ISSN Online: 2821-1693

Transactions on Machine Intelligence

Journal Homepage: https://tmachineintelligence.ir/index.php/TMI

Phase Transition of the Two-Dimensional Ising Model in a Homogeneous Magnetic Field Using the Metropolis Monte Carlo Algorithm and Separation of Different Phases via CNN

R. Rastgar Poor[1], N. Majd[2,*]

[1] School of Engineering Sciences, College of Engineering, University of Tehran, Tehran, Iran
[2] Assistant Professor, Algorithm and Computation Group, School of Engineering Sciences, College of Engineering, University of Tehran, Tehran, Iran

ARTICLE INFO

Article History:
Received 3 December 2024
Received in revised form 10 January 2025
Accepted 17 February 2025
Available online 7 March 2025

Keywords:
Quantum spin networks, Hamiltonian, deep learning, Convolutional Neural Network (CNN), two-dimensional Ising model.

ABSTRACT

Quantum spin networks represent configurations of spins arranged on a topological lattice, where the spin interactions are governed by the system's Hamiltonian. These networks are critical for understanding magnetic materials, as the arrangement of spins and the type of interaction between neighboring spins determine the macroscopic behavior of the system. The behavior of these systems is further influenced by the presence of external magnetic fields. In this paper, we first investigate the various phases of the two-dimensional Ising lattice with periodic boundary conditions under the influence of a uniform external magnetic field. The exploration of these phases is performed using the Metropolis Monte Carlo (MP-MN) algorithm, a well-established statistical method for simulating spin systems. Subsequently, we explore the potential of deep learning, specifically convolutional neural networks (CNN), in identifying and predicting these phases of spin lattices. The CNN's ability to classify different phases of the two-dimensional Ising model in the presence of a homogeneous magnetic field at a constant temperature is examined. The study aims to demonstrate how machine learning models, particularly CNNs, can effectively detect phase transitions and predict the system's behavior, which traditionally requires extensive computational methods. Finally, the performance of the CNN algorithm is evaluated by assessing its accuracy in predicting different phases of the Ising model.

1. INTRODUCTION

Spin networks are generally considered one of the fundamental and popular classes in machine learning. This paper aims to introduce the structure of the two-dimensional Ising model on a square lattice with periodic boundary conditions in a uniform magnetic field. Our goal in this paper is to investigate the behavior of the magnetic field in different phases of the model under consideration [1-11].

[*] Corresponding Author: naymajd@ut.ac.ir
Assistant Professor, Algorithm and Computation Group, School of Engineering Sciences, College of Engineering, University of Tehran, Tehran, Iran

http://dx.doi.org/10.47176/TMI.2025.17

The Ising model is a mathematical model of ferromagnetism in statistical mechanics. This model consists of discrete variables representing the magnetic dipole moments, "spins," of atoms, which can take one of two states (+1 or -1). The spins are placed on a two-dimensional lattice with periodic boundary conditions. Each spin interacts with its neighbors. The physical behavior of the spin system tends to minimize its energy state. However, heat and temperature disrupt this tendency and can create the possibility of various structural phases. The two-dimensional Ising model on a square lattice is one of the simplest statistical models to demonstrate phase transitions.

Machine learning can be considered a subset of artificial intelligence [12]. It primarily involves supervised learning, unsupervised learning, and reinforcement learning [13]. In this paper, our focus is on supervised learning. Supervised learning mainly consists of regression models, classification models, and neural networks (NN).

In this paper, we employ the deep neural network method, CNN, for classifying the Ising spin lattice in the presence of an external magnetic field and its effects on the spins in the network. This paper is structured as follows:

- Introduction to the Ising model with its macroscopic parameters
- Phase transition of the two-dimensional Ising model in the presence of a magnetic field
- Implementation of the MP-MN algorithm to obtain the ground state configurations of the Ising model
- Results of MP-MN simulations
- Description of CNN neural network
- Conclusion and simulation results

2. TWO-DIMENSIONAL ISING MODEL ON A SQUARE LATTICE

Just as the Turing machine is fundamental to classical computers, the Ising model is the simplest model for quantum computers. The Hamiltonian of a system represents the total energy of the system under consideration.

$$\mathrm{E(S)} = -J \sum_{<i,j>}^{N} s_i s_j \; - h \sum_{i}^{N} s_i \tag{1}$$

where s_i and s_j are the spins of particles i and j, taking values of +1 or -1. h is the external magnetic field, and J represents the exchange interaction strength between the particles. In the classical view, the Hamiltonian is understood as the sum of the kinetic and potential energies. For the simple Ising model, the Hamiltonian is defined as follows:

$$\mathrm{H(s)} = -\sum_{<i,j>}^{N} J_{i,j}\, s_i s_j \; - h \sum_{i}^{N} s_i \tag{2}$$

