r)] \tag{1.8.}$$

where $T_s[n(r)]$ is the kinetic energy of a noninteracting system with density $n(r)$, and the next term is the classical expression for the interaction energy. The equation also defines the remaining exchange correlation energy $E_{xc}$ (Kohn et al., 1996).

The physical content of the theory becomes identical to Hartree approximation (Payne et al., 1992) after ignoring $E_{xc}$. Then the Euler-Lagrange equation associated with the stationarity of $E_v[n]$ can be transformed into a new set of self-consistent equations called as Kohn-Sham equations,

$$\left(-\frac{1}{2}\nabla^2 + v(r) + \int \frac{n(r')}{|r-r'|}\mathrm{d}r' + v_{xc}(r) - \epsilon_j\right)\varphi_j(r) = 0 \qquad (1.9.)$$

$$n(r) = \sum_{j=1}^{N} \left|\emptyset_j(r)\right|^2 \qquad (1.10.)$$

$$v_{xc}(r) = \delta E_{xc}[n(r)]/\delta n(r) \qquad (1.11.)$$

which differ from the Hartree equations only by the inclusion of the exchange correlation potential $v_{xc}(r)$. $\varphi_j$ and $\epsilon_j$ are the individual eigen functions and eigenvalues of the Kohn-Sham equations. Like the Hartree equations, these equations must be solved self-consistently. Also with an appropriate approximation for $E_{xc}[n(r)]$, $v_{xc}$ is calculated in each cycle for self-consistency. The extra computational work is very minor compared to a Hartree calculation. Approximations of $E_{xc}$ is the only error in the theory (Kohn et al., 1996). The ground state energy is given in equation 1.12 by

$$E = \sum_{1}^{v} \epsilon_j - \frac{1}{2}\int \frac{n(r)n(r')}{|r-r'|}\mathrm{d}r\mathrm{d}r' - \int v_{xc}(r)n(r)\mathrm{d}r + E_{xc}[n(r)] \qquad (1.12.)$$

where the $\epsilon_j$ and $n$ are the self-consistent quantities (Kohn et al., 1996).

For calculating the properties of materials from first principles, BIOVIA Materials Studio CASTEP (CASTEP) can be defined as a leading code. It can simulate a wide range of properties of materials proprieties using DFT. Some of these proprieties can be listed as structure at the atomic level, energetics, electronic response and vibrational properties (Clark et al., 2005).

### 1.3. Machine Learning

The base of machine learning is to study computer algorithms for regression or classification. Machine learning might learn to make exact predictions, to finish a task, or to act intelligently. The learning can be done with some kind of data or observations like direct experience, samples, or instruction. As a result, machine learning is about doing better in the future using the past experiences (Nilsson, 1996).

Automatic methods are the important point of machine learning. So, the main aim is designing learning algorithms that learn automatically and do not need human intervention or help. In other words, it can be said that the machine learning is "programming by example". The expected future performance of a machine improves depending on changes of its program, structure or data (input or output data) (Nilsson, 1996).

The performed artificial intelligence based tasks changes in system is usually referred by machine learning. These tasks include diagnosis, robot control, planning, prediction, recognition, etc. The "changes" might be improvement of the current systems or application of ab initio to new systems. Many fields intersects with machine learning like statistics, mathematics, theoretical computer science, physics and more, however sure it is a subfield of artificial intelligence (Nilsson, 1996).

Generally machine learning is grouped under two types of learnings: unsupervised and supervised.

#### 1.3.1. Unsupervised and Supervised Learning

The descriptive or unsupervised learning approach is the first main type of machine learning. The inputs, $D = \{x_i\}N\ i = 1$, are given to find out "interesting patterns" in the input data. Also this process is named as knowledge discovery. The kinds of patterns looking for are not known and error metric to use is not obvious for the problem and these properties make this problem much less well-defined.

Differently from supervised learning, the prediction of $y$ for a given $x$ is compared to the observed value (Murphy, 2012).

The aim of predictive or supervised learning approach is to learn a mapping between inputs $x$ and outputs $y$. A set of input-output pairs, $D = \{(x_i, y_i)\}N\ i = 1,$ is used to implement this mapping where $D$ and $N$ is the training set, the number of training examples, respectively (Murphy, 2012).

$x_i$ is the training input and it is a D-dimensional vector of numbers that are named as attributes, features or covariates. Besides, $x_i$ could be a complex structured object, like a sentence, an image, a time series, an email message, a graph or a molecular shape (Murphy, 2012).

Although the output or response variable's form can in principle be anything, however generally $y_i$ is a categorical or nominal variable from some finite set and it means that the problem is a classification or pattern recognition problem. Also $y_i$ can be a real-valued scalar and this makes the problem regression (Murphy, 2012).

### 1.3.2. Regression-Based Supervised Methods

The relationship between inputs or independent variables and the outputs are modelled with regression-based supervised methods. It looks like the form of parametric equations that the parameters are estimated from the data. These methods provide estimates of measures of association between inputs and outcome, adjusted for other inputs, with standard error estimates provided from the modeling paradigm used (Dasgupta et al., 2011)

Linear, logistic, and Poisson regressions are the most known class of regression methods. A group of $k$ predictors $X_1, \dots, X_k$ are used to predict an outcome y. The main structure of these models is to predict y using an appropriately transformed linear function of the inputs $X_1, \dots, X_k$. The function can be described in equation 1.13 as

$$g[E(y)] = \beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k \quad (1.13.)$$

where $g$ is the link function and $\beta_0$, $\beta_1$, and $\beta_k$ are estimated regression coefficients. The link functions can be written as $g[E(y)] = E(y)$ for linear, $g[E(y)] = \log\left\{\frac{E(y)}{[1-E(y)]}\right\}$ for logistic, and $g[E(y)] = \log[E(y)]$ for Poisson regressions (Dasgupta et al., 2011).

Regression-based supervised methods use decision trees widely. The classification and regression tree (CART) (Breiman et al., 1984) is the most common method. Also there are some other methods used like C4.5 (Quinlan, 1993). In general, decision trees are classified as classification and regression trees. Although classification trees have binary or categorical outcomes, regression trees have continuous outcomes. The input or covariate space of growing decision trees is split into hypercubes, within which the outcome is relatively homogeneous. A sequence of binary splitting rules is used in a hierarchical manner for the inputs. All possible binary splits over all the inputs are efficiently examined by the algorithm. Then a split is chose to make the corresponding values of the output variable as distinct as possible. This is the maximized distance between the two groups of outputs defined by the split. Trees can be grown to purity, although this is a poor practice in general and can lead to unstable or highly variable estimates. A tree is learned using training data and then by sending the input values down the tree, the predicted outcome for each observation is determined. The most frequent class for classification trees or the mean of the outcomes for regression trees of the test data is took in the terminal node into which the observation falls. Decision trees should be optimized with respect to the number of variables used and the size or depth of the tree grown. The prediction error is the criterion that is mostly used for optimizing decision trees. The misclassification rate and the mean-square error are the prediction errors used for

classification and regression trees, respectively. Cross-validation is generally used for optimization (Dasgupta et al., 2011).

Vapnik et al. used support vector machine (SVM) first as a regression-based supervised method in 1997 (Smola and Vapnik, 1997). Support Vector Regression (SVR) relies on kernel functions and therefore it is considered as a nonparametric technique. SVR uses the same principles for classification with only a few minor differences according to the SVM. Firstly, to predict the output that is a real number and has infinite possibilities, is very difficult. A margin of tolerance (epsilon) requested from the problem is set in approximation to the SVM in the case of regression. Furthermore, the algorithm is more complicated therefore to be taken in consideration is also a more complicated reason. Even so, to minimize error, keeping in mind that part of the error is tolerated, and individualizing the hyperplane which maximizes the margin are the main ideas and are always same (Taghavifar and Mardani, 2014).

For modeling complex problems like image recognition, artificial neural networks (ANNs) are widely used. Also they are easily used for regression problems. If statistical models use adaptive weights, they can be termed an ANN and approximate non-linear functions of their inputs. So, when a more traditional regression model cannot fit a solution, usage of ANN regression for such problems is convenient. It is claimed that ANNs can learn from a set of data without the need for a full specification of the decision model and provide any needed data transformations. Also they are more durable in case of noise and distortion. Against dimensionality problems that hamper attempts to model nonlinear functions with large numbers of variables, ANNs show durable flexibility (Parsons et al., 2004).

# 2. RELATED WORK

In recent years many researchers studied on CNTs to obtain perfect CNTs and widen their application areas. Some of the studies are focused on geometry optimizations of the CNTs. Also many of the researchers that study on CNT calculations, incorporate artificial intelligence methods into their works. Nowadays, this incorporation trend has been increasing.

In this section some studies in literature are presented about DFT calculations and CNT geometry optimizations, and artificial intelligence in CNT calculations.

## 2.1. Previous Studies on Density Functional Theory and Carbon Nanotube Geometry Optimizations

DFT calculations in CASTEP were used by many researchers for CNT studies. Bauschlicher and Ricca utilized from DFT and the second-order Møller-Plesset approaches to study the interaction of NH3 with graphite and a (9,0) CNT. They used CASTEP for DFT calculations and denoted that the optimization calculations are very time consuming even running in parallel. The best estimation of their study for the NH3 binding energy is 2±2 kcal/mol. According to the results of the calculations they showed that NH3 is weakly bound to both graphite and a (9,0) CNT and the charge transfer was very little. They conclude that when NH3 was added, there was essentially no change in the nanotube band gap with band-structure calculations (Bauschlicher and Ricca, 2004). Hongxia et al achieved the structure of heterojunctions made up of an (8, 0) CNT and an (8, 0) boron nitride nanotube, and an (8, 0) CNT and an (8, 0) silicon carbide nanotube through geometry optimization implemented in the CASTEP with two distinct studies. They designed a two-probe system of the heterojunction and calculated its transport properties with a method combining nonequilibrium Green's function with DFT. They limited the bias range

-3.0 to 3.0 V to avoid from the time consuming calculations. They found out that the highest and lowest occupied molecular orbital of the heterojunction have a higher electron density on the CNT section (Hong-Xia et al., 2010, Hong-Xia et al., 2009). A numerical study of conjoined structures of graphene nanoribbons and CNTs were performed by Krasnenko et al and two types of structures were found. (i) Structures of CNT and graphene nanoribbon that formed by van der Waals forces with a distance close to 0.35 nm and (ii) structures of CNT and graphene nanoribbon that interconnected by short (0.17 nm) and strong chemical bonds. They performed all calculations in CASTEP. As the calculations were time consuming, they were made for relatively small super-cells, including several hundreds of atoms. The study showed that the latter bonds essentially perturb conjoined carbon C6 rings because of the pseudo-Jahn–Teller effect (Krasnenko et al., 2014). Mao and Zhong investigated the geometries and electronic properties of junctions constructed from armchair and zigzag CNT insertion graphene matrix using first-principles DFT calculations. They found that the insertion of the CNT into the graphene matrix leads to the formation of C–C covalent bonds between graphene and the CNT that distort the CNT geometry. The study was implemented in the CASTEP and based on the first-principles plane-wave pseudopotential DFT. Because of the larger number of atoms in the supercell and expensiveness of the calculation time, more K points along the tube axis are considered in the calculation of the band structure instead of the structural optimization (Mao and Zhong, 2009).

In many studies DFT calculations were utilized for geometry optimization of CNTs without using CASTEP. Kürti et al carried out first principles calculations using DFT to obtain the geometrical properties and the radial breathing mode frequency of 40 different SWCNTs with small diameter. They considered fourteen chiral nanotubes with diameters less than 0.8 nm and all zigzag and armchair nanotubes having a diameter less than 1.6 nm. They commented some important determination according to the results: (i) the deviation from the ideal behaviour

increases with increasing curvature, (ii) the lattice constant along the tube axis exhibits a slight shrinking, (iii) there is a general softening with increasing curvature and the softening of the radial breathing mode frequency can be described both by a simple function of the diameter, the chiral angle and the metallicity of the tube, (iv) the totally symmetric radial motion with the totally symmetric tangential motion(s) has a non-negligible effect (Kürti et al., 2003). Yagi et al investigated the interaction of 3d transition metal atoms and dimers with a SWCNT with armchair chirality by first principles DFT calculations. They showed that for Fe and Co the interactions are largely ferromagnetic with the local magnetic moments of the dimers being similar to the free dimers however most structures are nonmagnetic for Ni (Yagi et al., 2004). Akai and Saito, and Kanamitsu and Saito were studied the details of electronic structure, energetics and geometric structure of both chiral and zigzag CNTs using DFT calculations in two distinct papers. For the semiconducting nanotubes, they found the peak-to-peak separations can be classified into two types according to the chirality. Also, they showed that the distribution of bond lengths depends on the chirality. They specified that chiral nanotubes are less stable than zigzag nanotubes (Akai and Saito, 2005, Kanamitsu and Saito, 2002). Sanchez-Portal et al presented ab initio calculations on the structural, elastic, and vibrational properties of SWCNT with different radii and chiralities. They obtained these properties using an implementation of pseudopotential-DFT. The results are compared with the predictions of the simple zone-folding approximation. They concluded that except for the known deficiencies of the zone-folding procedure in the low-frequency vibrational regions, the calculations offer quite accurate results, even for relatively small radii (Sánchez-Portal et al., 1999). Moradian et al investigated the structural and electronic properties of semiconducting CNT bundles, isolated silicon carbide nanotube (SiCNT) bundles, double-walled SiCNTs, and Li-intercalated SiCNT bundles using ab initio DFT in four distinct studies. They demonstrated the effects of inter-tube coupling on the electronic dispersions of

nanotube bundles. The results showed that the inter-tube coupling decreases the energy gap in semiconducting CNT bundles. For isolated SiCNT bundles, the inter-tube interaction causes a very small radial deformation for such small diameter nanotubes, while band splitting and reduction of the semiconducting energy band gap are significant. They found that the formation energy of double-walled SiCNTs has a local minimum around the interwall distance of about 4.3 Å. The results also showed that the most prominent effect of Li intercalation on the electronic band structure is a shift of the Fermi energy which occurs as a result of charge transfer from lithium to the SiCNTs (Moradian et al., 2008, Moradian et al., 2009a, Moradian et al., 2009b).

DFT calculations are not only used for CNT problems and carbon element. There are also some studies utilized from DFT calculations for other elements. Tao et al investigated the hydrogen purification performance of two-dimensional porous graphene material using first principle DFT calculation and molecular dynamic simulation. The results showed that the selectivity of $H_2$ over several other gas molecules that often accompany $H_2$ in industrial steam methane reforming or dehydrogenation of alkanes is sensitive to the pore size of the membrane. They mentioned that the DFT calculation was highly accurate and the computational workload was very high. They only computed dozens of atoms in the simulation to save calculation time (Tao et al., 2014). Wang et al studied the chemical element-specificity, size-dependence, and boundary effects on the electronic structures and work function values of infinitely long lanthanide hexaboride and non-lanthanide hexaboride nanowires using DFT calculations in CASTEP. According to the results, they found out that the low work function of metal hexaboride nanowires is related to the distinguished metal-like electronic properties of these compounds which are attributed to the abundance of states contributed from the metal atoms positioned at the Fermi level. Also it is remarked that they achieved convergent results in calculations using the CASTEP codes however the codes are time consuming (Wang

et al., 2013). $TiO_2$ cluster anchoring on a graphene sheet is investigated using DFT calculations by Geng et al. They employed pristine grapheme (P-G), graphene with monovacancy (V-G) and graphene with epoxy (O-G) structures to represent the graphene surfaces. They calculated the geometric structures and electronic properties to clarify the interface interaction and enhanced photocatalytic mechanism. It is shown with the results that with Ti atoms connected to graphene sheet are more favorable. O-G interaction with the $TiO_2$ cluster showed stronger electron cloud overlaps and larger binding energies compared with P-G and V-G. For DFT calculations they used CASTEP and employed a small stoichiometric $(TiO_2)_3$ cluster to reduce time consuming calculations (Geng et al., 2013). Van Regemorter and Larsson theoretically analyzed the effect of an N dopant on the $CH_3$ adsorption and on the H abstraction from this newly adsorbed $CH_3$ species. They performed the calculations in CASTEP under periodic boundary conditions using DFT methods based on a spin-polarized GGA-PW91 functional and a plane-wave basis set. The geometry was first optimized using the spin-polarized local density approximation with the BFGS algorithm, and is followed by a single point GGA-PW91 calculation on the final geometrical structure to save calculation time (Van Regemorter and Larsson, 2008).

## 2.2. Previous Studies on Artificial Intelligence in Carbon Nanotube Calculations

In machine learning to classify or prediction purposes some analytic strategies were developed as a tool. Decision tree methods are some of them and they also called as recursive partitioning. In 1984, Breiman et al developed a type of decision tree methodology called as CART analysis (Breiman et al., 1984). There are not any studies about the usage of CART methodology in CNT research. Only, a CART-based meta-analysis presented by (Gernand and Casman, 2014). The study analyzes the rodent pulmonary toxicity studies of uncoated, non-functionalized CNT

exposure. Some data defects, like significant quantities of missing values, correlated variables, nonlinearity, and high dimensionality, impair other types of regression analysis and their CART models are relatively insensitive to these defects. However, CART methodology has increasingly been applied to different research areas including computer architecture, health sciences, agriculture, and traffic flow prediction etc.

Regression is a technique that is used for prediction and it is the excellent one of a group of techniques that are represented by SVMs. SVMs have simple basic theories and this situation is not a handicap for SVM methods to be very powerful in modeling applications. Its skill in the modeling of nonlinear relationships is one of the outstanding potentials of this technique (Smola and Vapnik, 1997). SVM methods have been studied for applications and modeling in CNT field.

There is an initial assumption that the gate voltage is directly proportional to the gas concentration as well as its temperature. An analytical model proposed by (Akbari et al., 2014) and it is developed with the assumption. The proposed mathematical model is obtained using the CNT conductance formulas. They obtained other models with combination of ANN and SVR algorithms for the current–voltage characteristic. In the prediction of current–voltage characteristics, SVR is performed. Moreover, results from both ANN and SVR algorithms are compared to determine the most advanced prediction method. Iteration method calculated the control parameters that are named as temperature control (α) and gas concentration control (β) parameters. More satisfactory results are obtained with SVR prediction model.

In another study, SVR is incorporated with particle swarm optimization by (Cheng et al., 2015) for its parameter optimization. Mechanical features of the CNTs/epoxy composites are predicted according to an experimental data set with the developed mathematical models. They determined that generalization skill of SVR model is very high according to the results of the leave-one-out cross-validation test

by SVR models. They calculated the statistical mean absolute percentage error as 3.96% and the correlation coefficient as 0.991 for tensile strength, 3.14% and 0.990 for elongation, and 2.62% and 0.997 for elastic modulus, respectively. Two usage area of the developed SVR model are proposed. First one is the usage of the model to accurately predict the mechanical features of CNTs/epoxy composites and the second is to make optimal controlling or designing of the experimental process/geometry in practice.

An accurate method is proposed by (Ensafi et al., 2010) to find out the electrochemical of enrofloxacin. Also the main metabolite of enrofloxacin named as ciprofloxacin is determined using a least-squares SVM and linear sweep voltammetry at a multiwall CNTs/glassy carbon electrode. In the voltammetric method, the oxidation potential of ciprofloxacin is affected by enrofloxacin with the usage of multiwall CNTs/glassy carbon electrode as a working electrode. However enrofloxacin is not affected from ciprofloxacin. Measurement of enrofloxacin is possible without any interfering effect of ciprofloxacin and to make a connection from a ciprofloxacin voltammogram to its concentration a least-squares SVM model is developed. The concentration of ciprofloxacin is predicted in spiked human plasma and pharmaceutical samples with the model. It is a fast and simple method and moreover there is no hazardous chemical wastes generated with it.

(Salehi et al., 2016) have been applied SVM approach to predict the adsorption of Cu(II) on chitosan/poly vinyl alcohol membrane adsorbents comprising different amounts of aminated multi-walled CNTs. 72 experimentally measured equilibrium adsorption data set are used to develop and test the model. Membrane types, initial ion concentrations and temperature were selected as input parameters for the model. Equilibrium adsorption results were assigned as output parameters. The data points were categorized into three random sets: 70% assigned for the model development, 15% for validation and the remaining 15% for verification of the model applicability. The statistical Leverage analysis was

employed for the examination of the outlier and doubtful data. According to the results an accurate and reliable model was developed. The average absolute relative deviation is 4.8% and correlation coefficients close to unity. Furthermore, the actual tendency of the equilibrium adsorption as an initial metal concentration function at different temperatures can be appropriately predicted with the proposed model.

In recent years the application of ANN has been developed as a powerful tool in solving problems in which the mathematical modeling are neither possible nor practical. Simple elements that operate in parallel form ANNs and these networks are used in engineering applications as strong tools for simulation and prediction. There are many studies in literature concerning ANN algorithms which has used CNT modelling and simulation.

An ANN is applied by (Hayati et al., 2010) to CNT metal–oxide-semiconductor field-effect transistors for modeling and simulation. The data derived from moscnt.1.0 that is a MATLAB script, is used to improve the proposed ANN model. The computational time is decreased by the model furthermore the accuracy of physics-based model such as non-equilibrium Green's function formalism is kept. A subcircuit is developed with the proposed model and imported into HSPICE software. According to the results the model can be used to simulate nanoscale circuits and this circuits can also be used as a block in circuit simulators like HSPICE, ADS and etc.

