High throughput screening of thermal interface materials by machine learning

Tengtao Wu

Shanghai Datong High School, Shanghai, China

wutt2006@163.com

Abstract. Till now, it remains a challenge for effective prediction and screening of novel materials with high thermal conductivity, as well as further optimization of the interface thermal resistance. Normally, people have to spend long time on tedious calculations when predicting and screening these materials. In this paper, I combined machine learning with molecular dynamics simulations to investigate the thermal conductive properties of materials with the aim of significantly reducing computational consumption. I first applied molecular dynamics simulations to obtain the relevant properties of materials, then generated models for predicting physical properties by machine learning, and finally made predictions of thermophysical properties of materials. The use of machine learning significantly reduces the prediction time compared to direct molecular dynamics simulations. Especially when the XGBoost and the neural network models are employed, the prediction efficiency is significantly improved. This work guides a new way for the future screening of high-performance thermal interface materials.

Keywords: High-performance thermal interface materials, Machine learning, Molecular dynamics, XGBoost, Neural network

1. Introduction

The efficient screening of thermal conductive materials and optimization of interfacial thermal resistance is a key issue governing the development of modern electronics. The use of theoretical simulations to explore material structure-property relationships is expected to accelerate the development of high-performance thermal interface materials. The use of molecular dynamics (MD) simulations to predict the thermal properties of materials has a history of more than 60 years. In last century, molecular dynamics methods based on the rigid sphere potential have been proposed [1]. Later, the properties of liquid argon were simulated using the Lennard-Jones potential function [2]. With the development of simulation methods, the understanding of molecular dynamics simulations has become more profound. The simulation conditions such as boundary conditions, force calculations, evolution algorithms, and force field parameters of molecular simulations have been explored in depth [3]. There are various methods to study the thermal transport properties of materials through MD simulations, including the calculation of thermal conductivity using equilibrium and nonequilibrium molecular dynamics.

With the development of computer technology, machine learning methods were also gradually used to study the thermophysical properties of materials. For example, in 2019 Alireza Moradzadeh and Narayana R. Aluru successfully used machine learning to model the radial distribution function properties of hydrodynamics, demonstrating the successful application of machine learning in this field [4]. Machine learning was used to better predict the interface thermal resistance, and to construct regression models for predicting the interatomic interaction forces seems indeed promising [5].

The progress of modern supercomputers combined with the sophisticated commercial software for molecular dynamics such as LAMMPS has steadily increased the computational speed of materials simulations [6]. Molecular dynamics simulations are playing an increasingly prominent role in research such as applied physics, materials science and nanoscale heat transfer.

In this project, I generated regression models for predicting thermophysical properties of materials based on molecular dynamics simulation results, which can significantly accelerate material design and guide effective modulation of nanostructures to achieve optimized property.

2. Methodology

2.1. Molecular dynamics simulation and classical heat transfer theory

Molecular dynamics simulation is an important method to study the structure and properties of atomic or molecular systems through computational simulations.

The trajectories of the particles are obtained by the Velocity-Verlet algorithm [7] in the molecular dynamics simulation, and the thermodynamic properties in the system can be found by the thermodynamic equation [8] based on the motion information of the particle trajectories. It is based on the classical mechanics of atomic or molecular systems and numerically solves the Newtonian equations of motion of all particles to obtain the coordinates and momentum of these particles at each moment, i.e. the trajectories in phase space, and then calculates the macroscopic properties of the system statistically.