In this relation, $s_i = \pm 1$ is the classical spin operator of 1/2. The first summation operation is performed over the nearest neighbors <i,j>, and the second summation is over the lattice vertices. Additionally, $J_{i,j} = 1$ represents the exchange interaction strength between the two particles i and j. The phase transition in this model, due to its simplicity, is an appropriate example for training a neural network. In section 6, we will focus on the training of the neural network.

3. PHASE TRANSITION OF THE TWO-DIMENSIONAL ISING MODEL IN A MAGNETIC FIELD

The two-dimensional Ising model, in the absence of an external magnetic field, experiences a phase transition at a critical temperature of 2.26 Kelvin, which is referred to as the critical temperature. The phase transition point for this model is written based on equation 3.

$$T_c = \frac{2J}{\ln(1+\sqrt{2})} \tag{3}$$

To observe the phase transition in the presence of an external magnetic field, we examine the magnetization (M) when the external field varies at a constant temperature (T). Figure 1 illustrates the magnetization of the ferromagnetic atoms in a spin lattice under an external magnetic field (H) ranging from -4 to +4. The temperature is considered constant and below the critical temperature (β_c). Based on the magnetization size in the external field, the lattice is classified into four different classes, which are distinguished by four colors in Figures 1 and 2.

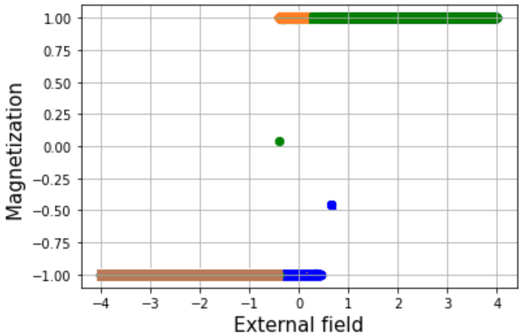

Fig. 1. Magnetization M of N ferromagnetic atoms as a function of the external magnetic field at a constant temperature where $\beta < \beta_c$.

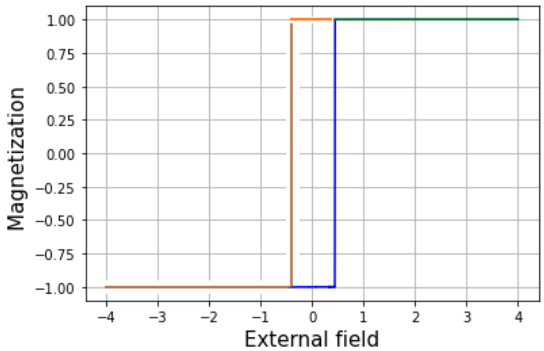

Fig. 2. Magnetization M of N ferromagnetic atoms as a function of the external magnetic field at a constant temperature where $\beta < \beta_c$.

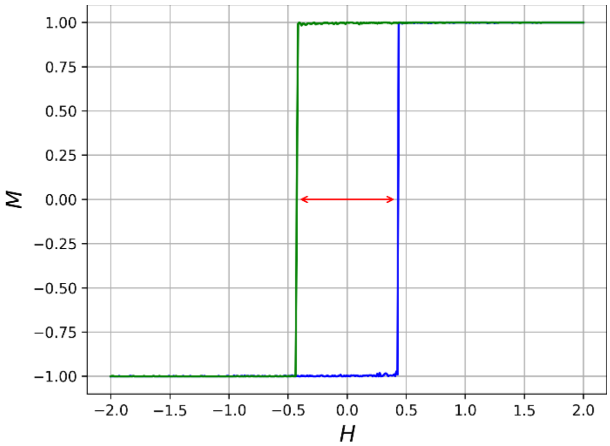

Fig. 3. The distance ΔH in the phase transition diagram is indicated by the red arrow.

The distance between the two lines in the phase transition diagram, as shown in Figure 3, is denoted as ΔH. This distance changes at different temperatures. In Sections 5 and 6, the relationship between ΔH and temperature is examined, and we plot the graph of ΔH as a function of temperature for three networks with different dimensions.