(Abo-Elhadeed, 2012) developed and proposed a model using ANN approach for CNT field-effect transistors. Proposed model predicts the current–voltage characteristics accurately for different structures of CNT field-effect transistors. The circuit simulator Eldo is used to implement the model. General user defined model template employed for the implementation. The current–voltage characteristics and device simulation results are compared to verify the accuracy of the proposed ANN model. Empiric data is used for both schottky barrier and conventional CNT field-effect transistors to approve the proposed model. The model

fit with under 1% average percentage error for both the empiric and device simulation data.

Two synthesized nanofluids' experimental conditions are predicted with a proposed ANN model by (Shanbedi et al., 2013). These nanofluids includes CNT/water and CNT-Ag/water. The tests were performed for different values of weight fraction and input power to determine the temperature performance of a two-phase closed thermosyphon. A tool that was reliable and certain should be applied to predict the other experimental conditions. The operating parameters were determined using ANN. The parameters included the temperature difference of condenser part between the input and the output water flows and wall temperature distribution. The learning algorithm of the network used the Levenberg–Marquardt (Moré, 1978) algorithm. The simulation and experiment results were very compatible with each other in a good way. According to the results, the study showed that the performances of different processes are predicted by using a powerful tool ANN.

(Hassanzadeh et al., 2015) used a radial basis function neural network for quantitative structure-property relationships modeling of adsorption on multi-walled CNTs surface. They combined the network with genetic algorithm (GA) in the study. They selected initial centers randomly from the calibration set in the proposed model. The locations of the initially selected centers are changed by GA to find the optimum positions of centers from the whole space of scores matrix. In this situation, highest prediction ability is obtained.

Some quantitative structure-property relationship models are formed by (Rahimi-Nasrabadi et al., 2015) to predict the CNTs adsorption of aromatic compounds. Stepwise MLR techniques and self-organizing map are combined to select five descriptors. With the help of linear and nonlinear modeling techniques, these five descriptors are used to connect the structure of the selected chemicals with their adsorption descriptor. MLR model had R of 0.93 and root-mean square error of

0.36 and multilayered perceptron neural network model had 0.99 and 0.29 respectively. This means that the developed nonlinear model had superiority over MLR model. According to the cross-validation test results, MLR neural network is reliable to predict the adsorption descriptor values for the aromatic contaminants. The study showed that the much affecting parameters to the adsorption of the compounds are molar volume and hydrogen bond accepting ability and also the adsorption of organic compounds by CNTs can be predicted with the developed ANN based model QSPR.

Rapid adsorption of Triamterene is investigated by (Ghaedi et al., 2016) using MWCNT and SWCNT. They modeled the impact of influential variables using MLR and ANN. Temperature, initial drug concentration, amount of adsorbent and contact time are some of influential variables. MATLAB was applied to predict the adsorption efficiency. The employed ANN had three layers. Tangent sigmoid and linear transfer function are used at hidden and output layers, respectively. Also Levenberg–Marquardt (Moré, 1978) back-propagation algorithm is used at output layer with 1000 iterations. 70% of data (45 samples) randomly are selected for training and %30 (19 samples) for testing sets. Adsorbent dosage, concentration and contact time neurons were applied in the input layer, 1 to 25 neurons in the hidden layer and removal percentage neuron in the output layer. Very large or small weights causes numerical overflows and to avoid of this they normalized the data between 0.1 and 0.9. According to the results, to predict the adsorption efficiency of the process, proposed ANN model is more convenient than MLR. With the ANN model, they obtained determination coefficient as 0.980 and mean squared error as 0.002 on SWCNT, and 0.986 and 5.4e-04 on MWCNT, respectively.

A new artificial intelligent based soft computing framework is developed by (Raja et al., 2016) with the help of FFNNs, GAs and Interior-Point Algorithm to solve nano-fluidic problems. MWCNT along a flat plate with Navier slip boundary has some nano-fluidic problems caused by heat transfer and fluid flow.

Fitness function of the network is built using differential equations of Feed Forward ANN models. GAs and Interior-Point Algorithm are used together for viable local convergence to determine the learning of weights of the network. By taking different base fluids and concentrations of MWCNTs, many scenarios are tested for nano-fluid dynamics. Performance of this model is determined with comparison of the results obtained from fully explicit Runge-Kutta state of art numerical solvers in MATHEMATICA software package.

In the studies summarized in this section, which have been examined so far, the use of artificial intelligence methods for atomic coordinate estimation has not been found. In this thesis, the use of artificial intelligence for the estimation of atomic coordinates is a new field of application in terms of literature.

# 3. MATERIALS AND METHODS

In this thesis, CASTEP software is used for the CNTs simulation, the calculation of geometry optimization and the output file creation for the initial and calculated atomic coordinates of related chiral vector. At the same time the calculated atomic coordinates of CNTs are estimated using regression-based supervised machine learning methods such as ANN, CART, MLR and SVR.

In rest of this section brief information about CASTEP is given, then the utilized machine learning methods are explained, and finally cross validation method is mentioned for assessing performance.

## 3.1. Electronic Structure Calculations of Carbon Nanotubes using CASTEP

The CASTEP software (Segall et al., 2002) is used to perform electronic structure calculations as a first principles quantum mechanical code. A wide range of materials like surfaces, crystalline solids, liquids, molecules and amorphous materials can be simulated with CASTEP within the density functional formalism (Clark et al., 2005).

Depending on the finite speed and memory of the used computer, CASTEP can calculate the properties of any material that formed from an assembly of nuclei and electrons. This is an ambitious approach that aims to rely on quantum mechanics instead of to use experimental (empirical) data (Clark et al., 2005).

The ground state electronic structure of a system is determined with a method that is given by DFT (Kohn et al., 1996) when solving the N-electron Schrödinger equation directly is not possible. The implemented electronic structure calculations methodology in CASTEP can be summarized as follows: using the plane-wave pseudopotential approach, a set of one-electron Schrödinger (Kohn-Sham) equations are solved (Payne et al., 1992). Periodic boundary conditions and Bloch's Theorem are used to define a plane-wave basis set that the wavefunctions

are expanded in. Means of ab initio pseudopotentials describe the electron-ion potential within both norm-conserving (Kleinman and Bylander, 1982) and ultrasoft (Vanderbilt, 1990) formulations. The electronic wavefunctions and corresponding charge density are obtained self-consistently using direct energy minimisation schemes. Density mixing (Kresse and Furthmüller, 1996) schemes and conjugate gradient (Payne et al., 1992) are especially implemented. Also, systems with partial occupancies particularly metals can use the robust electron ensemble DFT (Marzari et al., 1997).

CASTEP has numerous capabilities and it is a fully featured first principles code. The total energy derived from many other quantities is the basic quantity aiming to calculate any physical property of the system from first principles. For example stresses are given by the derivative of total energy with respect to atomic positions results in the forces and with respect to cell parameters. Then they are used for performing full geometry optimizations and possibly finite temperature molecular dynamics. Moreover, symmetry and constraints can be imposed in the calculations that defined by the user or automatically using in-built symmetry detection (Clark et al., 2005).

CASTEP can calculate many physical properties of materials and some of them can be listed as follows (Clark et al., 2005):

- Total energies: Calculation of total energy, stresses, forces, and elastic constants.
- Electronic structure: Electronic charge densities, electrostatic potentials, orbitals, total and partial electronic density of states, band structure, optical properties, and Mulliken population analysis.
- Geometry: Optimization of unit cell parameters and atomic positions, constrained/unconstrained and under external pressure and stresses.

- Molecular dynamics: Performing of finite temperature molecular dynamics under constant temperature, volume, energy, and pressure.
- Transition states: Using the Linear Synchronous Transit/Quadratic Synchronous Transit methods.
- Phonons: Phonon frequencies and eigenvectors are calculated using density functional perturbation theory.
- Electric field response: The response of the system of an external electric field yielding dielectric constants, bulk polarisabilities, longitudinal optical/transverse optical phonon splittings, Born effective charges and the infrared intensity of phonon modes using density functional perturbation theory.
- Exchange and correlation: Many functionals like Local Density, gradient-corrected and weighted density approximations, Hartree-Fock and exact/screened exchange are available.

### 3.2. Artificial Neural Networks

ANNs are a family of massively parallel architectures that are capable of learning and generalizing from examples and experience (Hakim, 2006). 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 (Ghaboussi et al., 1991). For forecasting and modeling of engineering problems, different types of ANN models are used in the literature. Although some studies have been done on CNT modelling, ANN is also used in other simulation and modelling areas such as multiprocessor architecture: ANN-based prediction models are proposed by (Zayid and Akay, 2013) to predict a message-passing multiprocessor architecture's performance measures. For prediction of the performance measures of the simultaneous optical multiprocessor exchange bus architecture, they used multi-

layer FFNNs. The architecture employs the message-passing programming model with acknowledges. They used OPNET Modeler (Lu and Yang, 2012) to simulate the message-passing simultaneous optical multiprocessor exchange bus architecture statistically. The model contains node number, traffic pattern, ratio of the mean message channel transfer time to the mean thread run time and thread number as input variables, and average channel utilization, average input waiting time, average channel waiting time, average processor utilization and average network response time as output. SEE and R values are calculated for the prediction models. All the results obtained from the models and GRNN, MLR and SVR are compared. According to the comparison, to predict the performance measures of a message-passing multiprocessor architecture, multi-layer FFNNs has a better performance against GRNN, MLR and SVR analysis.

An ANN and multiple regression based approach is proposed by (Asiltürk and Çunkaş, 2011) for prediction of the surface roughness in AISI 1040 steel. They implemented a full factorial experimental design to search the effect of the cutting parameters like feed rate, cutting speed and depth of cut on the surface roughness. The analysis of variance is used for testing multiple regression models. In neural network they used multilayer perception architecture and the architecture employed with back-propagation algorithm that has two different variants. They compared the performances of both ANN and multiple regression based models with the help of statistical methods. The prediction capability of the proposed methods are calculated with measured values. The results showed that the measured and predicted values are close and the model are effective for prediction of the surface roughness in turning process. The R value is achieved as 99.8% training and 99.4% for testing data in ANN and as 98.9% for MLR models. This situation showed that ANN performed better than MLR.

In this study, four ANN models are considered as follows.

### 3.2.1. Feed-Forward Neural Network and Function Fitting Neural Network

The simplest ANN model is FFNN and it has three layers named as input, hidden and output (Figure 3.1). 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 backwards through the system. Afterwards the weight coefficients are updated (Şen, 2004).

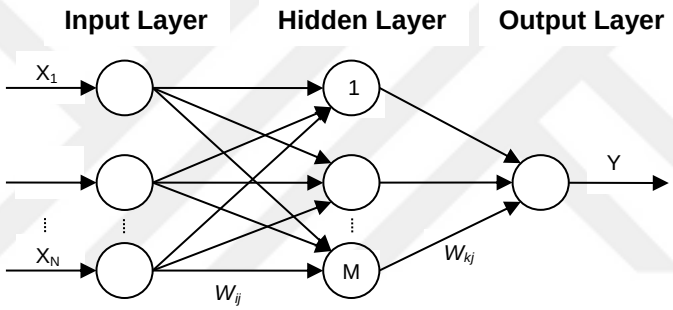
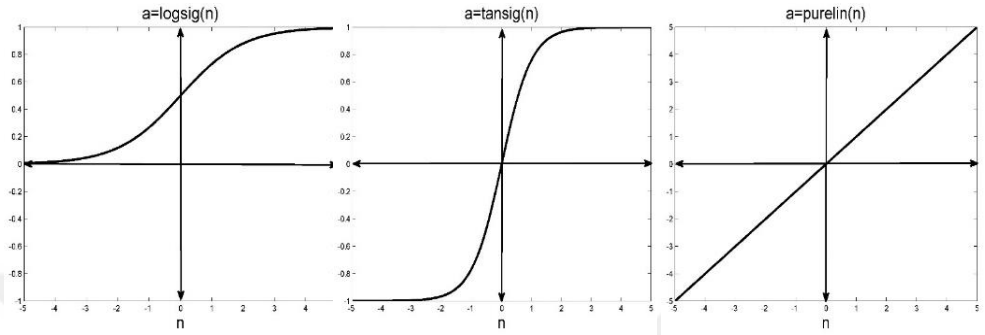



Figure 3.1. Schematic diagram of FFNN

In Figure 3.1, $X_i$ 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 (Beale et al., 2012).

ANN transfer functions are the way to simulate phenomena's reaction using input and out parameters (Patil, 2008). The Log-sigmoid transfer function (LOGSIG) is one of the most commonly used functions (Figure 3.2.a). The input is squashed the output into the range 0 to 1 by this transfer function. Hyperbolic tangent transfer function (TANSIG, Figure 3.2.b) in the term of neural networks, is related to a bipolar sigmoid which has an output in the range of -1 to +1. Pure linear transfer

function (PURELIN) (Figure 3.2.c) can be used representation of the values that is close enough to linear (Dorofki et al., 2012).




(a) (b) (c)

Figure 3.2. Transfer functions (a) LOGSIG (b) TANSIG (c) PURELIN

A type of FFNN is used called as FITNET to fit an input output relationship. 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 (Beale et al., 2012).

### 3.2.2. Cascade Forward Neural Network

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 (Hedayat et al., 2009). 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 (Figure 3.3) (Beale et al., 2012).

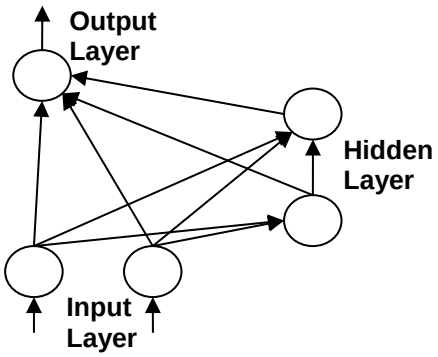



Figure 3.3. Schematic diagram of CFNN

### 3.2.3. Generalized Regression Neural Network

GRNN is a variation of the radial basis neural networks and it is based on kernel regression networks. Unlike back propagation 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 (Kim et al., 2004).

Input, pattern, summation and output layers form a GRNN as shown in Figure 3.4. In the input layer, each input corresponds to individual process parameter. Input layer is fully 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 are divided by each D-summation neuron, yielding the predicted value $Y_i'$ (equation 3.1) to an unknown input vector $x$ as (Kim et al., 2004):

$$Y_i' = \frac{\sum_{i=1}^{n} y_i \,.\, \exp[-D(x, x_i)]}{\sum_{i=1}^{n} \exp[-D(x, x_i)]} \tag{3.1.}$$

$$D(x, x_i) = \sum_{k=1}^{m} \left(\frac{x_k - x_{ik}}{\sigma}\right)^2 \tag{3.2.}$$

$Y_i'$ represents the weight connection between $i^{\text{th}}$ 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 $k^{\text{th}}$ element of $x$ and $x_{ik}$ is $k^{\text{th}}$ 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 (Jang et al., 1997, Kim et al., 2004).

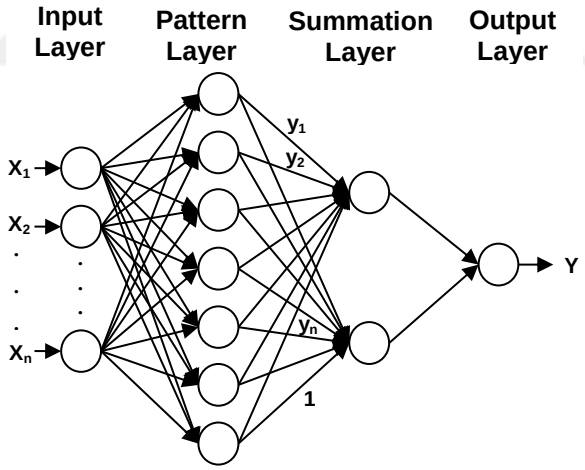



Figure 3.4. Schematic diagram of GRNN

### 3.3. Support Vector Regression

The support vector is a statistical learning theory based approach that was introduced in the sixties. Estimation of a function from the given data set is the aim of the statistical learning theory and it remained theoretical until the nineties. SVM learning algorithm that leads the theory becoming in practice, was proposed based on this theory in the mid-nineties (Vapnik, 1995). An optimal separating hyperplane is searched by SVMs from a given collection of data. Data samples can be separable with a linear hyperplane because they are mapped to a high-dimensional feature space. An adequate kernel function has to be chosen while the mapping is non-linear. Thus, two separated classes are revealed with a maximized margin from each other (Uslan, 2015).

SVMs can be used both for classification and real-value estimation tasks. SVR is the regression form of SVM that shows superior performance in many applications. $\epsilon$-insensitive loss function is used in SVR as shown in Figure 3.5. Tolerating against noise is one of the advantages of this function. A linear function $f(x)$ approximated in equation 3.3 by SVR as (Schölkopf and Burges, 1999)

$$f(x) = w^T x + b \tag{3.3.}$$

where w is weight vector and b is bias term. The following optimization problem constrains to this linear function (Schölkopf and Burges, 1999):

$$\min \frac{1}{2}\|w\|^2 + C\sum(\xi_+ + \xi_-) \tag{3.4.}$$

where $\xi_+$ and $\xi_-$ are the two nonzero slack variables in both directions. With an admissible parameter $\epsilon$, the bounded area aim to fit the data. The constant parameter C > 0 is the trade-off that it optimizes equation 3.4 between the complexity of the function and toleration up to the distance value of data samples outside the bounded

region (slack variables) which deviate greater than $\epsilon$. The support vectors are the data samples that are outside of the bounded zone within the distance of slack variables. The minimization function (equation 3.4) is subject to (Burges, 1998):

$$\begin{aligned} &y - (w^T x + b) \leq \epsilon + \xi_+ \\ &(w^T x + b) - y \leq \epsilon + \xi_- \\ &(\xi_+, \xi_-) \geq 0 \end{aligned} \tag{3.5.}$$

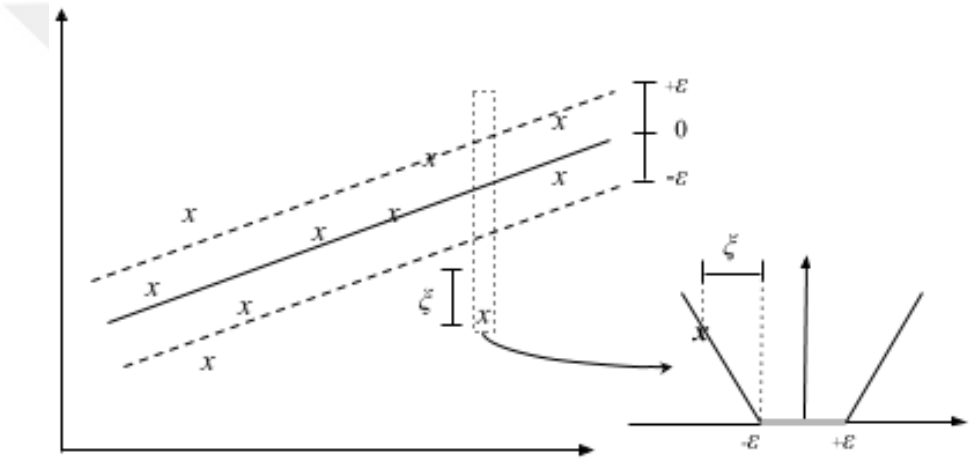



Figure 3.5. $\epsilon$ -insensitive loss function for a linear SVM (Uslan, 2015)

The method of standard dualization can be used to solve the constrained optimization problem equation 3.4 and 3.5. With the help of a dual set of parameters, the optimization function is reformulated dual formulation using the Lagrange multipliers. The following solution (equation 3.6) is yielded by dual optimization problem after a set of steps (Burges, 1998):

$$f(x) = \sum_{i=1}^{n} (\alpha - \alpha^*) K(x_i, x) + b \tag{3.6.}$$

where $\alpha$ and $\alpha^*$ are Lagrange multipliers and $K(x_i, x)$ represents the kernel function.

The non-linear input space can be mapped to the high-dimensional feature space by the kernel function and so that a linear solution may be possible. It is not easy to determine which kernel function can be used in the support vector based approach. It may depend on several factors, particularly depends on the data set that is being used. After determination of the kernel function, C, $\epsilon$ and kernel parameters are set properly. Thus, a suitable SVR solution that can best model the data set in use, can be leaded by a proper set of parameters. A better generalization performance can be expected from the constructed SVR model after properly selection of parameters (Uslan, 2015).

Efficiency of the support vector approach for very large data sets is another common limitation of it. It can be very hard to train such data sets as of the availability of the millions of support vectors. Fixing of kernel function and the training for very large data sets are still remain open research issues for SVMs (Uslan, 2015).

### 3.4. Classification and Regression Tree

(Breiman et al., 1984) introduced a statistical procedure named as CART. While the aim is to classify an object into two or more populations, CART is primarily used as a classification tool. CART uses the same technology and procedure to analyze both categorical and continuous data. There are three stages of the methodology outlined by (Breiman et al., 1984) Breiman et al. (1984). In the first stage a splitting criterion is utilized to grow the tree using a recursive partitioning technique to select variables and split points. Several criteria like gini, twoing and ordered twoing are used to determine the splits. Both primary and surrogate are identified and selected. Surrogate variables are closely related to the original splits and they have missing values for the primary variables and may be used in classifying observations.

In the second stage, a pruning procedure is used for the CART methodology after identifying a large tree. This procedure embodies a minimal cost complexity measure and its result is a nested subset of trees starting from the largest tree grown and continuing the process till one node of the tree remains. Estimates of future classification errors are provided for each subtree using cross-validation or a testing sample. If only small numbers of data points are available, cross-validation is used in building the CART models (Lee et al., 2006).

Selecting the optimal tree is the last stage of the methodology. The optimal tree corresponds to a tree that yields the lowest cross-validated or tests set error rate. In this stage, trees are unstable. Small sized and comparable accurate trees will be alternatively chosen to avoid this instability. This process is referred to as the one standard error rule and can be tuned to obtain trees of varying sizes and complexity. By observing the drop in the error rate, a measure of variable importance can be achieved when another variable is used instead of the primary split. Essentially, if a variable appears frequently as a primary or surrogate split, assigned importance score will be higher (Lee et al., 2006).