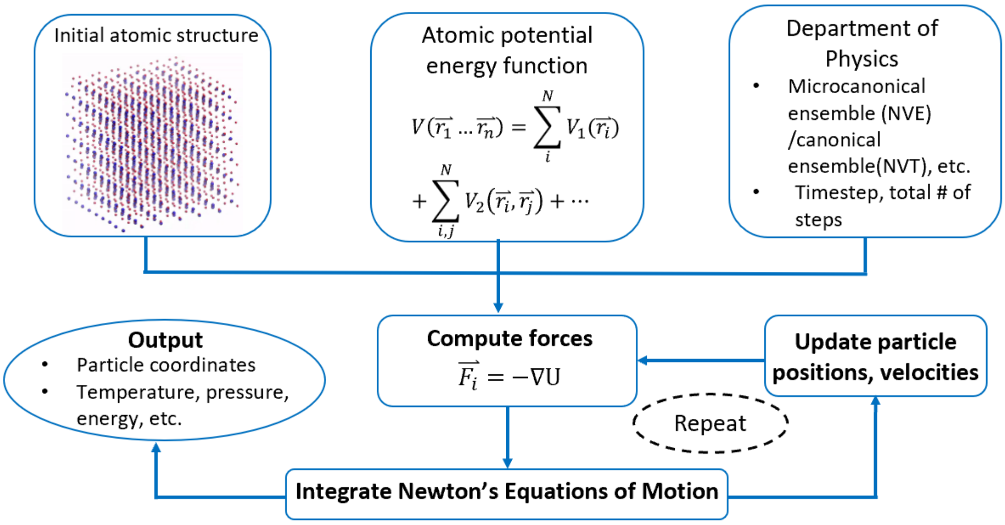
Initial atomic structure
Atomic potential energy function
$V(\vec{r_1}\ldots\vec{r_n}) = \sum_{i}^{N} V_1(\vec{r_i}) + \sum_{i,j}^{N} V_2(\vec{r_i},\vec{r_j}) + \cdots$
Department of Physics
• Microcanonical ensemble (NVE) /canonical ensemble(NVT), etc.
• Timestep, total # of steps
Output
• Particle coordinates
• Temperature, pressure, energy, etc.
Compute forces
$\vec{F_i} = -\nabla U$
Update particle positions, velocities
Repeat
Integrate Newton's Equations of Motion

Figure 1. Flow chart of molecular dynamics simulation.

According to the thermodynamics equations and Fourier's law, if a system is at different temperatures everywhere without an external heat source, the system is not in equilibrium and there is thermal conduction and a tendency to move toward equilibrium. Fourier's law describes the heat flux flow in this process, i.e., the amount of heat passing through a unit area per unit time:

$$J = -K\frac{dT}{dX} \tag{1}$$

in which J is the heat flux in W/m^2, T is the temperature, x is the direction of the heat flow, and K is the thermal conductivity. This equation reflects the tendency of heat transport, where the negative sign expresses the direction of heat flux in the opposite direction of the temperature gradient. Using this equation, not only the thermal conductivity can be calculated, but also the interfacial thermal conductivity can be simulated by performing nonequilibrium molecular dynamics simulations.

2.2. Construction of atomic models

All the MD simulations are performed by using LAMMPS. I implemented nonequilibrium molecular dynamics simulations to calculate the heat flux and monitor the temperature variation. An orthogonal three-dimensional simulation box is used. To facilitate the imposition of boundary conditions for heat flux in nonequilibrium molecular dynamics, I set the size of the simulation box along the heat flux direction (x) to 10.2 nm and the box cross-sectional area (yz cross-section) to 1.5 nm*1.5 nm. Periodic boundary condition in y and z directions is considered for the whole system [9]. Atoms at both ends are fixed to prevent the whole system from moving. The entire system is first relaxed at 300 K by using the Langevin thermostat algorithm, first for 0.1 ns under the NVE ensemble and then for 0.1 ns under the NVT ensemble. After that, the nonequilibrium molecular dynamics simulations are started, using the Langevin thermostat to form two thermal reservoirs with constant temperature (1.5 nm shown in Figure 2). The heat and cold reservoirs are adjacent to the two ends. The high- and low-temperature regions are set to be at 320 K and 280 K, respectively. Under canonical ensemble, the temperatures of the high- and low-temperature regions are kept constant by using Noose-Hover thermostat [10], to make the high-temperature region supply energy and the low-temperature region extract energy. The heat flux will then flow from the hot reservoirs to the cold reservoirs. After 0.5 ns the simulated system reaches an equilibrium state and the data are collected for 1 ns afterwards. The system is divided into a grid of 100 in the x-direction, and the profile of the steady-state temperature variation along the x-direction is obtained by averaging the data in each grid. The thermal conductivity was simulated based on the properties of the middle region, which is normally different from the hot and cold reservoirs to result in a temperature jump at the interface and the interfacial thermal resistance.