4. IMPLEMENTATION OF THE MP-MN ALGORITHM TO OBTAIN THE GROUND STATE OF THE ISING MODEL

We are seeking a state of the Ising lattice that has the least energy and has reached equilibrium; for this purpose, we use the MP-MN algorithm. In this algorithm, at each iteration, a particle is randomly selected, and the system's energy is calculated. To calculate this, it is sufficient to compute the energy of the selected particle in relation to its neighboring particles. The spin of the selected particle is flipped, and the energy is recalculated. The state with the lowest energy is considered favorable. This process continues until the network reaches equilibrium. The flowchart of the MP-MN algorithm is shown in Figure 4.

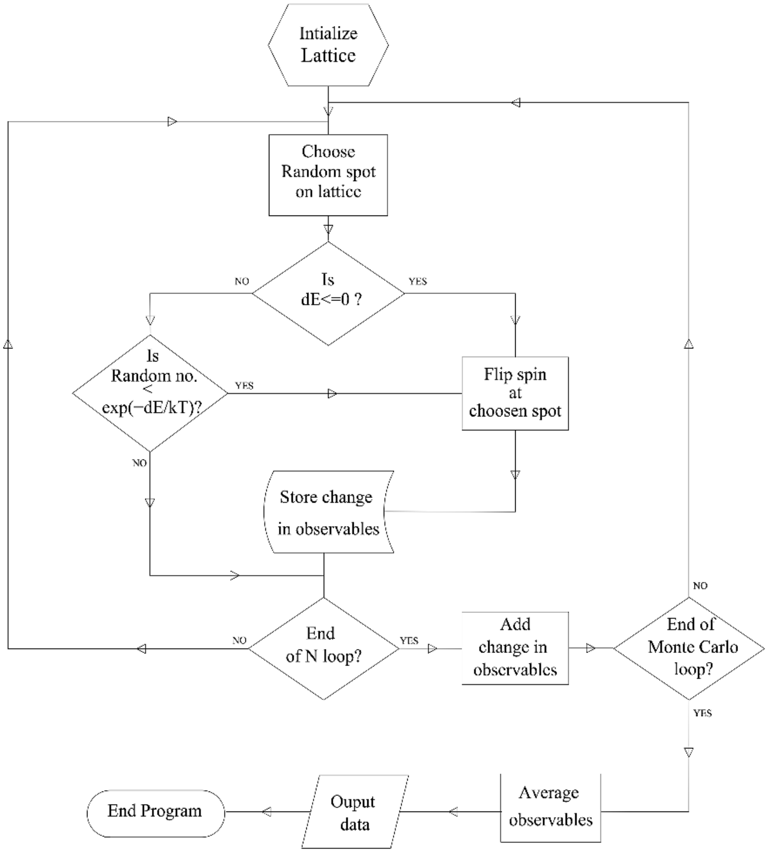

Fig. 4. Flowchart of the MP-MN algorithm.

After the network reaches equilibrium, we calculate the magnetization of the N×N lattice. This is performed for three lattices with dimensions of 8×8, 10×10, and 15×15 at a temperature β and an external magnetic field H. β_c and H_c represent the critical temperature and critical field, respectively. The number of steps in the algorithm, as shown in Table 1, is 4000 steps to reach equilibrium and 3000 steps during the MP-MN execution. For each lattice, the minimum value of ΔH is also calculated.

Table 1. Calculation of magnetization for three N×N lattices at different temperatures and external magnetic fields, and the minimum value of ΔH.

N	Number of Steps in MP-MN Execution	Number of Steps to Reach Equilibrium	Min ΔH	$\beta < \beta_c$ $H < H_c$	$\beta < \beta_c$ $H > H_c$	$\beta > \beta_c$ $H < H_c$	$\beta > \beta_c$ $H > H_c$
8	3000	4000	0.02706766917	-1	1	-0.95416666	0.975
10	3000	4000	0.08120300752	-1	1	-0.96266666	0.96266666
15	3000	4000	0.08320802005	-1	1	-0.96503703	0.96385185

5. MP-MN SIMULATION RESULTS

When we plot the phase transition diagram for a lattice at different temperatures, we observe that as the temperature approaches the critical temperature, ΔH decreases. For each of the three different lattice dimensions, we randomly generate 10 lattices at a specified temperature and calculate ΔH in their phase transition. This process is repeated at temperatures of 1.2, 1.4, 1.6, 1.8, 2.0, 2.2, and 2.4. As a result, ΔH is calculated 10 times at each temperature β. We then plot the average ΔH values against temperature, as shown in Figure 5. The blue curve in Figure 5 represents the average of 10 ΔH plots as a function of temperature for the 8×8 lattice. Similarly, the orange and green curves are plotted for the 10×10 and 15×15 lattices. As expected, ΔH decreases with increasing temperature until it reaches the critical temperature.