In literature there are many studies utilized from CART for prediction. A new model that predicts hard drive failure proposed by (Li et al., 2014). The model is based on CART and with Analysis, Self-Monitoring and Reporting Technology attributes good prediction performance is achieved. On a real-world dataset with 25,792 drives, over 95% of failures are predicted by CART model at a false alarm rate under 0.1%. They also aimed to apply prediction models practically with this study. Accordingly the model is tested with different drive families, with different model updating strategies, and with fewer number of drives. Results show that the CART model still performs good and steady performance. Also a CART based health degree model has proposed at this study. With this model, a health assessment is given to the drive instead of a simple classification result. Thus, in order of their health degrees the prediction model raises warnings and the approach can deal with

them. For RAID-6 systems, a reliability model has implemented with proactive fault tolerance and they show that the reliability of large-scale storage systems can be critically improved and/or construction and maintenance cost can be reduced by the CART model.

A survival prediction model have been proposed for liver cancer by (Chen et al., 2011) using CART. Satisfied results are obtained with liver cancer data of 227 patients those were newly diagnosed between 2004 and 2008. Using chi-square test and t-test, nine variables relevant to liver cancer survival were analyzed according to literature review, and expert consultation. The analyses showed that six variables are notable. The model was tested with one variable, six notable variables and all nine variables. One variable tests are done in clinical stage alone, nine variables consists of notable and non- notable variables. The output prediction was 5-year survival. Accuracy, specificity, positive and negative predictive values, sensitivity and the area under the receiver operating characteristic curve are used as performance measures. Performance results are obtained as 0.87, 0.87, 0.88 and 0.915 for accuracy, specificity, sensitivity and the area under receiver operating characteristic curve, respectively.

When confronting gill-associated virus, a prediction and decision model for prawn farmers was proposed by (Elliott and Owens, 2015). They aimed to use CART analysis to develop individual decision trees for production factors to provide the farmer with predictive models for crops of Penaeus monodon. The combination of Gill-associated virus, environmental factors, production parameters and two bio-indicators heat shock protein and ubiquitin are the chosen factors. These information are often measured by farmers however not used to make management decisions. According to the research variation and cross-validated variation are determined. The variation is relatively stable with yield, biomass and average body weight being the most accurate whilst and have values 80, 79 and 73 % respectively. Also cross-validated variation are stable and obtained percentages are 41, 53 and 55 for yield,

biomass and average body weight respectively. On the other hand fludarabine, cyclophosphamide and rituximab, and survival were less so. According to the study due to the time scale of measurements involved, the biomarkers were less useful possibly.

An original CART-based method presented by (Xu et al., 2013) to predict the short-term traffic volume. The volume predicted through clustering the current state vector into the most similar historical pattern and regression model. The 15 minutes average traffic volumes on freeways are used for testing the proposed method. Then the method is compared with a nonparametric and parametric method. K-nearest neighbors and Kalman filter models are chosen as nonparametric and parametric methods respectively. In view of results, for both the mean absolute percentage and the mean absolute scaled errors, CART-based prediction method perform better than the k-nearest neighbors and Kalman filter methods.

### 3.5. Multiple Linear Regression

Extension of a simple linear regression named as the multiple linear regression models and incorporate two or more explanatory variable in a prediction equation for a response variable. Because of its power and flexibility, multiple regression modeling is a reference point for statistical analysis in most fields. Estimation of very complicated models with large numbers of variables can be done with MLR models using very little effort (Zayid and Akay, 2013).

$$y_i = B_0 + B_1 x_{i1} + B_2 x_{i2} + \ldots + B_n x_{in} + E_i \qquad (3.7.)$$

is the general model of multiple regression where $i = 1, 2, \ldots, n$; $x$ is the independent parameter, $B_i$ is the residual and $E_i$ is the difference between the value of the dependent variable predicted by the model and the dependent variable (Norman, 1998).

MLR tries to find a mathematical relationship between groups of random variables. The created relationship that approximates all the individual data points best, is in the form of a straight line (linear). The equation 3.8 can be written as

$$LP_i = B_0 + B_1 x_{i1} + B_2 x_{i2} + \ldots + B_n x_{in} \quad (3.8.)$$

where $LP_i$ is the linear predictor that is the $y_i$ value predicted by the input variables. So, $E_i$ is the error term calculated from the difference of $y_i$ and $LP_i$ (Norman, 1998).

Chosen estimates $B_0 + B_1 + B_2 + \ldots + B_n$ minimize the sum of squares of the predicted error to fit the models. These estimates are named as ordinary least squares estimates. The fitted values and observed residuals are calculated using these estimates. Also the error term is estimated by the residuals (Norman, 1998).

MLR can be used along wide areas like analysis of covariance that is adjusting the effects of an input variable on a continuous output variable for the effects of confounders. Also the simultaneous effects of a number of categorical variables on an output variable can be analyzed by MLR. On the other hand, it is used to predict a value of an outcome, for given inputs (Zayid and Akay, 2013).

MLR is a statistical tool and it is the best known regression technique for prediction problems. The relation between multiple independent variables and a dependent variable is determined by the method. After the development and widespread use of regression-based supervised methods, MLR has become a common method used to compare the results that are obtained with various supervised methods. Another use of MLR in supervised method is developing hybrid algorithms which combine statistics and machine learning. There are a lot of studies concerning both regression with MLR and supervised methods. Some of them are (Asiltürk and Çunkaş, 2011, Ghaedi et al., 2016, Zayid and Akay, 2013)

### 3.6. Cross-Validation Method for Assessing Performance

Cross-validation assesses performance of both supervised and unsupervised learning methods and tunes parameters or models for optimal performance. Cross-validation is a general method and k-fold cross-validation is a standard version of it. Firstly, the data are partitioned into k random subsets in k-fold cross-validation. Then the algorithm of interest like learning machine and regression model is generated or trained on k-1 of the subsets and applied or tested on the remaining subset. This procedure is repeated over all the k possible arrangements of the subsets into these two groups. The measure of model performance of the algorithm leading to k estimates in the test set is computed at each iteration. The measure of model performance will be distance matrix for unsupervised clustering and prediction error for supervised learning. The average of these k performance estimates is the final model performance estimate. So, the algorithm is being trained and tested on statistically independent subsets and this is the important idea (Dasgupta et al., 2011).

There will be three comments applied. Firstly, generally k is chosen between 5 and 10. Otherwise, the variance of the estimate is tended to increase and became unstable with k value under 5. On the other hand, the estimation in terms of bias or variance is not improved with k value more than 10. Secondly, the segmentation into groups must be done to obtain truly representative groups of the population. Lastly, each subject in the data set has been used several times both as test and training object and for this reason forming an averaged model or estimate after the repeated training and testing is not quite the same as generating a single model or estimate on all the data. Generally, good estimates of a model or algorithm's performance is leaded with cross-validation (Kohavi, 1995).

A particular algorithm can be tuned with the performance measure obtained from k-fold cross-validation. For example, the set of independent variables that gives the lowest cross-validation error can be chosen for use in a final model in linear

regression. The number of branches to keep from a tree is determined with the help of cross-validation error in decision trees. Also, in cluster analysis, the number of groups in a particular data set is determined with it (Dasgupta et al., 2011).

# 4. RESULTS AND DISCUSSION

A dataset is generated in CASTEP for different chiral vectors and 10,721 data samples are obtained. MSE, MAE, SEE and R values are utilized as performance measures while evaluating the proposed models. For all proposed models all of the operations are done in MATLAB. As a result of the studies the performance measures are calculated for all models using obtained dataset with or without using cross validation method. At the same time, the predicted data according to the proposed models are used to decrease the number of iterations in calculation process in CASTEP. This operation is a two-staged simulation model and good levels of decline in iteration numbers are achieved for proposed models.

This section presents the dataset and its properties, used performance measures, and detailed performance results of the ANN, CART, MLR and SVR prediction models and performance improvement of CASTEP simulation after using predicted data.

## 4.1. Dataset

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 4.1 shows building step of a simulation where chiral vectors are selected for SWCNTs 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 Figure 4.2. CNT calculation parameters are used as default parameters to get general data. Figure 4.3 depicts a screenshot of CASTEP calculation interface.

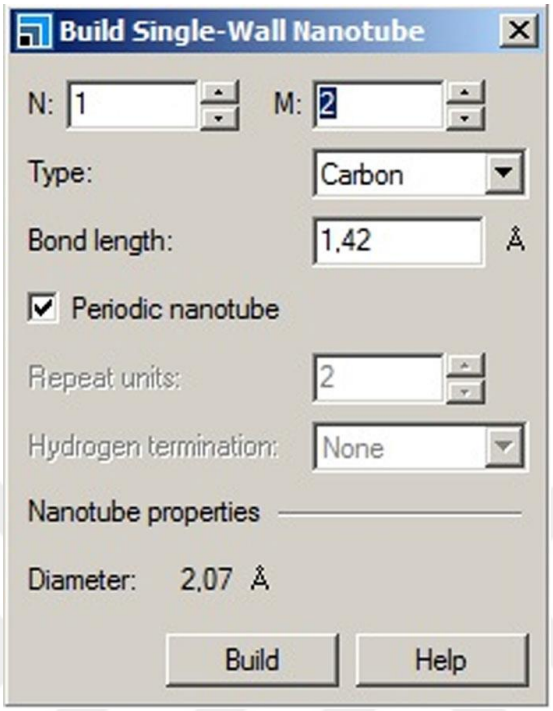



Figure 4.1. Building single-wall nanotube in CASTEP (Castep, 2016)

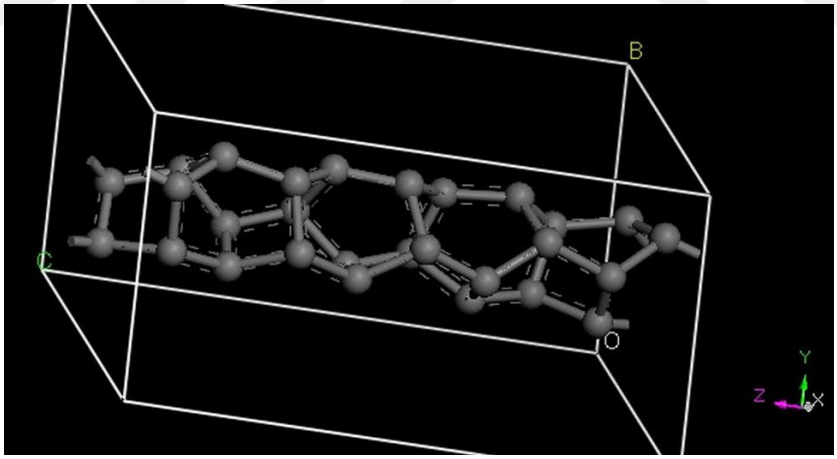



Figure 4.2. A screenshot of simulated CNT in CASTEP (Castep, 2016)

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 to finalize the computation. This parameter also determines the calculation level of inputs and outputs. The default value of the parameter is $1x10^{-5}$ eV per atom and is usually suitable (Castep, 2016). Examples of the charts of energy and convergence changes during the iterations are depicted in Figure 4.4 and 4.5, respectively.

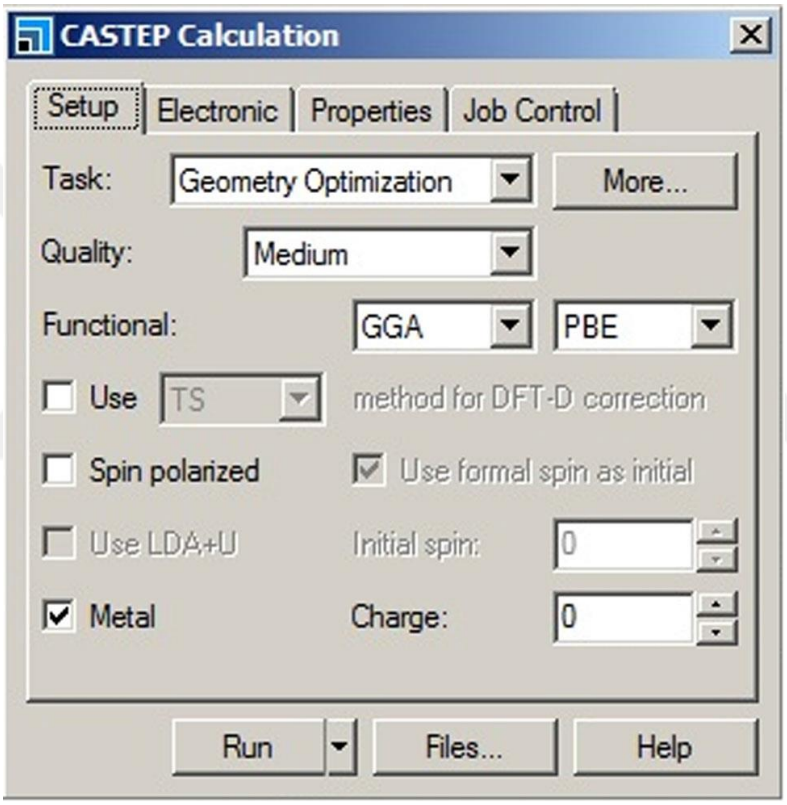



Figure 4.3. Calculation interface of CASTEP (Castep, 2016)

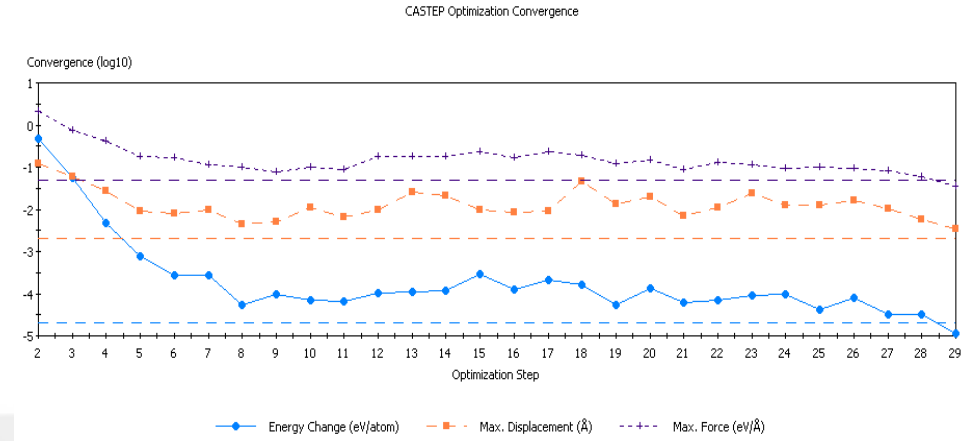



Figure 4.4. The chart of energy change during the iterations (Castep, 2016)

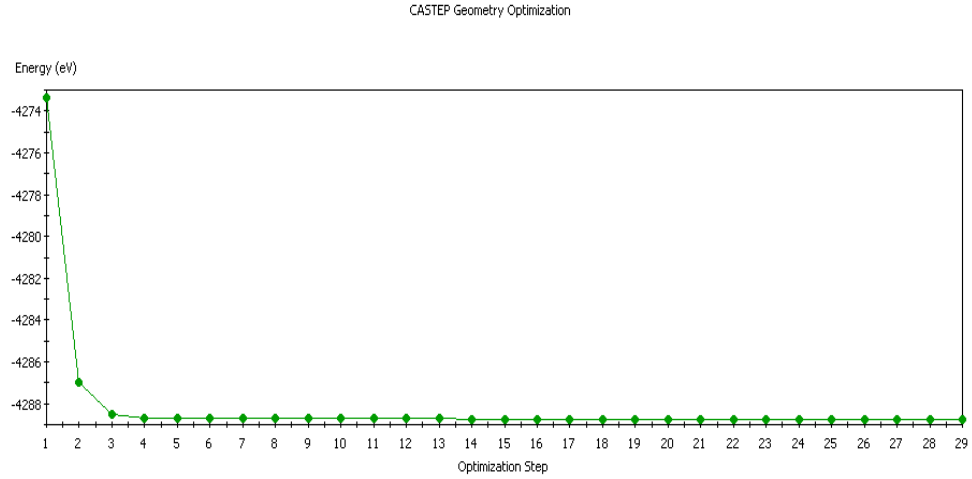



Figure 4.5. The chart of convergence change during the iterations (Castep, 2016)

After the calculation step, CASTEP creates many files and saves all calculated data in these files for each CNT simulation (Figure 4.6). SWNT.castep file is used as output file to create the input and the output datasets.

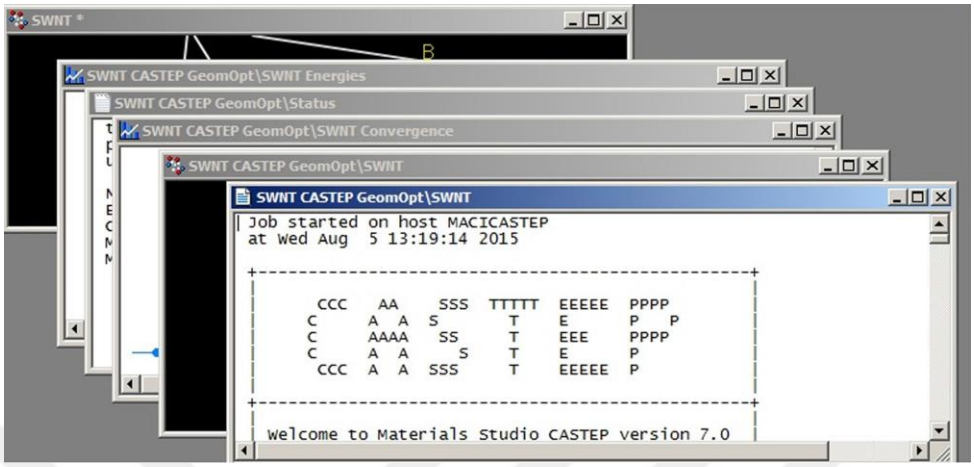



Figure 4.6. Data files created by CASTEP after calculation (Castep, 2016)

All calculations were run on a workstation that has a 2.0 Ghz power on 2 processors with 4 cores and 8 GB of RAM, using all cores under Linux operating system. The calculation time (Figure 4.7) according to these calculations are approximately varies from 10 minutes to 5 days as given in Table 4.1 in seconds. These calculation times will take weeks and months for higher atom numbers.

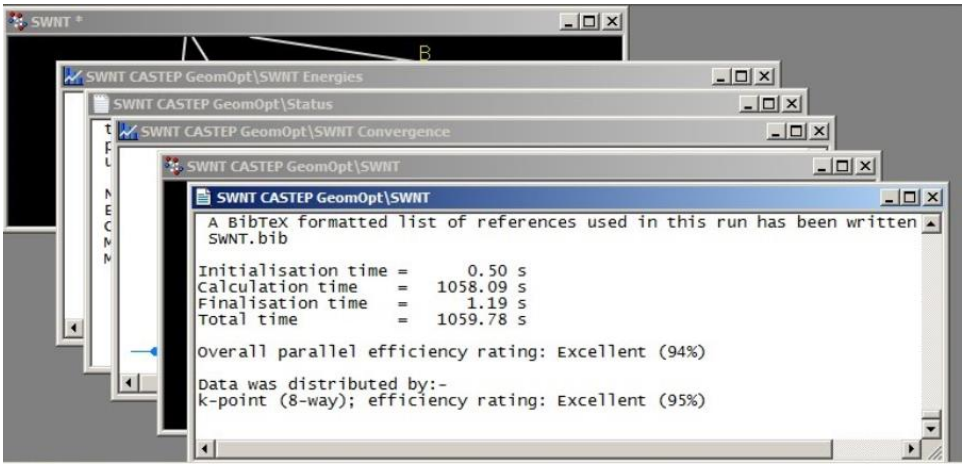



Figure 4.7. Calculation time by CASTEP (Castep, 2016)

Table 4.1. The calculation time for CNT simulations in seconds

| ID of CNT | Number of Atoms | Calculation Time (Seconds) |
|---|---|---|
| **1** | 28 | 1050.14 |
| **2** | 52 | 1256.40 |
| **3** | 76 | 2254.18 |
| **4** | 84 | 2816.61 |
| **5** | 56 | 834.04 |
| **6** | 148 | 18496.07 |
| **7** | 124 | 33280.80 |
| **8** | 156 | 9604.28 |
| **9** | 196 | 59998.78 |
| **10** | 244 | 19389.07 |
| **11** | 172 | 64102.59 |
| **12** | 104 | 7207.22 |
| **13** | 84 | 6842.67 |
| **14** | 152 | 7715.19 |
| **15** | 364 | 78421.39 |
| **16** | 228 | 13513.07 |
| **17** | 268 | 27799.53 |
| **18** | 316 | 35982.11 |
| **19** | 372 | 94767.02 |
| **20** | 436 | 266027.27 |
| **21** | 508 | 233190.94 |
| **22** | 292 | 27792.55 |
| **23** | 168 | 13292.91 |
| **24** | 388 | 106542.26 |
| **25** | 112 | 6838.91 |
| **26** | 516 | 201214.72 |
| **27** | 296 | 34136.32 |
| **28** | 364 | 103759.38 |
| **29** | 412 | 95275.60 |
| **30** | 156 | 9558.96 |
| **31** | 532 | 427956.63 |
| **32** | 228 | 27392.57 |
| **33** | 444 | 159893.39 |
| **34** | 248 | 74997.74 |
| **35** | 312 | 40999.97 |
| **36** | 140 | 10014.92 |
| **37** | 392 | 83173.05 |
| **38** | 588 | 397369.36 |
| **39** | 344 | 89653.65 |
| **40** | 252 | 26127.05 |
| **41** | 208 | 23123.48 |
| **42** | 168 | 11826.42 |

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. Cross-validation techniques that compared the actual values with the estimated values are used to determine the degree of accuracy. 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 90% of the training data and 10% of the test data. The generated prediction models are evaluated by whether using tenfold cross-validation or without using cross-validation. A summary of the descriptive statistics for the dataset is given in Table 4.2. These statistics include minimum, maximum, mean and standard deviation values of the dataset.