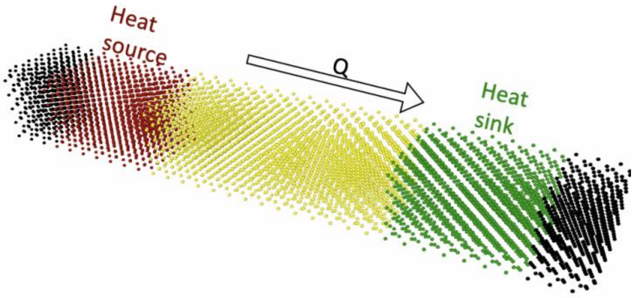

Figure 2. Model structure of non-equilibrium molecular dynamics.

I studied material systems in two different phase states, i.e., solid and fluid. For the ·solid-solid system, the interactions between atoms are described by the optimized Tersoff potential [11]. The time step of the MD simulations is 1 fs. At the interface, the non-bonded interactions are derived by the Lorentz-Berthelot [12] mixing rule:

$$\sigma_{12} = \frac{\sigma_1+\sigma_2}{2} \tag{2}$$

$$\epsilon_{12} = \sqrt{(\epsilon_1\epsilon_2)} \tag{3}$$

The atomic masses and lattice constants were chosen as features. The lattice constants range from 5.0 Å to 6.3 Å and the atomic masses are between 25 g/mol and 80 g/mol, which are determined by the atomic masses of Si and Ge, the two most commonly used semiconductor materials in electronic devices [13]. I studied 728 different solid-solid atomic structures and predicted the temperature variations by MD simulations.

For the solid-fluid system, the middle region of the model was filled with Lennard-Jones (LJ) atomic fluids [14], which are considered as simple single-atom fluids. The LJ interaction equation is given by the following equation:

$$U_{LJ} = 4\epsilon[(\frac{\sigma}{r})^{12} - (\frac{\sigma}{r})^{6}] \tag{4}$$

To introduce a reasonable fluid model inside the simulated box, I first froze all atoms in the solid-phase region, heated fluid atoms to 8000 K and maintained it for 0.1 ns using the Langevin thermostat, and then slowly quenched them to 300 K at a rate of 7.7 × 10^{-10} K/s. Afterwards, the whole system is optimized using the conjugate gradient method [15].

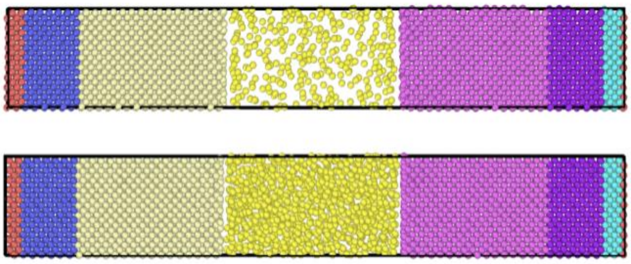

Figure 3. Schematic diagram of non-equilibrium molecular dynamics of a solid-fluid system.

For solid-fluid systems, the number of fluid atoms and the atomic mass are chosen as features. Actually, these two characteristics determine the fluid density.

$$\rho = \frac{M}{V} = \frac{\frac{Nm}{6.02\times10^{23}}}{V} \tag{5}$$

in which ρ is the mass density, N is the number of fluid atoms, m is the atomic mass, and V is the volume of the nanochannel. I studied 2520 solid-fluid systems and predicted the temperature variations by molecular dynamics simulations. The highest densities used in this study were close to those of amorphous solids or some special liquids (e.g., mercury).

Table 1. Solid-fluid characteristic table.

m [g/mol]	N	ρ [g/cm^3]
15	300	0.22
	2000	1.49
80	300	1.82
	2000	7.52

2.3. *Machine learning*

In this work, two methods, XGBoost and neural network, are used to predict the temperature variation at the solid-solid interface and the solid-liquid interface [16]. The framework is shown below.