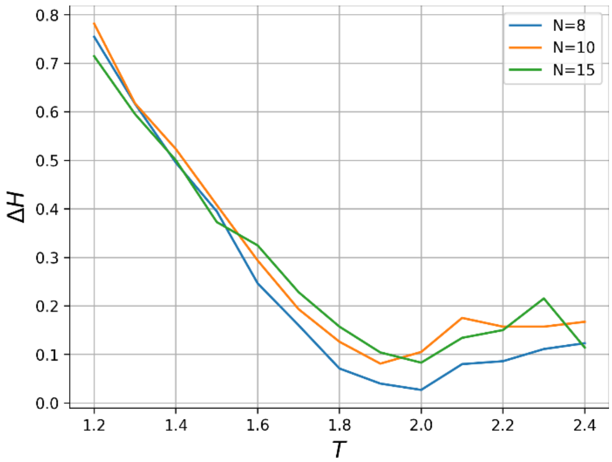

Fig. 5. Average ΔH as a function of temperature for three N×N lattices.

6. CNN METHOD

In the field of deep learning, Convolutional Neural Networks (CNN) are the most popular and widely used algorithm. The main advantage of CNN is that it automatically identifies relevant features without the need for human supervision. CNNs are widely applied across a variety of fields, including computer vision, speech processing, facial recognition, and more. The structure of CNN is inspired by the neurons of the human and animal brain, similar to a conventional neural network. A CNN structure consists of several layers. Below, we describe each layer in the CNN structure along with its function.

- **Convolutional Layer**: The convolutional layer is the most important component in the CNN structure. This layer contains a set of convolutional filters (also called kernels). The input image, represented as N-dimensional metrics, is convolved with these filters to produce the output feature map.

- **Pooling Layer**: The primary task of the pooling layer is subsampling the feature maps that are produced after the convolution operation. In other words, this approach reduces the size of large feature maps to smaller ones while preserving dominant information or features at each pooling stage. Similar to the convolutional operation, the step size and kernel are predetermined before pooling. Various pooling methods are available for use in different layers, including tree pooling, gated pooling, average pooling, minimum pooling, maximum pooling, global average pooling (GAP), and global maximum pooling. The most common and widely used pooling methods are maximum pooling, minimum pooling, and GAP.

- **Activation Function**: The mapping from input to output is the primary function of different activation functions in various neural networks. The input value is determined by calculating the weighted sum of the neuron's input along with its bias (if any). This means that the activation function determines whether a neuron should be activated based on a specific input and generates the corresponding output.

- **Fully Connected Layer (FC)**: This layer usually appears at the end of the CNN structure. In this layer, each neuron is connected to all neurons in the previous layer, which is known as the fully connected (FC) approach. This layer acts as the classifier of the CNN. The input to the FC layer comes from the last pooling or convolutional layer. This input is in the form of a vector, created after flattening the feature maps. The output of the FC layer represents the final output of the CNN.

- **Loss Functions**: The previous section presented various layers of the CNN structure. Additionally, the final classification is obtained from the output layer, which is the last layer of the CNN structure. Some cost functions in the output layer are used to compute the error in the predicted samples in the CNN model. This error represents the difference between the actual and predicted outputs. This error is then optimized through the learning process of the CNN [14-16].

The neural network, without knowledge of the system's Hamiltonian, can learn certain features of the system and classify each network into one of the four classes. For simulating the neural network (CNN), we first generate data samples for training and testing the network. Due to time complexity, we had to select a small square lattice of size L×L=20×20 with 400 spins. For each of these lattices at a constant temperature, we use an external magnetic field ranging from +4 to -4 for 5000 samples and then from -4 to +4 for another 5000 samples. In total, 10,000 data samples are available as the learning and testing set. We know that the samples are classified into four classes based on their spin configuration. However, the network needs to be trained to successfully classify them. This classification is performed using a CNN neural network with Python programming. The architecture of the model is shown in Figure 6.

```
#----Creating CNN model-----

from keras.models import Model
import keras
from keras import layers
from keras.optimizers import adam_v2
from keras.optimizers import gradient_descent_v2
from keras.losses import categorical_crossentropy
sgd = gradient_descent_v2.SGD()
#-------------------Architecture of model----------------
model = keras.Sequential(
  [
    keras.Input(shape=(20,20,1)),
    layers.Conv2D(64, kernel_size=(3, 3), activation="relu", padding='same'),
    layers.MaxPooling2D(pool_size=(2, 2)),
    layers.Conv2D(128, kernel_size=(3, 3), activation="relu", padding='same'),
    layers.MaxPooling2D(pool_size=(2, 2)),
    layers.Flatten(),
    layers.Dense(100, activation='relu'),
    layers.Dropout(0.1),
    layers.Dense(4, activation='sigmoid')
```

```
    ]
)
#---------------------------------------------------
model.summary()
batch_size = 128
epochs = 200
model.compile(loss="categorical_crossentropy", optimizer="adam", metrics=["accuracy"])
# Train CNN model
network_history =model.fit(train_x, train_y, batch_size=batch_size, epochs=epochs, validation_split=0.1)
```