Table 4.2. Descriptive statistics for the dataset

| | **Inputs** | | | | | **Outputs** | | |
|---|---|---|---|---|---|---|---|---|
| | **u** | **v** | **w** | **m** | **n** | **u'** | **v'** | **w'** |
| **Minimum** | 0.0451 | 0.0451 | 0 | 2 | 1 | 0.0385 | 0.0389 | 0 |
| **Maximum** | 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 |
| **Standard Deviation** | 0.2900 | 0.2900 | 0.2900 | 2.1400 | 1.6800 | 0.2909 | 0.2910 | 0.2884 |

### 4.2. Performance Measures

All prediction models are evaluated in terms of four performances measured, (1) $R$ value 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, (4) $SEE$ calculates the standard error of the estimates of the prediction models. Summaries of mathematic equations of these

performance measures are given in equations 4.1, 4.2, 4.3 and 4.4, respectively (Witten and Frank, 2005).

$$R = \sqrt{1 - \frac{\sum_{i=1}^{n}(O_i - P_i)^2}{\sum_{i=1}^{n}(O_i - O_m)^2}} \quad (4.1.)$$

$$MSE = \frac{1}{n}\left[\sum_{i=1}^{n}(O_i - P_i)^2\right] \quad (4.2.)$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|O_i - P_i| \quad (4.3.)$$

$$SEE = \sqrt{1 - \frac{\sum_{i=1}^{n}(O_i - P_i)^2}{n}} \quad (4.4.)$$

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. MATLAB (R2015b 64 bit) (Matlab, 2016) was utilized for designing proposed models and obtaining performance measures.

### 4.3. Performance Results of Artificial Neural Network 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 3 layers (input, hidden and output) and the input and output layers have 5 and 3 neurons respectively (Figure 4.8, Figure 4.9 and Figure 4.10). The hidden layers of FITNET and CFNN have 10 neurons and that of FFNN 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 (Moré, 1978) 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). The network parameters have been optimized by try-and-error (i.e. after testing the neural network with several different configurations and observing that these numbers yield the lowest error rates for prediction) in order to reach the accurate results. The parameter σ of GRNN model (also called 'spread' in MATLAB) determines the generalization capability of the GRNN. The best spread parameter is adopted as 2 through this work.

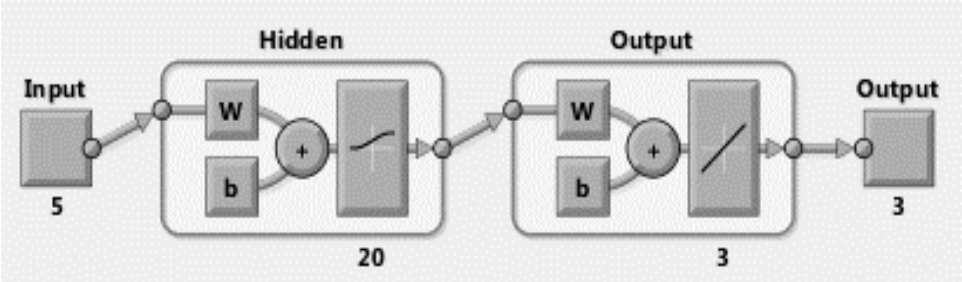



Figure 4.8. FFNN Model with 5, 20 and 3 neurons

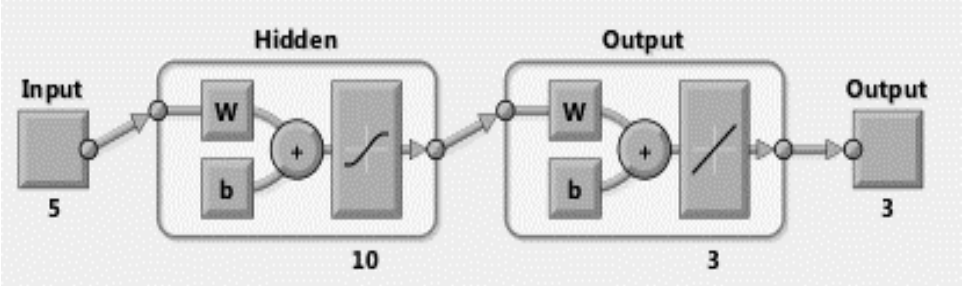



Figure 4.9. FITNET Model with 5, 10 and 3 neurons

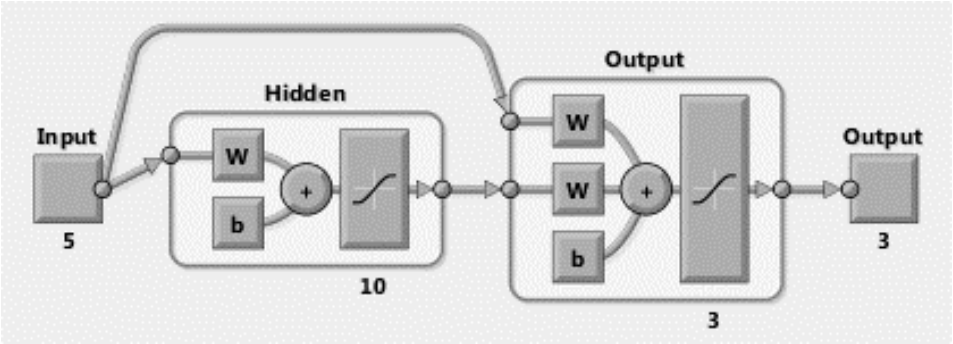



Figure 4.10. CFNN Model with 5, 10 and 3 neurons

Based on the performance results of u', v' and w' coordinates prediction using ANN prediction models with the non-cross-validated dataset (Table 4.3, Table 4.4, Table 4.5 and Figure 4.11, Figure 4.12, Figure 4.13), the following points can be made:

- The best result is obtained from w' coordinate prediction with CFNN model (MSE=5.418E-08, MAE=1.498E-04, SEE=1.000E+00, R=1.000E+00). However, all of the performance results are very close to each other for w' coordinate prediction regardless of the prediction model except GRNN.
- GRNN prediction model yielded the worst performance results for all coordinates.
- The minimum MAE, MSE and SEE results and the maximum R result are achieved from CFNN and FFNN prediction models for u' and v' coordinates prediction respectively.
- SEE and R metrics gave the same results for FFNN, FITNET and CFNN prediction models for all coordinates. Therefore, the performance of the prediction models are determined by MSE and MAE metrics.
- MSE and MAE performance of u' and v' coordinates prediction are similar for different prediction models except GRNN.

Table 4.6, Table 4.9, Table 4.12 and Table 4.15 summarized the results of u' coordinate prediction using FFNN, FITNET, CFNN and GRNN prediction models respectively. The results can be analyzed as follows:

- FFNN prediction model gave the best result at eighth fold (MSE=5.702E-06, MAE=1.733E-03, SEE=1.000E+00, R=1.000E+00).
- FITNET prediction model (MSE=8.300E-06, MAE=1.937E-03, SEE=1.000E+00, R=9.999E-01) and CFNN prediction model (MSE=7.940E-06, MAE=1.973E-03, SEE=1.000E+00, R=1.000E+00) yielded their best results which are very similar at fourth fold.
- GRNN prediction model showed the best performance results at eighth fold (MSE=7.531E-03, MAE=2.459E-01, SEE=9.591E-01, R=9.433E-01).
- FFNN prediction model is better than other prediction models and FITNET and CFNN prediction models presented early the same results in terms of MAE, SEE and R metrics.

Table 4.7, Table 4.10, Table 4.13 and Table 4.17 presented the results of v' coordinate prediction using FFNN, FITNET, CFNN and GRNN prediction models respectively. The results can be analyzed as follows:

- FFNN prediction model's best performance results (MSE=6.205E-06, MAE=1.755E-03, SEE=1.000E+00, R=1.000E+00) are better than other prediction models.
- FITNET prediction model's best results (MSE=7.605E-06, MAE=1.886E-03, SEE=1.000E+00, R=1.000E+00) and CFNN prediction model's best results (MSE=7.024E-06, MAE=1.859E-03, SEE=1.000E+00, R=1.000E+00) are very close to each other.

- The best results recorded by GRNN prediction model are worse than the others models' best results (MSE=7.423E-03, MAE=2.429E-01, SEE=9.593E-01, R=9.274E-01).

Table 4.8, Table 4.11, Table 4.14 and Table 4.17 showed the results of w' coordinate prediction using FFNN, FITNET, CFNN and GRNN prediction models respectively. The results can be interpreted as following:

- The prediction models' best results are yielded as follows: FFNN (MSE=5.848E-06, MAE=1.702E-03,SEE=1.000E+00, R=1.000E+00), FITNET (MSE=7.299E-06, MAE=1.837E-03, SEE=1.000E+00, R=1.000E+00), CFNN (MSE=7.081E-06, MAE=1.876E-03, SEE=1.000E+00, R=1.000E+00) and GRNN (MSE=7.582E-03, MAE=2.370E-01, SEE=9.582E-01, R=9.474E-01)
- FFNN prediction model presented the best results while comparing other models, FITNET and CFNN prediction models have nearly the same results and GRNN prediction model has the worst results.

The order of the models in terms of SEE and R (goodness of convergence) is the same for all coordinates except GRNN prediction model. When the evaluation is done in terms of MSE and MAE, FFNN prediction model is the best, FITNET and CFNN prediction models share the second place and GRNN prediction model is the worst.

Table 4.3. Performance results of u' coordinate prediction using ANN prediction models with the non-cross-validated dataset (The best results are outlined in bold)

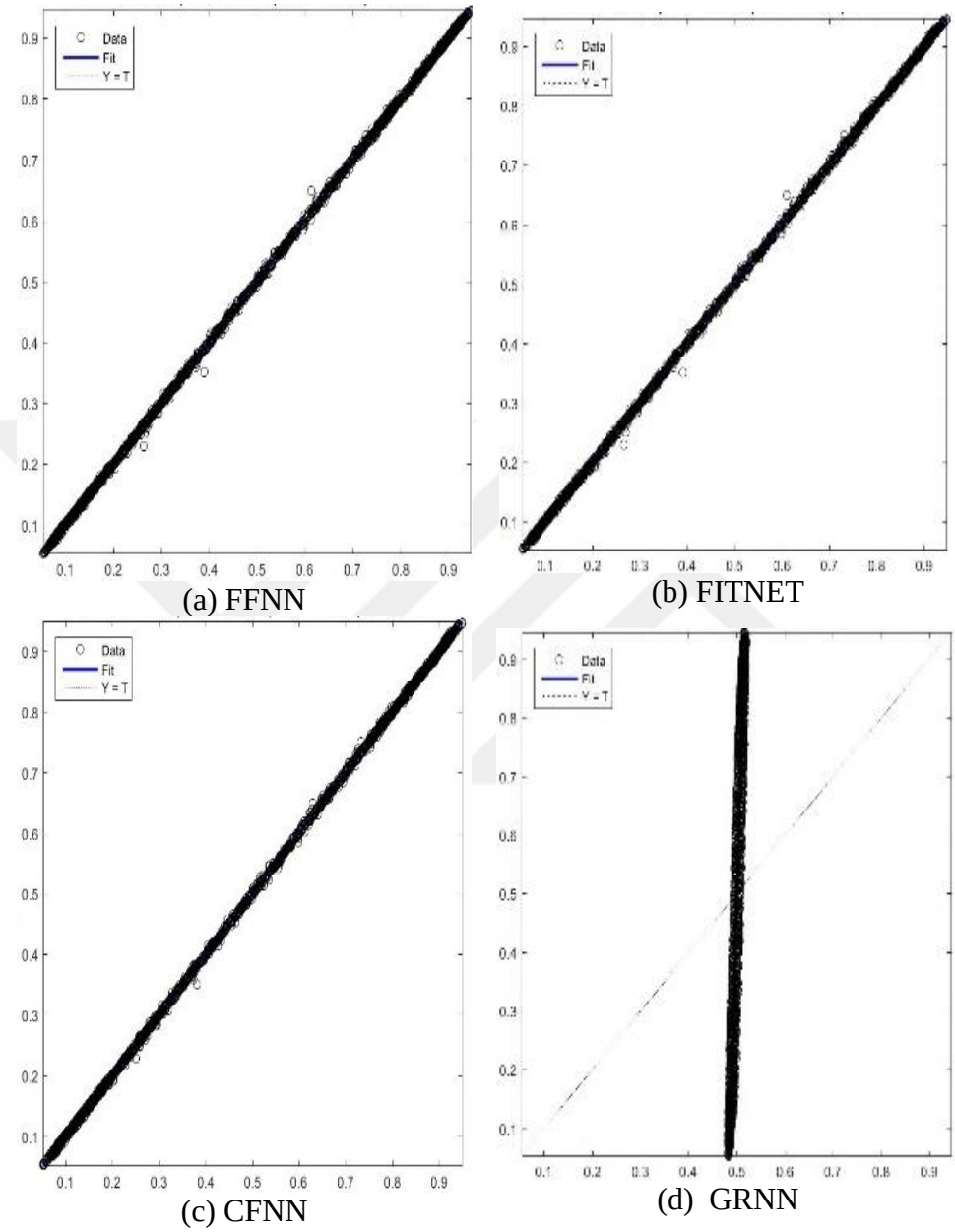
| **Models** | **Performance Metrics** | | | |
|---|---|---|---|---|
| | **MSE** | **MAE** | **SEE** | **R** |
| **FFNN** | 9.879E-06 | 2.084E-03 | 1.000E+00 | 9.999E-01 |
| **FITNET** | 1.130E-05 | 2.143E-03 | 1.000E+00 | 9.999E-01 |
| **CFNN** | **9.403E-06** | **2.025E-03** | **1.000E+00** | **9.999E-01** |
| **GRNN** | 7.911E-03 | 2.532E-01 | 1.000E+00 | 9.371E-01 |

Table 4.4. Performance results of v' coordinate prediction using ANN prediction models with the non-cross-validated dataset (The best results are outlined in bold)

| **Models** | **Performance Metrics** | | | |
|---|---|---|---|---|
| | **MSE** | **MAE** | **SEE** | **R** |
| **FFNN** | **7.307E-06** | **1.870E-03** | **1.000E+00** | **9.999E-01** |
| **FITNET** | 8.708E-06 | 1.997E-03 | 1.000E+00 | 9.999E-01 |
| **CFNN** | 8.537E-06 | 1.965E-03 | 1.000E+00 | 9.999E-01 |
| **GRNN** | 7.995E-03 | 2.554E-01 | 1.000E+00 | 9.298E-01 |

Table 4.5. Performance results of w' coordinate prediction using ANN prediction models with the non-cross-validated dataset (The best results are outlined in bold)

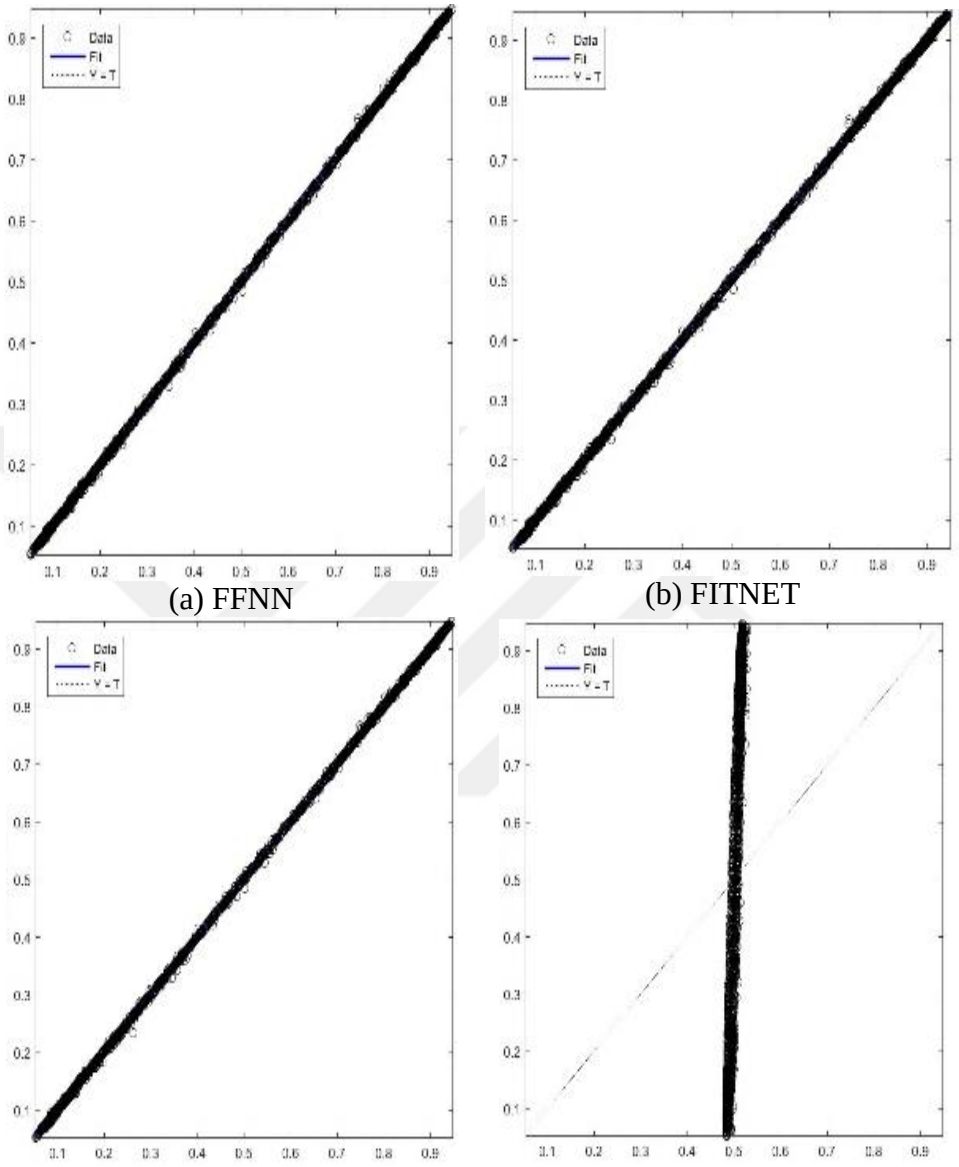
| **Models** | **Performance Metrics** | | | |
|---|---|---|---|---|
| | **MSE** | **MAE** | **SEE** | **R** |
| **FFNN** | 5.501E-08 | 1.531E-04 | 1.000E+00 | 1.000E+00 |
| **FITNET** | 5.554E-08 | 1.523E-04 | 1.000E+00 | 1.000E+00 |
| **CFNN** | **5.418E-08** | **1.498E-04** | **9.587E-01** | **1.000E+00** |
| **GRNN** | 8.088E-03 | 2.466E-01 | 1.000E+00 | 9.313E-01 |




(a) FFNN (b) FITNET

(c) CFNN (d) GRNN

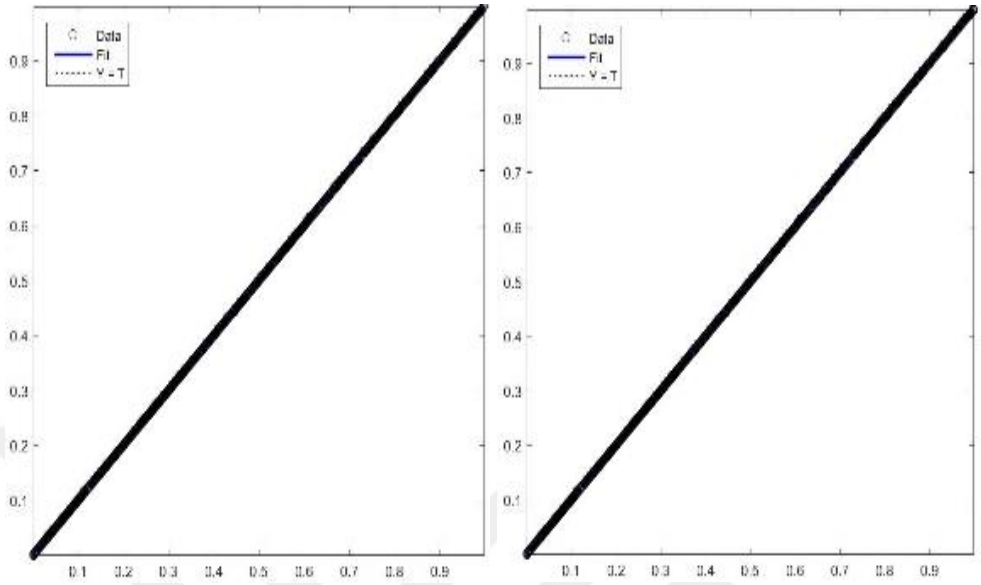
Figure 4.11. R values of u' coordinate prediction with the non-cross-validated dataset (a) FFNN (b) FITNET (c) CFNN (d) GRNN