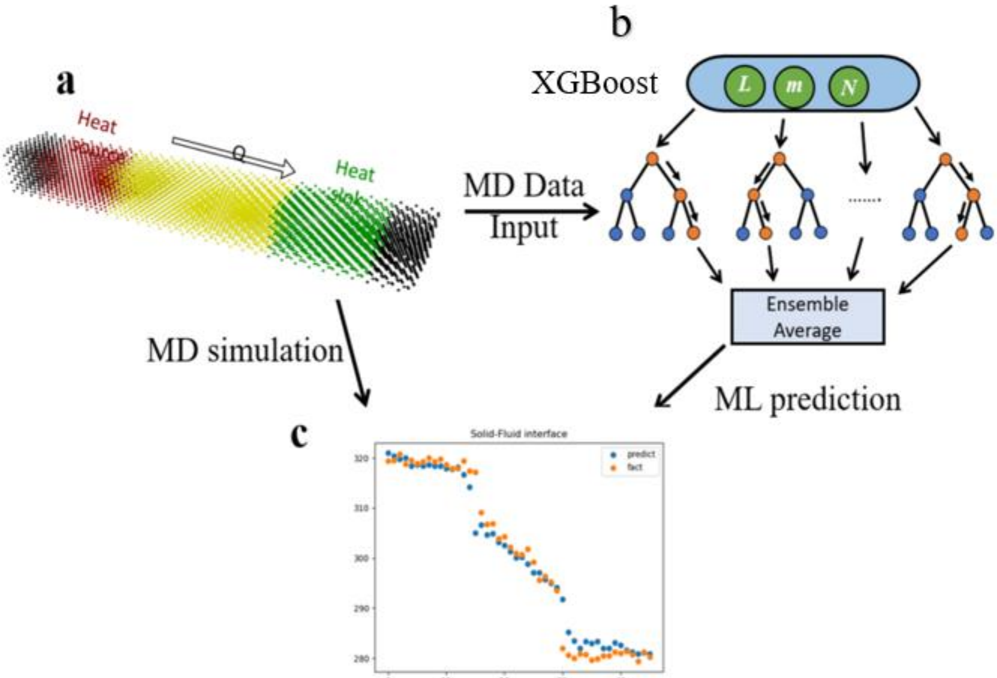

Figure 4. Machine learning workflow for predicting thermophysical properties.

2.3.1. XGBoost. Extreme Gradient Boosting (XGBoost [17]) is an enhanced integrated machine learning method for supervised problems. I employed the C++ version of the Gradient Boosting Machine, a package developed by Chen that automatically works in multi-processing mode to improve the efficiency. After extensive optimization, this model has possessed excellent accuracy and become a leader in the Kaggle competition.

The model combines several “underperforming” classifiers and regressors to form a unified model that is used in a decision tree by weighting the sum of each result of the regressors [18]. Compared with gradient boosting trees [19], XGBoost uses improved regularization. The regularization in XGBoost considers not only the leaves in the model, but also the weights of the leaves. At each iteration, a new tree is added to the integration based on the residuals of the previous tree. The schematic of XGBoost is shown below:

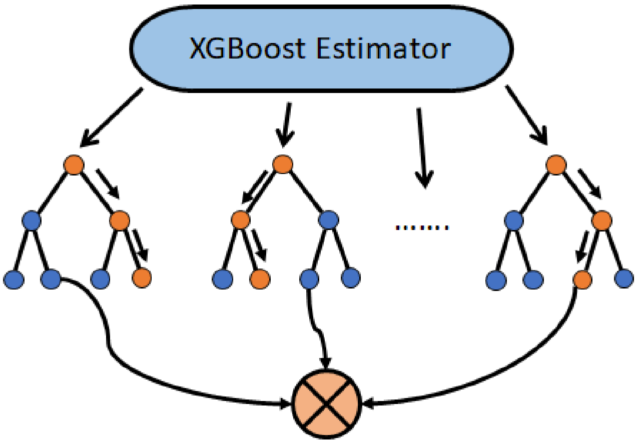

Figure 5. The schematic of XGBoost.

The hyperparameters of the XGBoost model are a key component. Optimal hyperparameters can prevent underfitting and overfitting, as well as more accurate results. The first part is the boosting parameter, which controls the integration, such as the number of Booster or threads to be used. The second part is the tree hyperparameter, which controls the performance of each tree. In this work, the stochastic search module in Sklearn is used to search for the best hyperparameters in the input set. Compared to grid search, stochastic search is faster and yields similar results.

XGBoost is easily implemented in the Python interface and can handle single-value regression problems. But in this work, temperatures at multiple regions should be predicted, corresponding to a multi-value prediction problem. The objective function of XGBoost can be customized to fit multi-value regression. Here, the Multi Output Regressor module in the Sklearn package is used with XGBoost to solve the multi-value regression. Each output has the same hyperparameters to save computational time and storage.