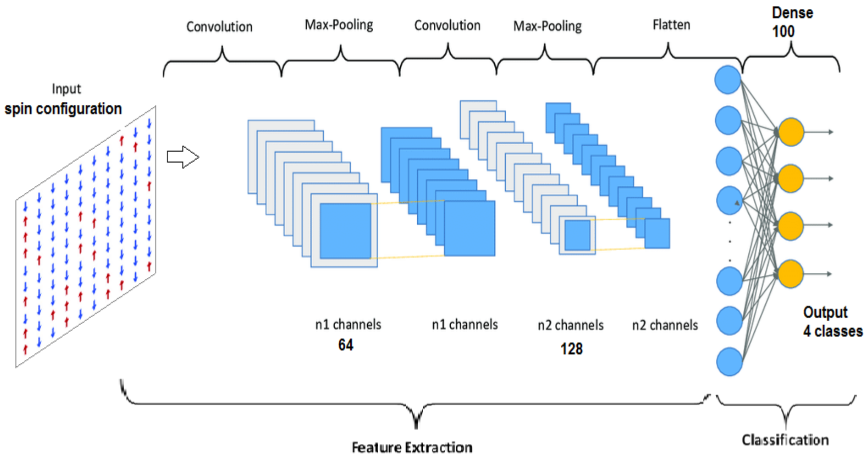

Fig. 6. Architecture of the CNN model.

7. CNN SIMULATION RESULTS

To evaluate the CNN model, we plot the cost and accuracy for 500 epochs. As shown in Figures 3 and 4, the cost or error on the test data decreases compared to the training data. Additionally, the accuracy on the test data is higher than the accuracy on the training data. As a result, the model we have trained is able to classify the system effectively, even without knowledge of the system's Hamiltonian.

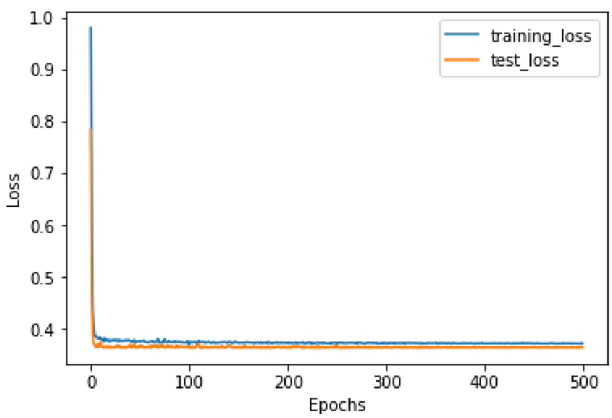

Fig. 7. Error plot as a function of the number of Epochs for training and test data.

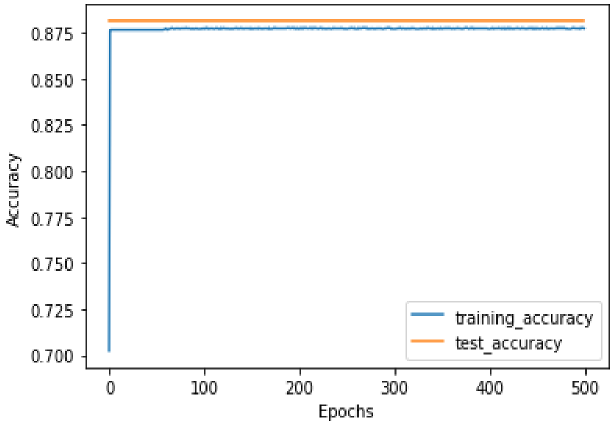

Fig. 8. Accuracy plot as a function of the number of Epochs for training and test data.