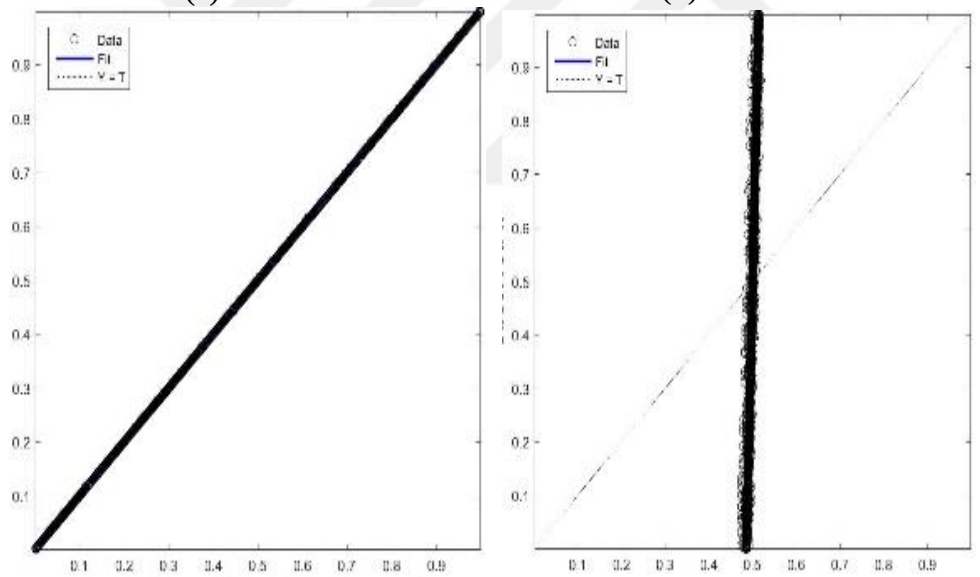

(a) FFNN (b) FITNET

(c) CFNN (d) GRNN

Figure 4.12. R values of v' coordinate prediction with the non-cross-validated dataset (a) FFNN (b) FITNET (c) CFNN (d) GRNN




(a) FFNN (b) FITNET




(c) CFNN (d) GRNN

Figure 4.13. R values of w' coordinate prediction with the non-cross-validated dataset (a) FFNN (b) FITNET (c) CFNN (d) GRNN

Table 4.6. Performance results of u' coordinate prediction using FFNN prediction model by means of 10-fold cross-validation (The best and average results are outlined in bold)

| Fold Number | Performance Metrics | | | |
|---|---|---|---|---|
| | **MSE** | **MAE** | **SEE** | **R** |
| **1** | 7.336E-06 | 1.973E-03 | 1.000E+00 | 1.000E+00 |
| **2** | 8.258E-06 | 2.029E-03 | 1.000E+00 | 1.000E+00 |
| **3** | 1.017E-05 | 2.114E-03 | 1.000E+00 | 9.999E-01 |
| **4** | 6.402E-06 | 1.776E-03 | 1.000E+00 | 1.000E+00 |
| **5** | 8.926E-06 | 2.118E-03 | 1.000E+00 | 9.999E-01 |
| **6** | 1.042E-05 | 2.106E-03 | 1.000E+00 | 9.999E-01 |
| **7** | 9.175E-06 | 2.024E-03 | 1.000E+00 | 9.999E-01 |
| **8** | **5.702E-06** | **1.733E-03** | **1.000E+00** | **1.000E+00** |
| **9** | 7.549E-06 | 1.919E-03 | 1.000E+00 | 1.000E+00 |
| **10** | 7.474E-06 | 1.807E-03 | 1.000E+00 | 1.000E+00 |
| **Average** | **8.141E-06** | **1.960E-03** | **1.000E+00** | **1.000E+00** |

Table 4.7. Performance results of v' coordinate prediction using FFNN prediction model by means of 10-fold cross-validation (The best and average results are outlined in bold)

| Fold Number | Performance Metrics | | | |
|---|---|---|---|---|
| | **MSE** | **MAE** | **SEE** | **R** |
| **1** | 7.454E-06 | 1.962E-03 | 1.000E+00 | 1.000E+00 |
| **2** | **6.205E-06** | **1.755E-03** | **1.000E+00** | **1.000E+00** |
| **3** | 7.028E-06 | 1.878E-03 | 1.000E+00 | 1.000E+00 |
| **4** | 6.441E-06 | 1.837E-03 | 1.000E+00 | 1.000E+00 |
| **5** | 7.144E-06 | 1.837E-03 | 1.000E+00 | 1.000E+00 |
| **6** | 7.803E-06 | 1.950E-03 | 1.000E+00 | 1.000E+00 |
| **7** | 7.131E-06 | 1.905E-03 | 1.000E+00 | 1.000E+00 |
| **8** | 7.500E-06 | 1.953E-03 | 1.000E+00 | 1.000E+00 |
| **9** | 6.969E-06 | 1.804E-03 | 1.000E+00 | 1.000E+00 |
| **10** | 8.070E-06 | 1.947E-03 | 1.000E+00 | 9.999E-01 |
| **Average** | **7.175E-06** | **1.883E-03** | **1.000E+00** | **1.000E+00** |

Table 4.8. Performance results of w' coordinate prediction using FFNN prediction model by means of 10-fold cross-validation (The best and average results are outlined in bold)

| Fold Number | Performance Metrics | | | |
|---|---|---|---|---|
| | **MSE** | **MAE** | **SEE** | **R** |
| **1** | 6.999E-06 | 1.873E-03 | 1.000E+00 | 1.000E+00 |
| **2** | 6.884E-06 | 1.806E-03 | 1.000E+00 | 1.000E+00 |
| **3** | 6.776E-06 | 1.886E-03 | 1.000E+00 | 1.000E+00 |
| **4** | 7.644E-06 | 1.932E-03 | 1.000E+00 | 1.000E+00 |
| **5** | 6.989E-06 | 1.853E-03 | 1.000E+00 | 1.000E+00 |
| **6** | 8.772E-06 | 2.063E-03 | 1.000E+00 | 9.999E-01 |
| **7** | **5.848E-06** | **1.702E-03** | **1.000E+00** | **1.000E+00** |
| **8** | 7.477E-06 | 1.943E-03 | 1.000E+00 | 1.000E+00 |
| **9** | 7.975E-06 | 1.907E-03 | 1.000E+00 | 1.000E+00 |
| **10** | 8.992E-06 | 2.013E-03 | 1.000E+00 | 9.999E-01 |
| **Average** | **7.436E-06** | **1.898E-03** | **1.000E+00** | **1.000E+00** |

Table 4.9. Performance results of u' coordinate prediction using FITNET prediction model by means of 10-fold cross-validation (The best and average results are outlined in bold)

| Fold Number | Performance Metrics | | | |
|---|---|---|---|---|
| | **MSE** | **MAE** | **SEE** | **R** |
| **1** | 9.267E-06 | 2.202E-03 | 1.000E+00 | 9.999E-01 |
| **2** | 9.212E-06 | 2.036E-03 | 1.000E+00 | 9.999E-01 |
| **3** | 1.045E-05 | 2.125E-03 | 1.000E+00 | 9.999E-01 |
| **4** | **8.300E-06** | **1.937E-03** | **1.000E+00** | **9.999E-01** |
| **5** | 8.926E-06 | 2.149E-03 | 1.000E+00 | 9.999E-01 |
| **6** | 1.124E-05 | 2.147E-03 | 1.000E+00 | 9.999E-01 |
| **7** | 8.961E-06 | 1.966E-03 | 1.000E+00 | 9.999E-01 |
| **8** | 9.793E-06 | 2.179E-03 | 1.000E+00 | 9.999E-01 |
| **9** | 8.741E-06 | 1.986E-03 | 1.000E+00 | 9.999E-01 |
| **10** | 1.033E-05 | 2.085E-03 | 1.000E+00 | 9.999E-01 |
| **Average** | **9.522E-06** | **2.081E-03** | **1.000E+00** | **9.999E-01** |

Table 4.10. Performance results of v' coordinate prediction using FITNET prediction model by means of 10-fold cross-validation (The best and average results are outlined in bold)

| Fold Number | Performance Metrics | | | |
|---|---|---|---|---|
| | MSE | MAE | SEE | R |
| **1** | 8.104E-06 | 1.929E-03 | 1.000E+00 | 1.000E+00 |
| **2** | 8.362E-06 | 1.918E-03 | 1.000E+00 | 9.999E-01 |
| **3** | 7.821E-06 | 1.995E-03 | 1.000E+00 | 1.000E+00 |
| **4** | **7.605E-06** | **1.886E-03** | **1.000E+00** | **1.000E+00** |
| **5** | 8.518E-06 | 1.975E-03 | 1.000E+00 | 9.999E-01 |
| **6** | 9.879E-06 | 2.037E-03 | 1.000E+00 | 9.999E-01 |
| **7** | 8.189E-06 | 1.948E-03 | 1.000E+00 | 1.000E+00 |
| **8** | 7.727E-06 | 1.968E-03 | 1.000E+00 | 1.000E+00 |
| **9** | 8.509E-06 | 1.923E-03 | 1.000E+00 | 9.999E-01 |
| **10** | 9.361E-06 | 1.963E-03 | 1.000E+00 | 9.999E-01 |
| **Average** | **8.408E-06** | **1.954E-03** | **1.000E+00** | **9.999E-01** |

Table 4.11. Performance results of w' coordinate prediction using FITNET prediction model by means of 10-fold cross-validation (The best and average results are outlined in bold)

| Fold Number | Performance Metrics | | | |
|---|---|---|---|---|
| | MSE | MAE | SEE | R |
| **1** | 7.804E-06 | 1.900E-03 | 1.000E+00 | 1.000E+00 |
| **2** | 7.708E-06 | 1.884E-03 | 1.000E+00 | 1.000E+00 |
| **3** | 8.320E-06 | 2.026E-03 | 1.000E+00 | 1.000E+00 |
| **4** | **7.299E-06** | **1.837E-03** | **1.000E+00** | **1.000E+00** |
| **5** | 8.930E-06 | 1.981E-03 | 1.000E+00 | 9.999E-01 |
| **6** | 9.649E-06 | 2.069E-03 | 1.000E+00 | 9.999E-01 |
| **7** | 7.589E-06 | 1.966E-03 | 1.000E+00 | 1.000E+00 |
| **8** | 8.468E-06 | 1.960E-03 | 1.000E+00 | 9.999E-01 |
| **9** | 9.414E-06 | 1.961E-03 | 1.000E+00 | 9.999E-01 |
| **10** | 9.153E-06 | 1.979E-03 | 1.000E+00 | 9.999E-01 |
| **Average** | **8.434E-06** | **1.956E-03** | **1.000E+00** | **9.999E-01** |

Table 4.12. Performance results of u' coordinate prediction using CFNN prediction model by means of 10-fold cross-validation (The best and average results are outlined in bold)

| Fold Number | Performance Metrics | | | |
|---|---|---|---|---|
| | **MSE** | **MAE** | **SEE** | **R** |
| **1** | 9.307E-06 | 2.180E-03 | 1.000E+00 | 9.999E-01 |
| **2** | 8.546E-06 | 2.035E-03 | 1.000E+00 | 1.000E+00 |
| **3** | 8.978E-06 | 2.088E-03 | 1.000E+00 | 9.999E-01 |
| **4** | **7.940E-06** | **1.973E-03** | **1.000E+00** | **1.000E+00** |
| **5** | 9.732E-06 | 2.154E-03 | 1.000E+00 | 9.999E-01 |
| **6** | 1.180E-05 | 2.216E-03 | 1.000E+00 | 9.999E-01 |
| **7** | 1.007E-05 | 2.055E-03 | 1.000E+00 | 9.999E-01 |
| **8** | 8.527E-06 | 2.116E-03 | 1.000E+00 | 9.999E-01 |
| **9** | 8.417E-06 | 2.022E-03 | 1.000E+00 | 9.999E-01 |
| **10** | 9.055E-06 | 2.095E-03 | 1.000E+00 | 9.999E-01 |
| **Average** | **9.237E-06** | **2.093E-03** | **1.000E+00** | **9.999E-01** |

Table 4.13. Performance results of v' coordinate prediction using CFNN prediction model by means of 10-fold cross-validation (The best and average results are outlined in bold)

| Fold Number | Performance Metrics | | | |
|---|---|---|---|---|
| | **MSE** | **MAE** | **SEE** | **R** |
| **1** | 7.860E-06 | 1.976E-03 | 1.000E+00 | 1.000E+00 |
| **2** | 7.418E-06 | 1.895E-03 | 1.000E+00 | 1.000E+00 |
| **3** | 7.902E-06 | 2.001E-03 | 1.000E+00 | 1.000E+00 |
| **4** | **7.024E-06** | **1.859E-03** | **1.000E+00** | **1.000E+00** |
| **5** | 7.984E-06 | 1.958E-03 | 1.000E+00 | 1.000E+00 |
| **6** | 8.574E-06 | 1.982E-03 | 1.000E+00 | 9.999E-01 |
| **7** | 7.110E-06 | 1.926E-03 | 1.000E+00 | 1.000E+00 |
| **8** | 7.629E-06 | 1.904E-03 | 1.000E+00 | 1.000E+00 |
| **9** | 7.854E-06 | 1.938E-03 | 1.000E+00 | 1.000E+00 |
| **10** | 8.616E-06 | 1.957E-03 | 1.000E+00 | 9.999E-01 |
| **Average** | **7.797E-06** | **1.940E-03** | **1.000E+00** | **1.000E+00** |

Table 4.14. Performance results of w' coordinate prediction using CFNN prediction model by means of 10-fold cross-validation (The best and average results are outlined in bold)

| Fold Number | Performance Metrics | | | |
|---|---|---|---|---|
| | **MSE** | **MAE** | **SEE** | **R** |
| **1** | 7.367E-06 | 1.919E-03 | 1.000E+00 | 1.000E+00 |
| **2** | 7.811E-06 | 1.906E-03 | 1.000E+00 | 1.000E+00 |
| **3** | 8.962E-06 | 2.120E-03 | 1.000E+00 | 9.999E-01 |
| **4** | **7.081E-06** | **1.876E-03** | **1.000E+00** | **1.000E+00** |
| **5** | 7.951E-06 | 1.987E-03 | 1.000E+00 | 1.000E+00 |
| **6** | 8.504E-06 | 2.008E-03 | 1.000E+00 | 9.999E-01 |
| **7** | 7.335E-06 | 1.971E-03 | 1.000E+00 | 1.000E+00 |
| **8** | 7.733E-06 | 1.929E-03 | 1.000E+00 | 1.000E+00 |
| **9** | 7.152E-06 | 1.932E-03 | 1.000E+00 | 1.000E+00 |
| **10** | 9.517E-06 | 2.052E-03 | 1.000E+00 | 9.999E-01 |
| **Average** | **7.941E-06** | **1.970E-03** | **1.000E+00** | **1.000E+00** |

Table 4.15. Performance results of u' coordinate prediction using GRNN prediction model by means of 10-fold cross-validation (The best and average results are outlined in bold)

| Fold Number | Performance Metrics | | | |
|---|---|---|---|---|
| | **MSE** | **MAE** | **SEE** | **R** |
| **1** | 7.740E-03 | 2.489E-01 | 9.605E-01 | 9.357E-01 |
| **2** | 8.006E-03 | 2.530E-01 | 9.616E-01 | 9.357E-01 |
| **3** | 7.624E-03 | 2.475E-01 | 9.611E-01 | 9.377E-01 |
| **4** | 7.638E-03 | 2.466E-01 | 9.611E-01 | 9.291E-01 |
| **5** | 7.960E-03 | 2.540E-01 | 9.594E-01 | 9.358E-01 |
| **6** | 7.830E-03 | 2.513E-01 | 9.601E-01 | 9.376E-01 |
| **7** | 7.951E-03 | 2.539E-01 | 9.594E-01 | 9.422E-01 |
| **8** | **7.531E-03** | **2.459E-01** | **9.591E-01** | **9.433E-01** |
| **9** | 7.622E-03 | 2.466E-01 | 9.611E-01 | 9.384E-01 |
| **10** | 7.918E-03 | 2.534E-01 | 9.596E-01 | 9.366E-01 |
| **Average** | **7.782E-03** | **2.501E-01** | **9.603E-01** | **9.372E-01** |

Table 4.16. Performance results of v' coordinate prediction using GRNN prediction model by means of 10-fold cross-validation (The best and average results are outlined in bold)

| Fold Number | Performance Metrics | | | |
|---|---|---|---|---|
| | MSE | MAE | SEE | R |
| **1** | 7.783E-03 | 2.505E-01 | 9.603E-01 | 9.262E-01 |
| **2** | 7.587E-03 | 2.466E-01 | 9.613E-01 | 9.201E-01 |
| **3** | 7.975E-03 | 2.536E-01 | 9.622E-01 | 9.187E-01 |
| **4** | 7.942E-03 | 2.544E-01 | 9.595E-01 | 9.122E-01 |
| **5** | 7.641E-03 | 2.473E-01 | 9.610E-01 | 9.258E-01 |
| **6** | 7.568E-03 | 2.458E-01 | 9.614E-01 | 9.155E-01 |
| **7** | 7.782E-03 | 2.484E-01 | 9.603E-01 | 9.260E-01 |
| **8** | 7.781E-03 | 2.500E-01 | 9.603E-01 | 9.268E-01 |
| **9** | 7.856E-03 | 2.522E-01 | 9.599E-01 | 9.124E-01 |
| **10** | **7.423E-03** | **2.429E-01** | **9.593E-01** | **9.274E-01** |
| **Average** | **7.734E-03** | **2.492E-01** | **9.606E-01** | **9.209E-01** |

Table 4.17. Performance results of w' coordinate prediction using GRNN prediction model by means of 10-fold cross-validation (The best and average results are outlined in bold)

| Fold Number | Performance Metrics | | | |
|---|---|---|---|---|
| | MSE | MAE | SEE | R |
| **1** | **7.582E-03** | **2.370E-01** | **9.582E-01** | **9.474E-01** |
| **2** | 8.184E-03 | 2.469E-01 | 9.612E-01 | 9.335E-01 |
| **3** | 7.585E-03 | 2.371E-01 | 9.613E-01 | 9.279E-01 |
| **4** | 8.011E-03 | 2.453E-01 | 9.591E-01 | 9.006E-01 |
| **5** | 7.842E-03 | 2.409E-01 | 9.600E-01 | 9.063E-01 |
| **6** | 7.883E-03 | 2.421E-01 | 9.598E-01 | 9.465E-01 |
| **7** | 8.179E-03 | 2.515E-01 | 9.583E-01 | 9.040E-01 |
| **8** | 7.750E-03 | 2.404E-01 | 9.605E-01 | 9.447E-01 |
| **9** | 7.843E-03 | 2.419E-01 | 9.600E-01 | 8.988E-01 |
| **10** | 7.744E-03 | 2.404E-01 | 9.605E-01 | 9.214E-01 |
| Average | **7.860E-03** | **2.424E-01** | **9.599E-01** | **9.231E-01** |

### 4.4. Performance Results of Support Vector Regression Prediction Model

SVR prediction model was designed by running "fitrsvm" function in MATLAB. Several combinations have been tried and Radial Basis Function was chosen as the kernel for performance comparison. The results in Table 4.18 show

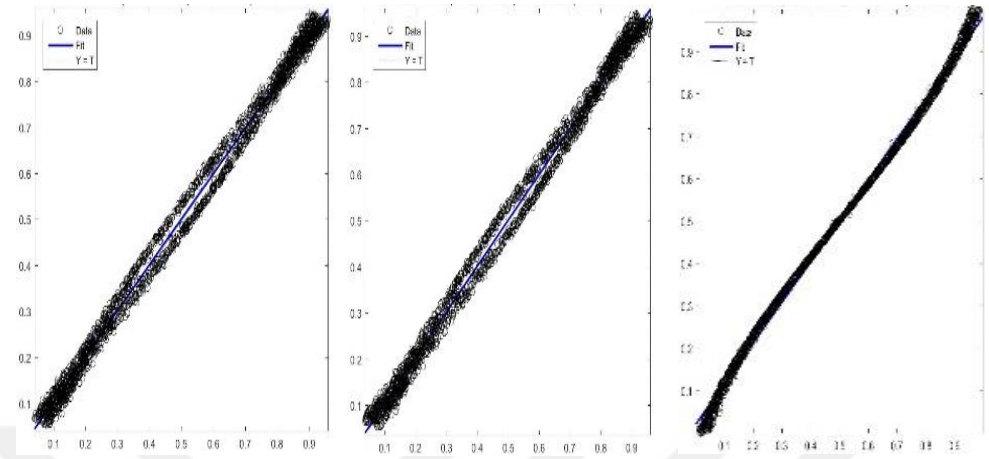
MSE, MAE, SEE and R values of u', v' and w' coordinates prediction using SVR prediction model with the non-cross-validated dataset. Figure 4.14 depicts R values of u', v' and w' coordinates prediction with the non-cross-validated dataset. Based on the results obtained from Table 4.18 and Figure 4.14, the following comments can be made:

- The lowest values of MSE, MAE and SEE are taken from w' coordinate prediction as 4.321E-04, 1.832E-02 and 9.997E-01, respectively. Also, the best fitting curve of R which is shown in Figure 4.14.c belongs to w' coordinate prediction.
- Both SEE and R values are very close to each other.
- MSE and MAE values of v' coordinate prediction show the second best performance for prediction.

Table 4.19 through Table 4.21 show the performance of SVR prediction model for u', v' and w' coordinates with 10-fold cross validation. As is seen from the tables, the performance of w' coordinate prediction is better than u' and v' coordinates prediction in terms of MSE and MAE metrics. For SEE and R metrics, the values of w' coordinate prediction are better or approximately equal to the performance of other coordinates.