2.3.2. Neural network. Neural network [20, 21] is a network model consisting of artificial neurons (i.e., perceptron). The original idea dates back to the 1950s. It became a popular topic and tool as the significant progress in supercomputer between the 1990s and 2000s. Neural network model can solve classification and regression end-to-end, indicating that the model can learn the appropriate features themselves from the dataset without explicit programming. Based on the Universal Approximation Theorem [22] (UAT), a feedforward neural network with one hidden layer combined with an appropriate nonlinear activation function [23] can approximate any continuous function. Considering the complexity of the simulated dataset, a four-layer structured neural network, including an input layer, an output layer, and two hidden layers, is used in this work. As with XGBoost, the appropriate number of hidden units in each hidden layer is optimized using the stochastic search module [24] in the Sklearn package. The MLPregressor module in Sklearn can optimize the quadratic loss using LBFGS [25] or the stochastic gradient descent algorithm [26]. The solver used for optimization in this model is the Adam solver. The activation function of the hidden layer is Relu. The structure of the neural network is shown in the following figure.

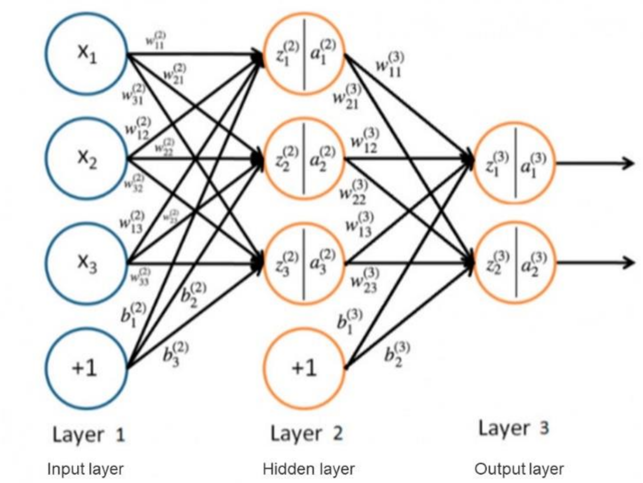

Figure 6. The design of neural network.

3. Result and discussion

I used XGBoost and neural network models for data analysis based on the MD simulation results of the solid-solid and solid-liquid interfaces. All the hyperparameter were adjusted and K-fold cross-validation was employed. The performance was evaluated by using mean square error (MSE). The best results of the solid-solid interface system were obtained with the XGBoost model with MSE of 30.167, while that of the solid-liquid interface were achieved by using a neural network with MSE of 10.128. The detailed results are presented in the following table.

Table 2. MSE of different machine learning models.

Interface	MSE	Method
Solid-Solid	30.167	XGBoost
Solid-Liquid	10.128	Neural network

The results are further compared in the following two figures:

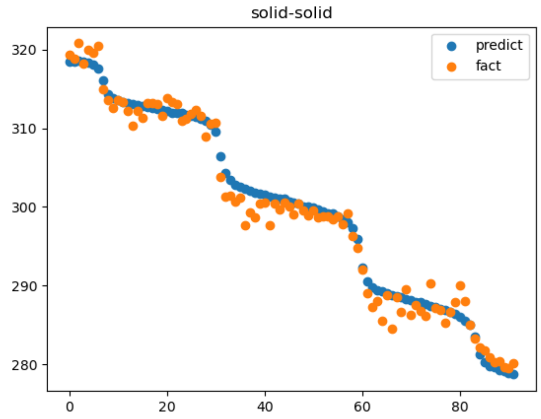

Figure 7. Temperature variation in the model of solid-solid interface.

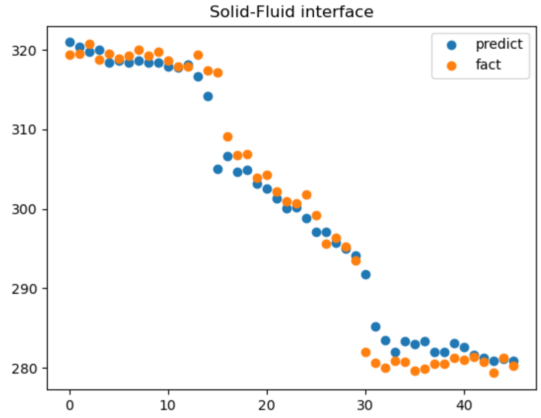

Figure 8. Temperature variation in the model of solid-fluid interface.