8. CONCLUSION

In this study, we successfully demonstrated the application of deep learning, specifically Convolutional Neural Networks (CNN), for classifying and predicting the phases of a two-dimensional Ising model in the presence of a uniform external magnetic field. The CNN model was trained using data generated by the Metropolis Monte Carlo (MP-MN) algorithm, which simulated the behavior of the system under various conditions. The results indicate that the CNN model was highly effective in detecting phase transitions and classifying the system into different phases, even without explicit knowledge of the system's Hamiltonian. Additionally, the model achieved superior accuracy on test data compared to training data, highlighting its ability to generalize well and make reliable predictions. This suggests that CNNs can be a powerful tool for solving complex problems in statistical mechanics and phase transition analysis, offering a promising alternative to traditional computational methods that often require extensive resources. Future work could focus on refining the CNN model to handle more complex systems and explore its applicability in other domains of physics and material science.

Declaration

We acknowledge that we used ChatGPT to enhance the academic writing of our manuscript while ensuring the originality and integrity of our work.

Transparency Statement

The data supporting this study are available upon reasonable request to the corresponding author, subject to ethical and confidentiality considerations.

Acknowledgments

We would like to express our gratitude to all individuals who contributed to this project.

Declaration of Interest

The authors declare that they have no competing interests.

Funding

This research received no specific grant from any funding agency, commercial, or not-for-profit sectors.

REFERENCES

[1] Huang, K. (1987). Statistical mechanics (2nd ed.). Wiley.

[2] Reichl, L. E. (2016). A modern course in statistical physics (4th ed.). Wiley-VCH.

[3] Landau, L. D., & Lifshitz, E. M. (1980). Statistical physics (3rd ed., Vol. 5). Pergamon Press.

[4] Pathria, R. K., & Beale, P. D. (2011). Statistical mechanics (3rd ed.). Elsevier.

[5] Binder, K., & Heermann, D. W. (1997). Monte Carlo simulation in statistical physics: An introduction (2nd ed.). Springer. https://doi.org/10.1007/978-3-662-03336-4

[6] Hu, W., Singh, R. R. P., & Scalettar, R. T. (2017). Discovering phases, phase transitions, and crossovers through unsupervised machine learning: A critical examination. Physical Review E, 95(6), 062122. https://doi.org/10.1103/PhysRevE.95.062122

[7] Avni, Y., Fruchart, M., Martin, D., Seara, D. S., & Vitelli, V. (2023). Nonreciprocal Ising model. Physical Review Letters, 134(11), 117103. https://doi.org/10.1103/PhysRevLett.134.117103

[8] Alhabeb, M., Maleski, K., Anasori, B., Lelyukh, P., Clark, L., Sin, S., & Gogotsi, Y. (2017). Guidelines for synthesis and processing of two-dimensional titanium carbide (Ti_3C_2Tx MXene). Chemistry of Materials, 29(18), 7633–7644. https://doi.org/10.1021/acs.chemmater.7b02847

[9] Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2019). A survey on bias and fairness in machine learning. ACM Computing Surveys (CSUR), 54(6), Article 115. https://doi.org/10.1145/3457607

[10] Altaisky, M. V. (2023). Poincaré group spin networks. International Journal of Theoretical Physics, 62(6), 2151–2163. https://doi.org/10.1007/s10773-024-05688-7

[11] He, W., Zhang, J., Dieckhöfer, S., Varhade, S., Brix, A. C., Lielpetere, A., Seisel, S., Junqueira, J., & Schuhmann, W. (2022). Splicing the active phases of copper/cobalt-based catalysts achieves high-rate tandem electroreduction of nitrate to ammonia. Nature Communications, 13(1), Article 2878. https://doi.org/10.1038/s41467-022-28728-4

[12] Xiao, H., Rasul, K., & Vollgraf, R. (2017). Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. ArXiv. https://arxiv.org/abs/1708.07747

[13] Hu, W., Singh, R. R. P., & Scalettar, R. T. (2017). Discovering phases, phase transitions, and crossovers through unsupervised machine learning: A critical examination. Physical Review E, 95(6), 062122. https://doi.org/10.1103/PhysRevE.95.062122

[14] Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A. Y., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al Amidie, M., & Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. Journal of Big Data, 8(1), Article 53. https://doi.org/10.1186/s40537-021-00444-8

[15] He, K., Gkioxari, G., Dollár, P., & Girshick, R. B. (2017). Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (pp. 2961–2969). https://doi.org/10.1109/ICCV.2017.322