Table 4.18. Performance results of u', v' and w' coordinates prediction using SVR prediction model with the non-cross-validated dataset (The best results are outlined in bold)

| **Performance Metrics** | **Coordinates** | | |
|---|---|---|---|
| | **u'** | **v'** | **w'** |
| **MSE** | 5.102E-04 | 4.985E-04 | **4.321E-04** |
| **MAE** | 1.957E-02 | 1.940E-02 | **1.832E-02** |
| **SEE** | 9.998E-01 | 9.998E-01 | **9.997E-01** |
| **R** | 9.970E-01 | 9.971E-01 | **9.983E-01** |



**(a)** **(b)** **(c)**

Figure 4.14. R values of u', v' and w' coordinates prediction with the non-cross-validated dataset using SVR prediction model (a) u' coordinate (b) v' coordinate (c) w' coordinate

Table 4.19. The performance results of u' coordinate prediction with 10-fold cross validation using SVR prediction model (The best and average results are outlined in bold)

| Fold Number | Performance Metrics | | | |
|---|---|---|---|---|
| | **MSE** | **MAE** | **SEE** | **R** |
| **1** | 5.306E-04 | 1.987E-02 | 9.997E-01 | 9.970E-01 |
| **2** | 5.122E-04 | 1.954E-02 | 9.997E-01 | 9.970E-01 |
| **3** | 5.009E-04 | 1.925E-02 | 9.997E-01 | 9.971E-01 |
| **4** | 5.107E-04 | 1.948E-02 | 9.998E-01 | 9.970E-01 |
| **5** | **4.839E-04** | **1.879E-02** | **9.997E-01** | **9.972E-01** |
| **6** | 5.600E-04 | 2.057E-02 | 9.997E-01 | 9.967E-01 |
| **7** | 5.333E-04 | 2.013E-02 | 9.997E-01 | 9.968E-01 |
| **8** | 5.204E-04 | 1.979E-02 | 9.997E-01 | 9.969E-01 |
| **9** | 4.910E-04 | 1.901E-02 | 9.998E-01 | 9.972E-01 |
| **10** | 5.308E-04 | 1.997E-02 | 9.997E-01 | 9.969E-01 |
| **Average** | **5.174E-04** | **1.964E-02** | **9.997E-01** | **9.970E-01** |

Table 4.20. The performance results of v' coordinate prediction with 10-fold cross validation using SVR prediction model (The best and average results are outlined in bold)

| Fold Number | Performance Metrics | | | |
|---|---|---|---|---|
| | **MSE** | **MAE** | **SEE** | **R** |
| **1** | 5.185E-04 | 1.984E-02 | 9.997E-01 | 9.970E-01 |
| **2** | 5.200E-04 | 1.987E-02 | 9.997E-01 | 9.969E-01 |
| **3** | 4.914E-04 | 1.923E-02 | 9.998E-01 | 9.971E-01 |
| **4** | 5.065E-04 | 1.944E-02 | 9.997E-01 | 9.970E-01 |
| **5** | 4.851E-04 | 1.892E-02 | 9.998E-01 | 9.972E-01 |
| **6** | 4.976E-04 | 1.935E-02 | 9.998E-01 | 9.971E-01 |
| **7** | **4.767E-04** | **1.890E-02** | **9.997E-01** | **9.972E-01** |
| **8** | 4.853E-04 | 1.910E-02 | 9.998E-01 | 9.972E-01 |
| **9** | 5.214E-04 | 1.977E-02 | 9.997E-01 | 9.969E-01 |
| **10** | 5.172E-04 | 1.985E-02 | 9.997E-01 | 9.969E-01 |
| **Average** | **5.020E-04** | **1.943E-02** | **9.997E-01** | **9.971E-01** |

Table 4.21. The performance results of w' coordinate prediction with 10-fold cross validation using SVR prediction model (The best and average results are outlined in bold)

| Fold Number | Performance Metrics | | | |
|---|---|---|---|---|
| | **MSE** | **MAE** | **SEE** | **R** |
| **1** | 4.630E-04 | 1.902E-02 | 9.998E-01 | 9.980E-01 |
| **2** | 4.423E-04 | 1.831E-02 | 9.998E-01 | 9.980E-01 |
| **3** | 4.474E-04 | 1.864E-02 | 9.998E-01 | 9.980E-01 |
| **4** | 4.696E-04 | 1.916E-02 | 9.998E-01 | 9.980E-01 |
| **5** | 4.558E-04 | 1.881E-02 | 9.998E-01 | 9.980E-01 |
| **6** | 4.789E-04 | 1.934E-02 | 9.998E-01 | 9.980E-01 |
| **7** | 4.604E-04 | 1.895E-02 | 9.998E-01 | 9.981E-01 |
| **8** | 4.598E-04 | 1.878E-02 | 9.998E-01 | 9.980E-01 |
| **9** | **4.282E-04** | **1.810E-02** | **9.998E-01** | **9.982E-01** |
| **10** | 4.637E-04 | 1.906E-02 | 9.998E-01 | 9.981E-01 |
| **Average** | **4.569E-04** | **1.882E-02** | **9.998E-01** | **9.980E-01** |

### 4.5. Performance Results of Classification and Regression Tree Prediction Model

CART is one of the most powerful data mining and machine learning methods for constructing prediction models from data (Xu et al., 2013). The leaf nodes in the tree structure represent a corresponding linear regression model with fully trained coefficients and the branches represent the conjunctions of the features that lead to those target variables for the issue of atomic coordinate prediction (Figure 4.15).

CART analysis was performed using "classregtree" function which specifies some optional parameters in MATLAB. The "prune" option computes the full tree and prunes the subtrees. The "minparent" option splits impure nodes which have k or more observations to be split for given number of k. The "qetoler" option defines tolerance on quadratic error per node for CART. Splitting nodes stops when quadratic error per node drops below qetoler*qed, where qed is the quadratic error for the entire data computed before the decision tree is grown. The "mergeleaves" option merges leaves that originate from the same parent node and give the sum of risk values greater or equal to the risk associated with the parent node. The values of the parameters were chosen as prune=on, minparent=50 (default is 10), qetoler=1E-7 (default is 1E-6), mergeleaves=on. All of the parameter values were decided by trial-and-error.

Table 4.22 shows MSE, MAE, SEE and R results of CART prediction model with the non-cross-validated dataset. R charts of u', v' and w' coordinates prediction can be seen in Figure 4.16. Like other prediction models, the lowest MSE and MAE values are obtained from w' prediction. SEE and R values are nearly the same for all coordinates.

Table 4.23, Table 4.24 and Table 4.25 present the performance results of u', v' and w' coordinates prediction with 10-fold cross validation. The best result is

obtained from w' coordinate prediction at eighth fold (MSE=1.683E-05, MAE=3.482E-03, SEE=1.000E+00, R=9.999E-01).

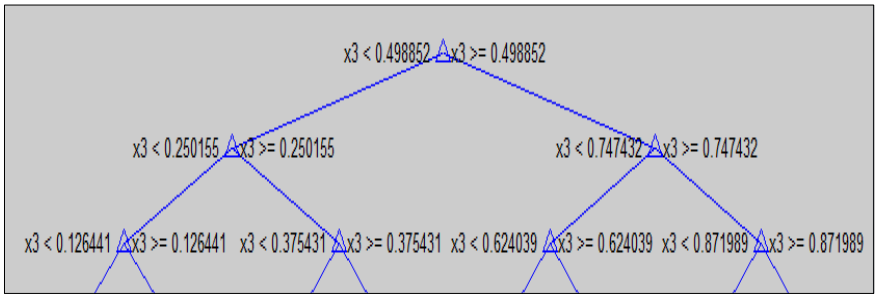
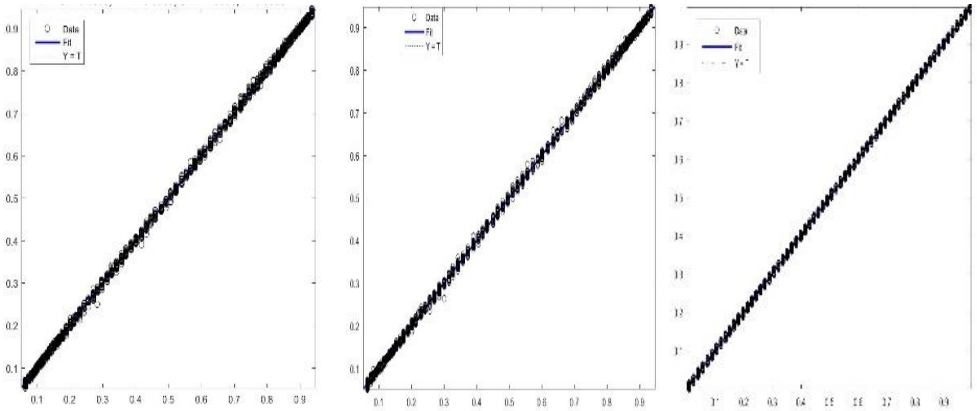



Figure 4.15. A sample CART prediction model for atomic coordinate prediction

Table 4.22. Performance results of u', v' and w' coordinates prediction using CART prediction model with the non-cross-validated dataset (The best results are outlined in bold)

| **Performance Metrics** | **Coordinates** | | |
|---|---|---|---|
| | **u'** | **v'** | **w'** |
| **MSE** | 2.471E-05 | 2.484E-05 | **2.140E-05** |
| **MAE** | 3.868E-03 | 3.835E-03 | **3.021E-03** |
| **SEE** | 1.000E+00 | 1.000E+00 | **1.000E+00** |
| **R** | 9.999E-01 | 9.998E-01 | **9.999E-01** |



**(a)** **(b)** **(c)**

Figure 4.16. R values of u', v' and w' coordinates prediction with the non-cross-validated dataset using CART prediction model (a) u' coordinate (b) v' coordinate (c) w' coordinate

Table 4.23. The performance results of u' coordinate prediction with 10-fold cross validation using CART prediction model (The best and average results are outlined in bold)

| Fold Number | Performance Metrics | | | |
|---|---|---|---|---|
| | **MSE** | **MAE** | **SEE** | **R** |
| **1** | 2.558E-05 | 3.717E-03 | 1.000E+00 | 9.998E-01 |
| **2** | 2.416E-05 | 3.765E-03 | 1.000E+00 | 9.999E-01 |
| **3** | **2.398E-05** | **3.674E-03** | **1.000E+00** | **9.999E-01** |
| **4** | 2.511E-05 | 3.830E-03 | 1.000E+00 | 9.999E-01 |
| **5** | 2.475E-05 | 3.862E-03 | 1.000E+00 | 9.999E-01 |
| **6** | 2.487E-05 | 3.797E-03 | 1.000E+00 | 9.999E-01 |
| **7** | 2.433E-05 | 3.739E-03 | 1.000E+00 | 9.999E-01 |
| **8** | 2.707E-05 | 3.835E-03 | 1.000E+00 | 9.998E-01 |
| **9** | 2.628E-05 | 3.905E-03 | 1.000E+00 | 9.998E-01 |
| **10** | 2.413E-05 | 3.860E-03 | 1.000E+00 | 9.999E-01 |
| **Average** | **2.503E-05** | **3.798E-03** | **1.000E+00** | **9.999E-01** |

Table 4.24. The performance results of v' coordinate prediction with 10-fold cross validation using CART prediction model (The best and average results are outlined in bold)

| Fold Number | Performance Metrics | | | |
|---|---|---|---|---|
| | **MSE** | **MAE** | **SEE** | **R** |
| **1** | 2.386E-05 | 3.639E-03 | 1.000E+00 | 9.999E-01 |
| **2** | 2.263E-05 | 3.720E-03 | 1.000E+00 | 9.999E-01 |
| **3** | 2.393E-05 | 3.831E-03 | 1.000E+00 | 9.999E-01 |
| **4** | 2.570E-05 | 3.864E-03 | 1.000E+00 | 9.998E-01 |
| **5** | 2.272E-05 | 3.661E-03 | 1.000E+00 | 9.999E-01 |
| **6** | 2.275E-05 | 3.692E-03 | 1.000E+00 | 9.999E-01 |
| **7** | 2.335E-05 | 3.703E-03 | 1.000E+00 | 9.999E-01 |
| **8** | **2.207E-05** | **3.582E-03** | **1.000E+00** | **9.999E-01** |
| **9** | 2.366E-05 | 3.690E-03 | 1.000E+00 | 9.999E-01 |
| **10** | 2.467E-05 | 3.768E-03 | 1.000E+00 | 9.999E-01 |
| **Average** | **2.353E-05** | **3.715E-03** | **1.000E+00** | **9.999E-01** |

Table 4.25. The performance results of w' coordinate prediction with 10-fold cross validation using CART prediction model (The best and average results are outlined in bold)

| Fold Number | Performance Metrics | | | |
|---|---|---|---|---|
| | **MSE** | **MAE** | **SEE** | **R** |
| **1** | 1.788E-05 | 3.590E-03 | 1.000E+00 | 9.999E-01 |
| **2** | 1.925E-05 | 3.752E-03 | 1.000E+00 | 9.999E-01 |
| **3** | 1.721E-05 | 3.537E-03 | 1.000E+00 | 9.999E-01 |
| **4** | 1.868E-05 | 3.721E-03 | 1.000E+00 | 9.999E-01 |
| **5** | 1.894E-05 | 3.732E-03 | 1.000E+00 | 9.999E-01 |
| **6** | 1.759E-05 | 3.579E-03 | 1.000E+00 | 9.999E-01 |
| **7** | 1.759E-05 | 3.570E-03 | 1.000E+00 | 9.999E-01 |
| **8** | **1.683E-05** | **3.482E-03** | **1.000E+00** | **9.999E-01** |
| **9** | 1.958E-05 | 3.801E-03 | 1.000E+00 | 9.999E-01 |
| **10** | 1.912E-05 | 3.724E-03 | 1.000E+00 | 9.999E-01 |
| **Average** | **1.827E-05** | **3.649E-03** | **1.000E+00** | **9.999E-01** |

### 4.6. Performance Results of Multiple Linear Regression Prediction Model

MLR prediction model is designed with the purpose of comparing Machine Learning methods with a statistical regression method. The model is coded in Statistics and Machine Learning Toolbox of MATLAB using "fitlm" function. Model specification is specified as "linear" and model specification equations are set for each coordinate such as 'u' ~ u+v+w+m+n' is set for u' coordinate prediction; 'v' ~ u+v+w+m+n' is set for v' coordinate prediction; and 'w' ~ u+v+w+m+n' is set for w' coordinate prediction.

Table 4.26 through Table 4.29 show the performance of MLR prediction model with or without using 10-fold cross validation. The best attainable values of MSE, MAE, SEE and R from MLR prediction model are 4.110E-02 for MSE, 1.394E-01 for MAE, 3.000E+00 for SEE and 6.000E-01 for R.

Table 4.26. Performance results of u', v' and w' coordinates prediction using MLR prediction model with the non-cross-validated dataset (The best results are outlined in bold)

| **Performance Metrics** | **Coordinates** | | |
|---|---|---|---|
| | **u'** | **v'** | **w'** |
| **MSE** | 1.525E-01 | 1.298E-01 | **4.865E-02** |
| **MAE** | 2.482E+00 | 2.300E+00 | **1.476E-01** |
| **SEE** | 3.000E+00 | 3.000E+00 | **3.000E+00** |
| **R** | 5.999E-01 | 6.000E-01 | **6.000E-01** |

Table 4.27. The performance results of u' coordinate prediction with 10-fold cross validation using MLR prediction model (The best and average results are outlined in bold)

| **Fold Number** | **Performance Metrics** | | | |
|---|---|---|---|---|
| | **MSE** | **MAE** | **SEE** | **R** |
| **1** | 1.411E-01 | 2.516E+00 | 3.000E+00 | 5.999E-01 |
| **2** | 1.416E-01 | 2.462E+00 | 3.000E+00 | 5.999E-01 |
| **3** | 1.192E-01 | 2.446E+00 | 3.000E+00 | 6.000E-01 |
| **4** | 1.270E-01 | 2.431E+00 | 3.000E+00 | 6.000E-01 |
| **5** | 1.852E-01 | 2.599E+00 | 3.000E+00 | 5.999E-01 |
| **6** | **1.099E-01** | **2.307E+00** | **3.000E+00** | **6.000E-01** |
| **7** | 1.418E-01 | 2.485E+00 | 3.000E+00 | 5.999E-01 |
| **8** | 1.335E-01 | 2.468E+00 | 3.000E+00 | 6.000E-01 |
| **9** | 1.369E-01 | 2.488E+00 | 3.000E+00 | 6.000E-01 |
| **10** | 1.573E-01 | 2.508E+00 | 3.000E+00 | 5.999E-01 |
| **Average** | **1.393E-01** | **2.471E+00** | **3.000E+00** | **5.999E-01** |

Table 4.28. The performance results of v' coordinate prediction with 10-fold cross validation using MLR prediction model (The best and average results are outlined in bold)

| Fold Number | Performance Metrics | | | |
|---|---|---|---|---|
| | MSE | MAE | SEE | R |
| **1** | 1.420E-01 | 2.458E+00 | 3.000E+00 | 5.999E-01 |
| **2** | 1.406E-01 | 2.389E+00 | 3.000E+00 | 5.999E-01 |
| **3** | 1.379E-01 | 2.350E+00 | 3.000E+00 | 5.999E-01 |
| **4** | **1.114E-01** | **2.195E+00** | **3.000E+00** | **6.000E-01** |
| **5** | 1.731E-01 | 2.516E+00 | 3.000E+00 | 5.999E-01 |
| **6** | 1.255E-01 | 2.391E+00 | 3.000E+00 | 6.000E-01 |
| **7** | 1.133E-01 | 2.292E+00 | 3.000E+00 | 6.000E-01 |
| **8** | 1.229E-01 | 2.335E+00 | 3.000E+00 | 6.000E-01 |
| **9** | 1.371E-01 | 2.356E+00 | 3.000E+00 | 6.000E-01 |
| **10** | 1.453E-01 | 2.394E+00 | 3.000E+00 | 5.999E-01 |
| **Average** | **1.349E-01** | **2.368E+00** | **3.000E+00** | **6.000E-01** |

Table 4.29. The performance results of w' coordinate prediction with 10-fold cross validation using MLR prediction model (The best and average results are outlined in bold)

| Fold Number | Performance Metrics | | | |
|---|---|---|---|---|
| | MSE | MAE | SEE | R |
| **1** | 4.710E-02 | 1.487E-01 | 3.000E+00 | 6.000E-01 |
| **2** | 4.497E-02 | 1.416E-01 | 3.000E+00 | 6.000E-01 |
| **3** | 4.428E-02 | 1.467E-01 | 3.000E+00 | 6.000E-01 |
| **4** | 4.414E-02 | 1.450E-01 | 3.000E+00 | 6.000E-01 |
| **5** | 4.671E-02 | 1.470E-01 | 3.000E+00 | 6.000E-01 |
| **6** | 4.279E-02 | 1.400E-01 | 3.000E+00 | 6.000E-01 |
| **7** | 5.668E-02 | 1.482E-01 | 3.000E+00 | 6.000E-01 |
| **8** | 4.288E-02 | 1.406E-01 | 3.000E+00 | 6.000E-01 |
| **9** | **4.110E-02** | **1.394E-01** | **3.000E+00** | **6.000E-01** |
| **10** | 5.347E-02 | 1.500E-01 | 3.000E+00 | 6.000E-01 |
| **Average** | **4.641E-02** | **1.447E-01** | **3.000E+00** | **6.000E-01** |

### 4.7. Overall Comparison of the Prediction Models

In this section of the thesis, the performance values of the prediction models are compared with each other according to atomic coordinates. The summary of the detailed results are given in Table 4.30 through Table 4.38.

All prediction models performed good levels of successes when we look at all results with a general view. However, the results of FFNN, FITNET and CFNN prediction models have a superiority over other prediction models. This implies that ANN based models can closely estimate atomic coordinates of CNTs. CART and SVR prediction models produced average performance results comparing to other prediction models.

The training times for FFNN, FITNET and CFNN prediction models are longer than other prediction models. Since there is no training phase in GRNN and MLR, the GRNN and MLR prediction models produced results much faster than the other prediction models however they cannot yielded better results than other prediction models.

When the cross-validation is not considered, the following inferences can be made from Table 4.30 through Table 4.32:

- CFNN prediction model has the best performance values for u' and w' coordinates prediction.
- For v' coordinate prediction, FFNN prediction model is the best.
- The lowest values of MSE and MAE are obtained from w' coordinate prediction.
- MLR prediction model performs with high errors than others and is not a good predictor for this problem.
- SEE and R results of prediction models are very similar regardless of coordinate except GRNN and MLR prediction models.
- The order of performance of the prediction models except FFNN, FITNET and CFNN is CART, SVR, GRNN and MLR respectively.

When the prediction models are evaluated by 10-fold cross-validation, Table 4.33 through Table 4.35 summaries the best results whereas Table 4.36 through

Table 4.38 summaries the average results. Based on the results obtained in Table 4.33 through Table 4.38 the following comments can be made :

- FFNN prediction model recorded the best values of MSE for u' and MAE for w' coordinates prediction with 10-fold cross-validated dataset.
- When the average values are considered, FFNN prediction model performs best performance values again. FITNET and CFNN prediction models follow it with a small margin.
- FFNN, FITNET, CFNN, SVR and CART prediction models show relative typical results when assessing SEE and R. Therefore, the performance of these models are evaluated by means of MSE and MAE.
- MLR prediction model is not suitable for the use of atomic coordinate prediction.
- All prediction models show compatible results to the models which tested with non-cross-validated dataset.