It's obvious that, the machine learning model can well describe the temperature variation in the region of relatively stable temperature. However, the prediction results perform poorly in the region of sharp temperature changes, i.e., at the material interface. The performance of the prediction for the region close to the interface is apparently worse than that for the bulk region. The possible reasons include 1) the dataset is not large enough to contain all the information of the model to describe the uncertainty of the model, and 2) in XGBoost, the random search results require different optimal parameters for the prediction model in different regions, but in this work, the machine learning process uses the same hyperparameters for all regions considering the limitations of the computational consumption, resulting in overfitting or underfitting specific regions. The performance is significantly improved if the points at the interface are removed. However, simulating the temperature change at the interface is critical in this work, thus in future, I have to further investigate how to make the model more accurate in predicting the temperature change at the interface. Further, I analyzed the feature importance in the XGBoost model to study the contribution of each parameter. Figures 9 and 10 show that both features selected are important for predicting the temperature variations. For the model of solid-solid interface, the lattice constant of the system is more important than the atomic mass; whereas for the model of the solid-fluid interface, the density of fluid particle is more critical.

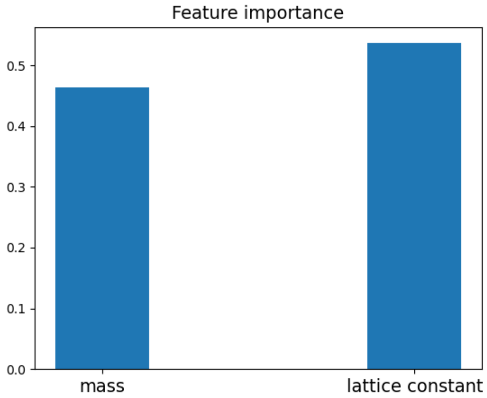

Figure 9. The key features of the model of solid-solid interface.

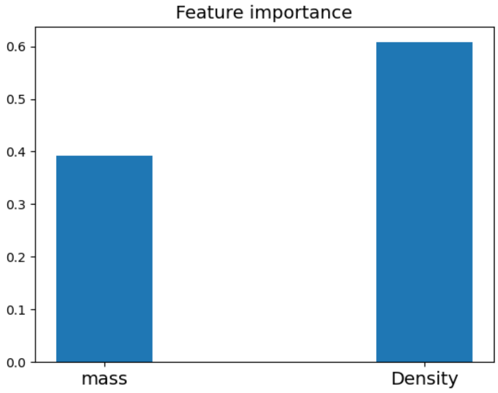

Figure 10. The key features of the model of solid-fluid interface.

4. Conclusion

I combined classical molecular dynamics simulations and novel machine learning methods to study thermal interface materials. Datasets are obtained from molecular dynamics simulations, fitted using neural networks and XGBoost models of machine learning. Then the fitted models are cross validated with molecular dynamics simulation results to demonstrate that machine learning methods can improve the efficiency of material property prediction. According to the results, the models fitted by machine learning can indeed predict the thermophysical properties of different materials more accurately. The fitted results from both XGBoost and neural network models are in good consistent with the molecular dynamics predictions. In summary, I tremendously improved the efficiency of predicting and screening the thermophysical properties of materials by combining machine learning with molecular dynamics simulations, thus developing a new way of screening the thermal interface materials in future.

References

[1] B. J. Alder and T. E. Wainwright, “Phase Transition for a Hard Sphere System,” J. Chem. Phys., 1957, 27, 5, 1208. https://doi.org/10.1063/1.1743957

[2] A. Rahman, “Correlations in the Motion of Atoms in Liquid Argon,” Phys. Rev., 1964, 136, A405 https://doi.org/10.1103/PhysRev.136.A405

[3] Michael P. Allen, “Introduction to Molecular Dynamics Simulation,” in Computational Soft Matter: From Synthetic Polymers to Proteins, NIC Series vol. 23, N. Attig, K. Binder, H. Grubmüller, and Kurt Kremer Eds. Jülich, German, NIC-Directors, pp.1-28, 2004

[4] Alireza Moradzadeh and Narayana R. Aluru, “Transfer-Learning-Based Coarse-Graining Method for Simple Fluids: Toward Deep Inverse Liquid-State Theory,” J. Phys. Chem. Lett., 2019/10/6, 1242–1250. https://doi.org/10.1021/acs.jpclett.8b03872

[5] Introduction to Machine Learning March 2018 Publisher: China Machine Press ISBN: 9787111548683 Project: machine learning Authors: Yong Wang Guoqiang Zhong

[6] Molecular dynamics simulation and nanoscale heat transfer (1): an introduction to lammps simulation software. Shanghai 2018-03-10

[7] A magnetic velocity Verlet method December 2020American Journal of Physics 88(12):1075-1082DOI:10.1119/10.0001876 Authors: A. Chambliss J. Franklin

[8] Engineering Thermodynamics January 2011 Publisher: Huazhong University of Science and Technology Press ISBN: 7560972470 Author: Huang Xiaoming Liu Zhichun Fan Aiwu

[9] Electromagnetic Field Boundary Conditions for Moving Boundaries January 2002 Acta Physica Sinica-Chinese Edition-51 (10): 2202DOI: 10.7498/aps.51.2202Authors: Liang Hong-Chang Zhu Qing-Xin

[10] Thermal wave heat transfer in a superfluid helium bath Chinese Journal of Physics, January 2002, 51 (6): 1350 DOI: 10.7498/aps.51.1350 by Zhang Peng Wang Ruzu Masahide Murakami

[11] Application of Tersoff-Brenner Potential Function in the Study of Carbon Nanotubes. Li Feng, Bai Shuo, Zheng Hong, Zhou Longguang, Cheng Huiming.

[12] Limitations of the Derjaguin Approximation and the Lorentz-Berthelot Mixing Rule February 2010Langmuir26(7):4555-8 Source PubMed Authors: Jan Forsman Clifford Woodward

[13] High Temperature Semiconductor Materials March 1964 Chinese Journal 15 (3): 231-236DOI: 10.1360/csb1964-9-3-231Authors: Lan Yinglin

[14] CharacteristicCurvesoftheLennardJonesFluidApril2022DOI:10.31219/osf.io/p32yz Authors: Simon Stephan Ulrich Deiters

[15] Global Convergence of the Hestenes-Stiefel Conjugate Gradient Method December 1995 Chinese Version 40 (23): 2113-2117DOI: 10.1360/csb1995-40-23-2113 Author: Guo Wenying

[16] Introduction to Machine Learning March 2018 Publisher: China Machine Press ISBN: 9787111548683 Authors: Yong Wang Ocean University of China Guoqiang Zhong Ocean University of China Xin Sun

[17] XGBoost: A Deep into Boosting February 2020 by Shubam Malik Simon Fraser University Rohan Harold Simon Fraser University Akash Singh Chitkara University

[18] Decision Tree Algorithm Based on Space Optimization January 2007 Authors: Dong Xiaoming, Lin Chuang, Chen Zhen

[19] Design of short-term electricity load forecasting system based on gradient lifting tree April 2021 DOI: 10. 14022/j.issn1674-6236.2020.22.004Authors: Henan Science and Technology

[20] Neurons and Neural Network Models November 1992 Authors: Zheng Degao

[21] Neural Networks June 2022 DOI:10.1201/9781003305835-10 Authors: Paul Roberts A Note on the Approximation Theorem of A. Cellina Acta Sinica Feb. 1997 Wu Congxin, Li Lei, Wang Xiaomin

[22] A Note on the Approximation Theorem of A. Cellina Acta Sinica Feb. 1997 Wu Congxin, Li Lei, Wang Xiaomin

[23] A Survey of Convolutional Neural Networks and Their Applications in Intelligent Transportation Systems September 2021 DOI: 10. 19818/J. CNKI. 1671-1637.2021.04.003 Projects: Neural

Networks Principle & Application Computer Vision Authors: yingjie Mashisheng chengyunting maYide Ma

[24] Multiple linear regression using sklearn January 2022 by Dee Sandyman

[25] A Note on the Approximation Theorem of A. Cellina Acta Sinica Feb. 1997 Wu Congxin, Li Lei, Wang Xiaomin

[26] Randomized search algorithm December 1990 Acta Sinica 35 (24): 1905-1906 DOI: 10.1360/csb1990-35-24-1905 Author: Zhou Jianqin