Table 4.30. Overall comparison of the u' coordinate prediction models with non-cross validated dataset (The best results are outlined in bold)

| **Models** | **Performance Metrics** | | | |
|---|---|---|---|---|
| | **MSE** | **MAE** | **SEE** | **R** |
| **FFNN** | 9.879E-06 | 2.084E-03 | 1.000E+00 | 9.999E-01 |
| **FITNET** | 1.130E-05 | 2.143E-03 | 1.000E+00 | 9.999E-01 |
| **CFNN** | **9.403E-06** | **2.025E-03** | **1.000E+00** | **9.999E-01** |
| **GRNN** | 7.911E-03 | 2.532E-01 | 1.000E+00 | 9.371E-01 |
| **SVR** | 5.102E-04 | 1.957E-02 | 9.998E-01 | 9.970E-01 |
| **CART** | 2.471E-05 | 3.868E-03 | 1.000E+00 | 9.999E-01 |
| **MLR** | 1.525E-01 | 2.482E+00 | 3.000E+00 | 5.999E-01 |

Table 4.31. Overall comparison of the v' coordinate prediction models with non-cross validated dataset (The best results are outlined in bold)

| Models | Performance Metrics | | | |
|---|---|---|---|---|
| | MSE | MAE | SEE | R |
| **FFNN** | **7.307E-06** | **1.870E-03** | 1.000E+00 | **9.999E-01** |
| **FITNET** | 8.708E-06 | 1.997E-03 | 1.000E+00 | 9.999E-01 |
| **CFNN** | 8.537E-06 | 1.965E-03 | 1.000E+00 | 9.999E-01 |
| **GRNN** | 7.995E-03 | 2.554E-01 | 1.000E+00 | 9.298E-01 |
| **SVR** | 4.985E-04 | 1.940E-02 | **9.998E-01** | 9.971E-01 |
| **CART** | 2.484E-05 | 3.835E-03 | 1.000E+00 | 9.998E-01 |
| **MLR** | 1.298E-01 | 2.300E+00 | 3.000E+00 | 6.000E-01 |

Table 4.32. Overall comparison of the w' coordinate prediction models with non-cross validated dataset (The best results are outlined in bold)

| Models | Performance Metrics | | | |
|---|---|---|---|---|
| | MSE | MAE | SEE | R |
| **FFNN** | 5.501E-08 | 1.531E-04 | 1.000E+00 | 1.000E+00 |
| **FITNET** | 5.554E-08 | 1.523E-04 | 1.000E+00 | 1.000E+00 |
| **CFNN** | **5.418E-08** | **1.498E-04** | 1.000E+00 | **1.000E+00** |
| **GRNN** | 8.088E-03 | 2.466E-01 | **9.587E-01** | 9.313E-01 |
| **SVR** | 4.321E-04 | 1.832E-02 | 9.997E-01 | 9.983E-01 |
| **CART** | 2.140E-05 | 3.021E-03 | 1.000E+00 | 9.999E-01 |
| **MLR** | 4.865E-02 | 1.476E-01 | 3.000E+00 | 6.000E-01 |

Table 4.33. Overall comparison of the best values of u' coordinate prediction models with 10-fold cross-validated dataset (The best results are outlined in bold)

| Models | Performance Metrics | | | |
|---|---|---|---|---|
| | MSE | MAE | SEE | R |
| **FFNN** | **5.702E-06** | **1.733E-03** | 1.000E+00 | **1.000E+00** |
| **FITNET** | 8.300E-06 | 1.937E-03 | 1.000E+00 | 9.999E-01 |
| **CFNN** | 7.940E-06 | 1.973E-03 | 1.000E+00 | 1.000E+00 |
| **GRNN** | 7.531E-03 | 2.459E-01 | **9.591E-01** | 9.433E-01 |
| **SVR** | 4.839E-04 | 1.879E-02 | 9.997E-01 | 9.972E-01 |
| **CART** | 2.398E-05 | 3.674E-03 | 1.000E+00 | 9.999E-01 |
| **MLR** | 1.099E-01 | 2.307E+00 | 3.000E+00 | 6.000E-01 |

Table 4.34. Overall comparison of the best values of v' coordinate prediction models with 10-fold cross-validated dataset (The best results are outlined in bold)

| Models | Performance Metrics | | | |
|---|---|---|---|---|
| | **MSE** | **MAE** | **SEE** | **R** |
| **FFNN** | **6.205E-06** | **1.755E-03** | 1.000E+00 | **1.000E+00** |
| **FITNET** | 7.605E-06 | 1.886E-03 | 1.000E+00 | 1.000E+00 |
| **CFNN** | 7.024E-06 | 1.859E-03 | 1.000E+00 | 1.000E+00 |
| **GRNN** | 7.423E-03 | 2.429E-01 | **9.593E-01** | 9.274E-01 |
| **SVR** | 4.767E-04 | 1.890E-02 | 9.997E-01 | 9.972E-01 |
| **CART** | 2.207E-05 | 3.582E-03 | 1.000E+00 | 9.999E-01 |
| **MLR** | 1.114E-01 | 2.195E+00 | 3.000E+00 | 6.000E-01 |

Table 4.35. Overall comparison of the best values of w' coordinate prediction models with 10-fold cross-validated dataset (The best results are outlined in bold)

| Models | Performance Metrics | | | |
|---|---|---|---|---|
| | **MSE** | **MAE** | **SEE** | **R** |
| **FFNN** | **5.848E-06** | **1.702E-03** | 1.000E+00 | **1.000E+00** |
| **FITNET** | 7.299E-06 | 1.837E-03 | 1.000E+00 | 1.000E+00 |
| **CFNN** | 7.081E-06 | 1.876E-03 | 1.000E+00 | 1.000E+00 |
| **GRNN** | 7.582E-03 | 2.370E-01 | **9.582E-01** | 9.474E-01 |
| **SVR** | 4.282E-04 | 1.810E-02 | 9.998E-01 | 9.982E-01 |
| **CART** | 1.683E-05 | 3.482E-03 | 1.000E+00 | 9.999E-01 |
| **MLR** | 4.110E-02 | 1.394E-01 | 3.000E+00 | 6.000E-01 |

Table 4.36. Overall comparison of the average values of u' coordinate prediction models with 10-fold cross-validated dataset (The best results are outlined in bold)

| Models | Performance Metrics | | | |
|---|---|---|---|---|
| | **MSE** | **MAE** | **SEE** | **R** |
| **FFNN** | **8.141E-06** | **1.960E-03** | 1.000E+00 | **1.000E+00** |
| **FITNET** | 9.522E-06 | 2.081E-03 | 1.000E+00 | 9.999E-01 |
| **CFNN** | 9.237E-06 | 2.093E-03 | 1.000E+00 | 9.999E-01 |
| **GRNN** | 7.782E-03 | 2.501E-01 | **9.603E-01** | 9.372E-01 |
| **SVR** | 5.174E-04 | 1.964E-02 | 9.997E-01 | 9.970E-01 |
| **CART** | 2.503E-05 | 3.798E-03 | 1.000E+00 | 9.999E-01 |
| **MLR** | 1.393E-01 | 2.471E+00 | 3.000E+00 | 5.999E-01 |

Table 4.37. Overall comparison of the average values of v' coordinate prediction models with 10-fold cross-validated dataset (The best results are outlined in bold)

| Models | Performance Metrics | | | |
|---|---|---|---|---|
| | **MSE** | **MAE** | **SEE** | **R** |
| **FFNN** | **7.175E-06** | **1.883E-03** | 1.000E+00 | **1.000E+00** |
| **FITNET** | 8.408E-06 | 1.954E-03 | 1.000E+00 | 9.999E-01 |
| **CFNN** | 7.797E-06 | 1.940E-03 | 1.000E+00 | 1.000E+00 |
| **GRNN** | 7.734E-03 | 2.492E-01 | **9.606E-01** | 9.209E-01 |
| **SVR** | 5.020E-04 | 1.943E-02 | 9.997E-01 | 9.971E-01 |
| **CART** | 2.353E-05 | 3.715E-03 | 1.000E+00 | 9.999E-01 |
| **MLR** | 1.349E-01 | 2.368E+00 | 3.000E+00 | 6.000E-01 |

Table 4.38. Overall comparison of the average values of w' coordinate prediction models with 10-fold cross-validated dataset (The best results are outlined in bold)

| Models | Performance Metrics | | | |
|---|---|---|---|---|
| | **MSE** | **MAE** | **SEE** | **R** |
| **FFNN** | **7.436E-06** | **1.898E-03** | 1.000E+00 | **1.000E+00** |
| **FITNET** | 8.434E-06 | 1.956E-03 | 1.000E+00 | 9.999E-01 |
| **CFNN** | 7.941E-06 | 1.970E-03 | 1.000E+00 | 1.000E+00 |
| **GRNN** | 7.860E-03 | 2.424E-01 | **9.599E-01** | 9.231E-01 |
| **SVR** | 4.569E-04 | 1.882E-02 | 9.998E-01 | 9.980E-01 |
| **CART** | 1.827E-05 | 3.649E-03 | 1.000E+00 | 9.999E-01 |
| **MLR** | 4.641E-02 | 1.447E-01 | 3.000E+00 | 6.000E-01 |

### 4.8. Improvement on CASTEP Simulation Time Consumption

Another usage of these predicted data is to decrease number of iterations in calculation process in CASTEP. If CASTEP simulation started with random atomic coordinates, the calculation may takes several days. The number of iterations in CASTEP can be decreased thus the calculation time decreases from days to minutes by using prediction results of proposed prediction models as initial atomic coordinates. The design of this approach is depicted in Figure 4.17. The random values of atomic coordinates are used as inputs of prediction models and after that the predicted values are used to calculate final atomic coordinates by CASTEP (two-staged simulation). Table 4.39 shows the number of iterations before and after using

the initial values that predicted by FITNET, FFNN, CFNN, GRNN, SVR, CART and MLR models. The rate of decline in iteration number is related to performances of proposed prediction models. The best decline is achieved by FITNET based model and the worst is MLR based model. Table 4.40 shows percentage of declines in number of iterations. As expected, the highest percentage of decline reached up to ≈85% by FITNET based model.

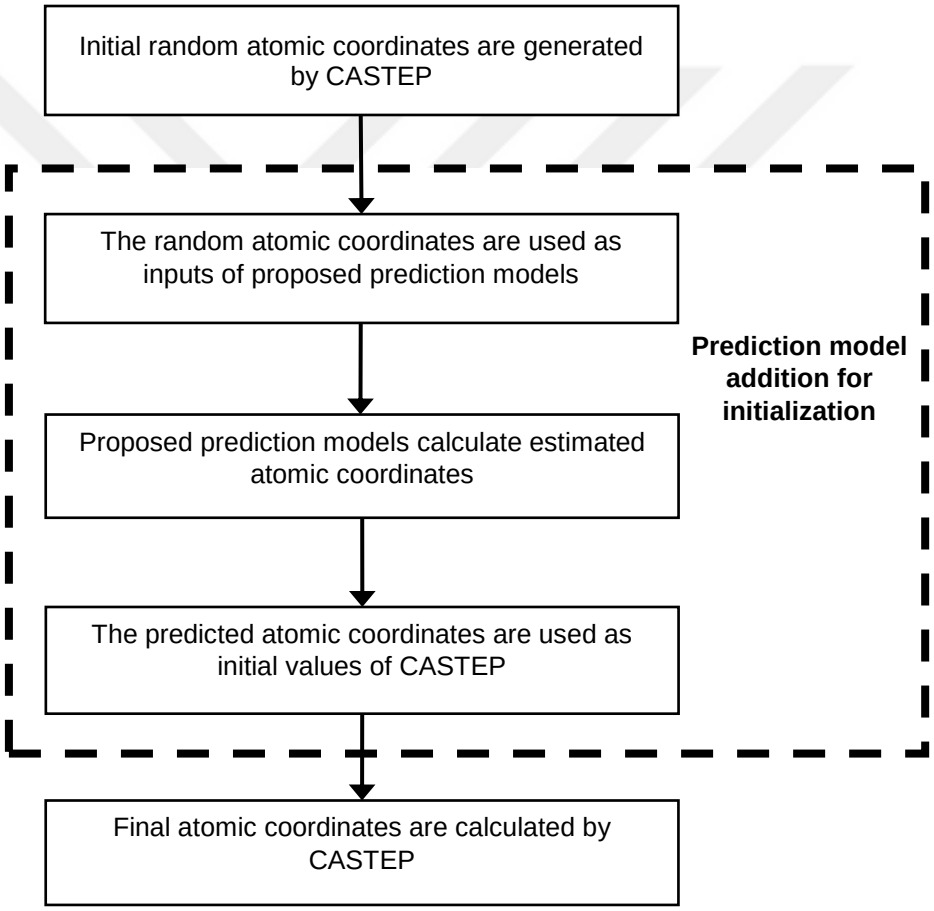



Figure 4.17. Steps of the two-staged CASTEP simulation

Table 4.39. Number of iterations before and after the two-staged CASTEP simulation

| Atom Number | Iteration Number | Number of Iterations After Applied Model | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | FITNET | FFNN | CFNN | GRNN | SVR | CART | MLR |
| 28 | 26 | 4 | 5 | 7 | 9 | 8 | 8 | 18 |
| 52 | 18 | 3 | 4 | 5 | 7 | 7 | 6 | 13 |
| 76 | 13 | 3 | 3 | 4 | 6 | 5 | 4 | 10 |
| 84 | 13 | 3 | 4 | 5 | 7 | 6 | 6 | 9 |
| 56 | 6 | 2 | 2 | 3 | 4 | 4 | 4 | 6 |
| 148 | 19 | 4 | 5 | 5 | 8 | 7 | 7 | 14 |
| 124 | 57 | 12 | 13 | 15 | 21 | 18 | 17 | 48 |
| 156 | 8 | 2 | 2 | 2 | 4 | 4 | 3 | 7 |
| 196 | 23 | 5 | 5 | 6 | 8 | 7 | 7 | 16 |
| 244 | 11 | 3 | 3 | 4 | 6 | 5 | 5 | 8 |
| 172 | 32 | 7 | 8 | 8 | 14 | 12 | 11 | 26 |
| 104 | 11 | 4 | 4 | 5 | 7 | 7 | 7 | 10 |
| 84 | 34 | 7 | 7 | 9 | 16 | 14 | 12 | 29 |
| 152 | 15 | 4 | 5 | 5 | 9 | 8 | 7 | 12 |
| 364 | 12 | 4 | 4 | 4 | 6 | 5 | 5 | 11 |
| 228 | 10 | 3 | 4 | 4 | 6 | 6 | 5 | 9 |
| 268 | 14 | 3 | 3 | 5 | 7 | 6 | 6 | 11 |
| 316 | 11 | 3 | 3 | 3 | 6 | 5 | 5 | 9 |
| 372 | 18 | 4 | 5 | 6 | 7 | 7 | 7 | 14 |
| 436 | 31 | 6 | 7 | 9 | 12 | 11 | 10 | 23 |
| 508 | 17 | 4 | 4 | 5 | 9 | 7 | 7 | 14 |
| 292 | 10 | 3 | 3 | 3 | 5 | 5 | 4 | 8 |
| 168 | 22 | 6 | 5 | 7 | 13 | 10 | 10 | 17 |
| 388 | 17 | 5 | 6 | 6 | 8 | 8 | 7 | 14 |
| 112 | 18 | 5 | 5 | 6 | 8 | 8 | 8 | 14 |
| 516 | 15 | 5 | 5 | 6 | 8 | 7 | 7 | 12 |
| 296 | 12 | 4 | 3 | 4 | 7 | 6 | 6 | 10 |
| 364 | 23 | 7 | 7 | 8 | 10 | 10 | 9 | 17 |
| 412 | 14 | 3 | 3 | 3 | 5 | 4 | 4 | 11 |
| 156 | 22 | 5 | 5 | 7 | 10 | 8 | 9 | 18 |
| 532 | 33 | 8 | 9 | 10 | 17 | 15 | 14 | 26 |
| 228 | 21 | 5 | 6 | 7 | 14 | 10 | 11 | 16 |
| 444 | 19 | 5 | 5 | 7 | 10 | 9 | 8 | 15 |
| 248 | 52 | 14 | 16 | 17 | 28 | 25 | 23 | 42 |
| 312 | 13 | 4 | 4 | 4 | 6 | 6 | 5 | 10 |
| 140 | 15 | 5 | 6 | 6 | 8 | 7 | 7 | 11 |
| 392 | 13 | 5 | 5 | 5 | 6 | 7 | 6 | 10 |
| 588 | 16 | 5 | 5 | 6 | 7 | 7 | 7 | 12 |
| 344 | 18 | 6 | 5 | 7 | 9 | 8 | 8 | 14 |
| 252 | 11 | 3 | 4 | 4 | 7 | 6 | 6 | 10 |
| 208 | 18 | 5 | 5 | 6 | 8 | 7 | 7 | 14 |
| 168 | 32 | 7 | 7 | 9 | 14 | 14 | 12 | 26 |

Table 4.40. Percentage of declines in number of iterations

| Atom Number | Iteration Number | Percentage of Decline After Applied Model (%) | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | FITNET | FFNN | CFNN | GRNN | SVR | CART | MLR |
| 28 | 26 | 84.62 | 80.77 | 73.08 | 65.38 | 69,23 | 69,23 | 30,77 |
| 52 | 18 | 83.33 | 77.78 | 72.22 | 61.11 | 61,11 | 66,67 | 27,78 |
| 76 | 13 | 76.92 | 76.92 | 69.23 | 53.85 | 61,54 | 69,23 | 23,08 |
| 84 | 13 | 76.92 | 69.23 | 61.54 | 46.15 | 53,85 | 53,85 | 30,77 |
| 56 | 6 | 66.67 | 66.67 | 50.00 | 33.33 | 33,33 | 33,33 | 0,00 |
| 148 | 19 | 78.95 | 73.68 | 73.68 | 57.89 | 63,16 | 63,16 | 26,32 |
| 124 | 57 | 78.95 | 77.19 | 73.68 | 63.16 | 68,42 | 70,18 | 15,79 |
| 156 | 8 | 75.00 | 75.00 | 75.00 | 50.00 | 50,00 | 62,50 | 12,50 |
| 196 | 23 | 78.26 | 78.26 | 73.91 | 65.22 | 69,57 | 69,57 | 30,43 |
| 244 | 11 | 72.73 | 72.73 | 63.64 | 45.45 | 54,55 | 54,55 | 27,27 |
| 172 | 32 | 78.13 | 75.00 | 75.00 | 56.25 | 62,50 | 65,63 | 18,75 |
| 104 | 11 | 63.64 | 63.64 | 54.55 | 36.36 | 36,36 | 36,36 | 9,09 |
| 84 | 34 | 79.41 | 79.41 | 73.53 | 52.94 | 58,82 | 64,71 | 14,71 |
| 152 | 15 | 73.33 | 66.67 | 66.67 | 40.00 | 46,67 | 53,33 | 20,00 |
| 364 | 12 | 66.67 | 66.67 | 66.67 | 50.00 | 58,33 | 58,33 | 8,33 |
| 228 | 10 | 70.00 | 60.00 | 60.00 | 40.00 | 40,00 | 50,00 | 10,00 |
| 268 | 14 | 78.57 | 78.57 | 64.29 | 50.00 | 57,14 | 57,14 | 21,43 |
| 316 | 11 | 72.73 | 72.73 | 72.73 | 45.45 | 54,55 | 54,55 | 18,18 |
| 372 | 18 | 77.78 | 72.22 | 66.67 | 61.11 | 61,11 | 61,11 | 22,22 |
| 436 | 31 | 80.65 | 77.42 | 70.97 | 61.29 | 64,52 | 67,74 | 25,81 |
| 508 | 17 | 76.47 | 76.47 | 70.59 | 47.06 | 58,82 | 58,82 | 17,65 |
| 292 | 10 | 70.00 | 70.00 | 70.00 | 50.00 | 50,00 | 60,00 | 20,00 |
| 168 | 22 | 72.73 | 77.27 | 68.18 | 40.91 | 54,55 | 54,55 | 22,73 |
| 388 | 17 | 70.59 | 64.71 | 64.71 | 52.94 | 52,94 | 58,82 | 17,65 |
| 112 | 18 | 72.22 | 72.22 | 66.67 | 55.56 | 55,56 | 55,56 | 22,22 |
| 516 | 15 | 66.67 | 66.67 | 60.00 | 46.67 | 53,33 | 53,33 | 20,00 |
| 296 | 12 | 66.67 | 75.00 | 66.67 | 41.67 | 50,00 | 50,00 | 16,67 |
| 364 | 23 | 69.57 | 69.57 | 65.22 | 56.52 | 56,52 | 60,87 | 26,09 |
| 412 | 14 | 78.57 | 78.57 | 78.57 | 64.29 | 71,43 | 71,43 | 21,43 |
| 156 | 22 | 77.27 | 77.27 | 68.18 | 54.55 | 63,64 | 59,09 | 18,18 |
| 532 | 33 | 75.76 | 72.73 | 69.70 | 48.48 | 54,55 | 57,58 | 21,21 |
| 228 | 21 | 76.19 | 71.43 | 66.67 | 33.33 | 52,38 | 47,62 | 23,81 |
| 444 | 19 | 73.68 | 73.68 | 63.16 | 47.37 | 52,63 | 57,89 | 21,05 |
| 248 | 52 | 73.08 | 69.23 | 67.31 | 46.15 | 51,92 | 55,77 | 19,23 |
| 312 | 13 | 69.23 | 69.23 | 69.23 | 53.85 | 53,85 | 61,54 | 23,08 |
| 140 | 15 | 66.67 | 60.00 | 60.00 | 46.67 | 53,33 | 53,33 | 26,67 |
| 392 | 13 | 61.54 | 61.54 | 61.54 | 53.85 | 46,15 | 53,85 | 23,08 |
| 588 | 16 | 68.75 | 68.75 | 62.50 | 56.25 | 56,25 | 56,25 | 25,00 |
| 344 | 18 | 66.67 | 72.22 | 61.11 | 50.00 | 55,56 | 55,56 | 22,22 |
| 252 | 11 | 72.73 | 63.64 | 63.64 | 36.36 | 45,45 | 45,45 | 9,09 |
| 208 | 18 | 72.22 | 72.22 | 66.67 | 55.56 | 61,11 | 61,11 | 22,22 |
| 168 | 32 | 78.13 | 78.13 | 71.88 | 56.25 | 56,25 | 62,50 | 18,75 |

## 5. CONCLUSIONS

In this thesis, Machine Learning based prediction models such as FFNN, FITNET, CFNN, GRNN, SVR and CART are proposed to predict the atomic coordinates of CNTs. A statistical regression method MLR is used for the purpose of comparison with Machine Learning methods. In order to carry out these prediction models, a dataset is created using CASTEP. The dataset of 10721 data samples contains three atomic coordinates (u, v, w) and the chiral vector (m, n) totally five inputs and three final atomic coordinates outputs (u', v', w'). The dataset is organized into two types for evaluation strategy: cross validated dataset and non-cross validated dataset. Training of the prediction models is done by 90% of the cross validated dataset. The rest of the dataset usually serves as an independent test set by which the generalization performance of the prediction models. The dataset is divided randomly into training, validating and test data randomly, in which there are 70-15-15% training, validating and testing sets, respectively for the non-cross validated dataset. MSE, MAE, SSE and R values of the developed models have been calculated for performance evaluation of the models.

The first aim of this thesis is to use predicted atomic coordinates instead of simulated ones. This aim is reached by i) obtaining very low MSE and MAE values with FFNN, FITNET and CFNN prediction models; ii) getting very high R values for all prediction models when the average values are considered for the cross validated dataset. FFNN, FITNET, CFNN, SVR and CART prediction models show relative typical results when assessing SEE and R. CFNN prediction model is a good predictor for u’ and w’ coordinates for the non-cross validated dataset. FFNN prediction model is the best for v’ coordinate prediction. The lowest MSE and MAE values of this study are obtained from w’ coordinate prediction when the cross validation is not taken into consideration.

FITNET, FFNN and CFNN prediction models perform good levels of successes for all average MSE, MAE, SSE and R performance measures obtained for both u', v' and w' coordinates. FITNET prediction model performs better than the other prediction models. The best performance measure results are achieved with FFNN prediction model which tested cross-validated dataset. If the cross-validation is not used, FFNN and FITNET prediction models show nearly the same performance.

The second aim of this thesis is to use predicted atomic coordinates as initial values to calculate simulation results in CASTEP. This aim is reached by decreasing the number of iterations up to ≈85% of their first value. The use of Machine Learning methods ensures that the exact results of atomic coordinates can be obtained in a shorter time.

Consequently, regression based Machine Learning methods can be used as a predictor of atomic coordinates of CNTs in the field of nanotechnology. The findings obtained by this study demonstrate the benefits of employing Machine Learning techniques on the calculations and simulations that have very long execution time.

As a result of this work, the effectiveness of artificial intelligence solutions, which estimate atomic coordinates that can be integrated into software, has been observed and the results that have significantly shortened the simulation processes in the field of nanotechnology have been presented to be integrated into scientific or commercial software.

## REFERENCES


Abo-Elhadeed, A. F., 2012. Modeling Carbon Nanotube Transistors Using Neural Networks Approach. International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD),125-128.

Ajayan, P. and Ebbesen, T., 1997. Nanometre-Size Tubes of Carbon. Reports on Progress in Physics, 60(10): 1025.

Akai, Y. and Saito, S., 2005. Electronic Structure, Energetics and Geometric Structure of Carbon Nanotubes: A Density-Functional Study. Physica E: Low-Dimensional Systems and Nanostructures, 29(3): 555-559.

Akbari, E., Buntat, Z., Enzevaee, A., Ebrahimi, M., Yazdavar, A. H. and Yusof, R., 2014. Analytical Modeling and Simulation of I–V Characteristics in Carbon Nanotube Based Gas Sensors Using ANN and SVR Methods. Chemometrics and Intelligent Laboratory Systems, 137(1): 173-180.

Asıltürk, I. and Çunkaş, M., 2011. Modeling and Prediction of Surface Roughness in Turning Operations Using Artificial Neural Network and Multiple Regression Method. Expert Systems with Applications, 38(5): 5826-5832.

Bauschlicher, C. and Ricca, A., 2004. Binding of NH3 to Graphite and to a (9, 0) Carbon Nanotube. Physical Review B, 70(11): 115409(1-6).

Beale, M. H., Hagan, M. T. and Demuth, H. B., 2012. Neural Network Toolbox™ User's Guide. The MathWorks, Inc., Natick, 410p.

Breiman, L., Friedman, J., Stone, C. J. and Olshen, R. A., 1984. Classification and Regression Trees. CRC Press, Boca Raton, 368p.

Burges, C. J., 1998. A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery, 2(2): 121-167.

CASTEP, 2016. Biovia Materials, http://accelrys.com/products/collaborative-science/biovia-materials-studio/quantom-catalysis-software.html.

Chen, C. M., Hsu, C. Y., Chiu, H. W. and Rau, H. H., 2011. Prediction of Survival in Patients with Liver Cancer Using Artificial Neural Networks and Classification and Regression Trees. Seventh International Conference on Natural Computation (ICNC), 811-815.

Cheng, W., Cai, C., Luo, Y., Li, Y. and Zhao, C., 2015. Mechanical Properties Prediction for Carbon Nanotubes/Epoxy Composites by Using Support Vector Regression. Modern Physics Letters B, 29(05): 1550016.

Chico, L., Crespi, V. H., Benedict, L. X., Louie, S. G. and Cohen, M. L., 1996. Pure Carbon Nanoscale Devices: Nanotube Heterojunctions. Physical Review Letters, 76(6): 971-974.

Clark, S. J., Segall, M. D., Pickard, C. J., Hasnip, P. J., Probert, M. I., Refson, K. and Payne, M. C., 2005. First Principles Methods Using CASTEP. Zeitschrift für Kristallographie-Crystalline Materials, 220(5/6): 567-570.

Dasgupta, A., Sun, Y. V., König, I. R., Baıley-Wilson, J. E. and Malley, J. D., 2011. Brief Review of Regression-Based and Machine Learning Methods in Genetic Epidemiology: The Genetic Analysis Workshop 17 Experience. Genetic Epidemiology, 35(1): 5-11.

Dorofki, M., Elshafie, A. H., Jaafar, O., Karim, O. A. and Mastura, S., 2012. Comparison of Artificial Neural Network Transfer Functions Abilities to Simulate Extreme Runoff Data. International Proceedings of Chemical, Biological and Environmental Engineering, 33(1): 39-44.

Eatemadi, A., Daraee, H., Karimkhanloo, H., Kouhi, M., Zarghami, N., Akbarzadeh, A., Abasi, M., Hanifehpour, Y. and Joo, S.W., 2014. Carbon Nanotubes: Properties, Synthesis, Purification, and Medical Applications. Nanoscale Research Letters, 9(1): 393-405.

Elliott, L. and Owens, L., 2015. Cart Analysis of Environmental Factors, Biomarkers and Gill-Associated Virus to Predict Production Outcomes for Farmed Penaeus Monodon. Aquaculture, 448(1): 298-305.

Ensafi, A. A., Taei, M., Khayamian, T. and Hasanpour, F., 2010. Simultaneous Voltammetric Determination of Enrofloxacin and Ciprofloxacin in Urine and Plasma Using Multiwall Carbon Nanotubes Modified Glassy Carbon Electrode by Least-Squares Support Vector Machines. Analytical Sciences, 26(7): 803-808.

Geng, W., Liu, H. and Yao, X., 2013. Enhanced Photocatalytic Properties of Titania-Graphene Nanocomposites: A Density Functional Theory Study. Physical Chemistry Chemical Physics, 15(16): 6025-6033.

Gernand, J. M. and Casman, E. A., 2014. A Meta-Analysis of Carbon Nanotube Pulmonary Toxicity Studies-How Physical Dimensions and Impurities Affect the Toxicity of Carbon Nanotubes. Risk Analysis, 34(3): 583-597.

Ghaboussi, J., Garrett Jr, J. and Wu, X., 1991. Knowledge-Based Modeling of Material Behavior with Neural Networks. Journal of Engineering Mechanics, 117(1): 132-153.

Ghaedi, M., Reza Rahimi, M., Ghaedi, A., Tyagi, I., Agarwal, S. and Gupta, V. K., 2016. Application of Least Squares Support Vector Regression and Linear Multiple Regression for Modeling Removal of Methyl Orange onto Tin Oxide Nanoparticles Loaded on Activated Carbon and Activated Carbon Prepared from Pistacia Atlantica Wood. Journal of Colloid and Interface Science, 461(1): 425-434.

Hakim, S., 2006. Development of Artificial Neural Network with Application to Some Civil Engineering Problems. Master Thesis, University Putra Malaysia.

Hassanzadeh, Z., Kompany-Zareh, M., Ghavami, R., Gholami, S. and Malek-Khatabi, A., 2015. Combining Radial Basis Function Neural Network with Genetic Algorithm to QSPR Modeling of Adsorption on Multi-Walled Carbon Nanotubes Surface. Journal of Molecular Structure, 1098(1): 191-198.

Hayati, M., Rezaei, A. and Seifi, M., 2010. CNT-MOSFET Modeling Based on Artificial Neural Network: Application to Simulation of Nanoscale Circuits. Solid-State Electronics, 54(1): 52-57.

Hedayat, A., Davilu, H., Barfrosh, A. A. and Sepanloo, K., 2009. Estimation of Research Reactor Core Parameters Using Cascade Feed Forward Artificial Neural Networks. Progress in Nuclear Energy, 51(6): 709-718.

Hohenberg, P. and Kohn, W., 1964. Inhomogeneous Electron Gas. Physical Review 136(B864–B871).

Hong-Xia, L., He-Ming, Z., Jiu-Xu, S. and Zhi-Yong, Z., 2010. Electronic Transport Properties of an (8,0) Carbon/Boron Nitride Nanotube Heterojunction. Chinese Physics B, 19(3): 037104(1-5).

Hong-Xia, L., He-Ming, Z. and Zhiyong, Z., 2009. Electronic Transport Properties of an (8,0) Carbon/Silicon-Carbide Nanotube Heterojunction. Journal of Semiconductors, 30(5): 052002(1-4).

Iijima, S., 1991. Helical Microtubules of Graphitic Carbon. Nature, 354(6348): 56-58.

Iijima, S. and Ichihashi, T., 1993. Single-Shell Carbon Nanotubes of 1-nm Diameter. Nature, 363(1): 603- 605.

Jang, J. S. R., Sun, C. T. and Mizutani, E., 1997. Neuro-Fuzzy and Soft Computing; a Computational Approach to Learning and Machine Intelligence. Pearson, New Jersey, 614p.

Kanamitsu, K. and Saito, S., 2002. Geometries, Electronic Properties, and Energetics of Isolated Single Walled Carbon Nanotubes. Journal of the Physical Society of Japan, 71(2): 483-486.

Kim, B., Lee, D.W., Park, K. Y., Choi, S.R. and Choi, S., 2004. Prediction of Plasma Etching Using a Randomized Generalized Regression Neural Network. Vacuum, 76(1): 37-43.

Kim, H., Lee, J., Kahng, S.-J., Son, Y.-W., Lee, S., Lee, C.-K., Ihm, J. and Kuk, Y., 2003. Direct Observation of Localized Defect States in Semiconductor Nanotube Junctions. Physical Review Letters, 90(21): 216107.

Kleinman, L. and Bylander, D., 1982. Efficacious Form for Model Pseudopotentials. Physical Review Letters, 48(20): 1425-1428.

Kohavi, R., 1995. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. Proceedings of the 14th International Joint Conference on Artificial Intelligence, San Francisco, 1137-1145.

Kohn, W., Becke, A. D. and Parr, R. G., 1996. Density Functional Theory of Electronic Structure. The Journal of Physical Chemistry, 100(31): 12974-12980.

Kohn, W. and Sham, L., 1965. Quantum Density Oscillations in an Inhomogeneous Electron Gas. Physical Review, 137(6A): A1697.

Krasnenko, V., Boltrushko, V., Klopov, M. and Hizhnyakov, V., 2014. Conjoined Structures of Carbon Nanotubes and Graphene Nanoribbons. Physica Scripta, 89(4): 044008.

Kresse, G. and Furthmüller, J., 1996. Efficient Iterative Schemes for Ab Initio Total-Energy Calculations Using a Plane-Wave Basis Set. Physical Review B, 54(16): 11169-11186.

Kürti, J., Zólyomi, V., Kertesz, M. and Sun, G., 2003. The Geometry and the Radial Breathing Mode of Carbon Nanotubes: Beyond the Ideal Behaviour. New Journal of Physics, 5(1): 125(1-21).

Lee, T. S., Chiu, C. C., Chou, Y. C. and Lu, C. J., 2006. Mining the Customer Credit Using Classification and Regression Tree and Multivariate Adaptive Regression Splines. Computational Statistics & Data Analysis, 50(4): 1113-1130.

Li, J., Ji, X., Jia, Y., Zhu, B., Wang, G., Li, Z. and Liu, X., 2014. Hard Drive Failure Prediction Using Classification and Regression Trees. 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, 383-394.

Lu, Z. and Yang, H., 2012. Unlocking the Power of OPNET Modeler. Cambridge University Press, Cambridge, 254p.

Mao, Y. and Zhong, J., 2009. The Computational Design of Junctions by Carbon Nanotube Insertion into a Graphene Matrix. New Journal of Physics, 11(9): 093002(1-12).

Marzari, N., Vanderbilt, D. and Payne, M. C., 1997. Ensemble Density-Functional Theory for Ab Initio Molecular Dynamics of Metals and Finite-Temperature Insulators. Physical Review Letters, 79(7): 1337-1340.

MATLAB, 2016. https://www.mathworks.com/.

Moradian, R., Behzad, S. and Chegel, R., 2008. Ab Initio Density Functional Theory Investigation of Structural and Electronic Properties of Silicon Carbide Nanotube Bundles. Physica B: Condensed Matter, 403(19): 3623-3626.

Moradian, R., Behzad, S. and Chegel, R., 2009a. Ab Initio Density Functional Theory Investigation of Li-Intercalated Silicon Carbide Nanotube Bundles. Physics Letters A, 373(26): 2260-2266.

Moradian, R., Behzad, S. and Chegel, R., 2009b. Ab Initio Density Functional Theory Investigation of Structural and Electronic Properties of Double-Walled Silicon Carbide Nanotubes. Physica E: Low-Dimensional Systems and Nanostructures, 42(2): 172-175.

Moré, J. J., 1978. The Levenberg-Marquardt Algorithm: Implementation and Theory (G. A. Watson Editor). Numerical Analysis, Springer, p.105-116.

Murphy, K. P., 2012. Machine Learning: A Probabilistic Perspective. MIT Press, London, 1104p.

Myers, W. and Swiatecki, W., 1996. Nuclear Properties According to the Thomas-Fermi Model. Nuclear Physics A, 601(2): 141-167.

Nilsson, N. J., 1996. Introduction to Machine Learning. Stanford University Press, Stanford, 188p.

Norman, R. D., 1998. Applied Regression Analysis. Third Edition Wiley-Interscience, Canada, 736p.

O'connell, M. J., 2006. Carbon Nanotubes: Properties and Applications. Taylor & Francis, London, 360p.

Ouyang, M., Huang, J. L., Cheung, C. L. and Lieber, C. M., 2001. Atomically Resolved Single-Walled Carbon Nanotube Intramolecular Junctions. Science, 291(5501): 97-100.

Parsons, T. D., Rizzo, A. A. and Buckwalter, J. G., 2004. Backpropagation and Regression: Comparative Utility for Neuropsychologists. Journal of Clinical and Experimental Neuropsychology, 26(1): 95-104.

Patil, S. R., 2008. Regionalization of an Event Based Nash Cascade Model for Flood Predictions in Ungauged Basins. PhD Thesis, Universität Stuttgart.

Payne, M.C., Teter, M.P., Allan, D.C., Arias, T. and Joannopoulos, J., 1992. Iterative Minimization Techniques for Ab Initio Total-Energy Calculations: Molecular Dynamics and Conjugate Gradients. Reviews of Modern Physics, 64(4): 1045.

Quinlan, J. R., 1993. C4.5: Programs for Machine Learning. Elsevier, California, 302p.

Rahimi-Nasrabadi, M., Akhoondi, R., Pourmortazavi, S. M. and Ahmadi, F., 2015. Predicting Adsorption of Aromatic Compounds by Carbon Nanotubes Based on Quantitative Structure Property Relationship Principles. Journal of Molecular Structure, 1099(1): 510-515.

Raja, M. A. Z., Farooq, U., Chaudhary, N. I. and Wazwaz, A. M., 2016. Stochastic Numerical Solver for Nanofluidic Problems Containing Multi-Walled Carbon Nanotubes. Applied Soft Computing, 38(1): 561-586.

Salehi, E., Abdi, J. and Aliei, M. H., 2016. Assessment of Cu (II) Adsorption from Water on Modified Membrane Adsorbents Using LS-SVM Intelligent Approach. Journal of Saudi Chemical Society, 20(2):213-219.

Sánchez-Portal, D., Artacho, E., Soler, J. M., Rubio, A. and Ordejón, P., 1999. Ab Initio Structural, Elastic, and Vibrational Properties of Carbon Nanotubes. Physical Review B, 59(19): 12678-12687.

Schölkopf, B. and Burges, C. J., 1999. Advances in Kernel Methods: Support Vector Learning. MIT Pressi New York, 386p.

Segall, M., Lindan, P. J., Probert, M. A., Pickard, C., Hasnip, P., Clark, S. and Payne, M., 2002. First-Principles Simulation: Ideas, Illustrations and the CASTEP Code. Journal of Physics: Condensed Matter, 14(11): 2717.

Shanbedi, M., Jafari, D., Amiri, A., Heris, S. Z. and Baniadam, M., 2013. Prediction of Temperature Performance of a Two-Phase Closed Thermosyphon Using Artificial Neural Network. Heat and Mass Transfer, 49(1): 65-73.

Smola, A. and Vapnik, V., 1997. Support Vector Regression Machines. Advances in Neural Information Processing Systems, 9(1): 155-161.

Şen, Z., 2004. Principles of Artificial Neural Networks. Water Foundation Publications, İstanbul, 185p.

Taghavifar, H. and Mardani, A., 2014. A Comparative Trend in Forecasting Ability of Artificial Neural Networks and Regressive Support Vector Machine Methodologies for Energy Dissipation Modeling of Off-Road Vehicles. Energy, 66(1): 569-576.

Tao, Y., Xue, Q., Liu, Z., Shan, M., Ling, C., Wu, T. and Li, X., 2014. Tunable Hydrogen Separation in Porous Graphene Membrane: First-Principle and Molecular Dynamic Simulation. ACS Applied Materials & Interfaces, 6(11): 8048-8058.

Tserpes, K. I. and Silvestre, N., 2014. Modeling of Carbon Nanotubes, Graphene and Their Composites. Springer, London, 332p.

Uslan, V. 2015. Support Vector Machine-Based Fuzzy Systems for Quantitative Prediction of Peptide Binding Affinity. PhD, De Montfort University.

Van Regemorter, T. and Larsson, K., 2008. Effect of Substitutional N on the Diamond CVD Growth Process: A Theoretical Approach. Diamond and Related Materials, 17(7): 1076-1079.

Vander Wal, R., Berger, G. and Ticich, T., 2003. Carbon Nanotube Synthesis in a Flame Using Laser Ablation for in Situ Catalyst Generation. Applied Physics A, 77(7): 885-889.

Vanderbilt, D., 1990. Soft Self-Consistent Pseudopotentials in a Generalized Eigenvalue Formalism. Physical Review B, 41(11): 7892-7895.

Vapnik, V., 1995. The Nature of Statistical Learning Theory. Springer Science & Business Media, New Jersey, USA, 314p.

Wang, L., Luo, G., Valencia, D., Llavina, C. H. S., Sabirianov, R. F., Lu, J., Lu, J. Q., Mei, W. N. and Cheung, C. L., 2013. Electronic Structures and Properties of Lanthanide Hexaboride Nanowires. Journal of Applied Physics, 114(14): 143709(1-9).

Witten, I. H. and Frank, E., 2005. Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, Burlington, 664p.

Xu, Y., Kong, Q. J. and Liu, Y., 2013. Short-Term Traffic Volume Prediction Using Classification and Regression Trees. Intelligent Vehicles Symposium, 493-498.

Yagi, Y., Briere, T. M., Sluiter, M. H., Kumar, V., Farajian, A. A. and Kawazoe, Y., 2004. Stable Geometries and Magnetic Properties of Single-Walled Carbon Nanotubes Doped with 3D Transition Metals: A First-Principles Study. Physical Review B, 69(7): 075414(1-9).

Zayid, E. I. M. and Akay, M. F., 2013. Predicting the Performance Measures of a Message-Passing Multiprocessor Architecture Using Artificial Neural Networks. Neural Computing and Applications, 23(7-8): 2481-2491.

## BIOGRAPHY

Mehmet ACI was born in Adana in 1981. He graduated from Adana Anatolian High School in 1999. He received his BSc degree in Computer Engineering, Eastern Mediterranean University in 2004. He completed MSc program of the Computer Engineering, Çukurova University in 2009. He received his PhD degree in Electric and Electronics Engineering, Çukurova University in 2016. His research interests are machine learning and artificial intelligence.