Conference on Modelling Fluid Flow (CMFF'22)
The 18th International Conference on Fluid Flow Technologies
Budapest, Hungary, August 30-September 2, 2022

Classification of Percolation Clusters with Artificial Neural Networks

Kálmán Kustány[1], Gergely Hajgató[2], Bendegúz Dezső Bak[3], Tamás Kalmár-Nagy[4]

[1] Corresponding Author. Department of Fluid Mechanics, Faculty of Mechanical Engineering, Budapest University of Technology and Economics. Bertalan Lajos u. 4 - 6, H-1111 Budapest, Hungary. Tel.: +36 1 463 2464, Fax: +36 1 463 3464, E-mail: kalman.kustany96@edu.bme.hu
[2] Department of Telecommunications and Media Informatics, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics. E-mail: ghajgato@tmit.bme.hu
[3] Department of Fluid Mechanics, Faculty of Mechanical Engineering, Budapest University of Technology and Economics. E-mail: bak@ara.bme.hu
[4] Department of Fluid Mechanics, Faculty of Mechanical Engineering, Budapest University of Technology and Economics.

ABSTRACT

The classification capabilities of artificial neural networks (ANNs) were examined in the study focusing on recognizing 2-dimensional percolating lattices. The performance of a classifier based on a multi-layer perceptron (MLP) was compared to a classifier based on a convolutional neural network (CNN). A large number of lattices were generated with different occupation probabilities from 0 to 1 to form a dataset that was split in three different ways to find out the possible difficulties of the learning.

While the classification recall was above 77% for each ANN in each case, significant differences were seen when the data were imbalanced. In such cases, the CNN-based classifier clearly outperformed the one based on MLP in recognizing the lattices of the underrepresented kind.

When the ANNs were trained on 4397 samples drawn randomly uniform from the full dataset, both of the networks achieved a similar recall being the CNN the better with a recall of ≈ 0.93. Moreover, the CNN-based classifier achieved a higher recall in general in all three cases suggesting its superiority compared to the MLP for recognizing percolating lattices.

Keywords: artificial neural networks, convolutional neural network, machine learning, multilayer perceptron, percolation, supervised learning

1. INTRODUCTION

The development and usability of machine learning (ML) algorithms went through a massive upsurge in the last decades due to the increasing computational capacity and the rapidly growing amount of available data. The main tools can be roughly divided into three large groups: supervised, unsupervised, and reinforcement learning. Supervised techniques fit labeled datasets while unsupervised methods are able to learn useful features in the absence of labels. Reinforcement learning is based on agent-environment interaction which is useful for learning control policies or to learn from such datasets where the loss function cannot be described by an exact mathematical formula.

Artificial neural networks (ANNs) are widely used for classification due to the fact that they are universal function approximators [1]. Owing to this property, ANNs were successfully applied in the field of computer vision, natural language processing, recommendation systems, etc. Besides these, ANNs have also been adapted in science and engineering – such as in this case – for the classification of percolating lattices [2, 3, 4, 5].

Percolation theory is applied in the field of mercury porosimetry (among others) that was investigated by Bak and Kalmár-Nagy in [6]. Mercury porosimetry is applied to specify the pore size distribution of rock samples primarily in the oil industry. During the process, the mercury is forced into the sample with constantly increasing pressure while the volume of the injected mercury is measured vs. the applied pressure, which is the saturation curve. In practice, this curve is assumed to be directly related to the cumulative pore size distribution. On the other hand, the real distribution of the pore size does not coincide with the cumulative pore size distribution because of the "non-accessibility" of the pores at a given pressure. The goal of the study by Bak and Kalmár-Nagy was to determine a more accurate cumulative pore size distribution which was achieved by treating the mercury propagation as a percolation process. Their results showed a good agreement between the experimental saturation curve and those

obtained from their method [6, 7, 8].

Bottiglione et al. [9] used percolation theory and contact mechanics to investigate fluid leakage through seals. They observed that it is not enough to have a partial connection between the sealing surfaces (because of the surface roughness) but a channel has to be also formed from the previously independent chambers to have leakage. This phenomenon can be effectively described by percolation theory.

Zhao et al. [10] studied the conductivity of liquid metal embedded elastomers (LMEEs). These elastomers have attracted interest from researchers due to their low elastic modulus, tunable electrical and/or thermal conductivity, and high flexibility. One impressive feature of LMEEs is that their electrical resistance stays constant when stretched, which means that the resistance is no longer proportional to the length of the conductive material if only a moderate length change occurs. This feature makes LMEEs desirable in real-world applications. During the research, finite element analysis was carried out to examine the electromechanical coupling response of a percolated liquid droplet. After that, they generalized the results to a network of LMEE droplets.

Salt caverns are a promising opportunity to store renewably produced hydrogen underground to reduce CO2 emissions by enhancing the utilization of volatile renewable energy sources. Rock salt is able to reduce the infiltration of hydrogen into the cavern walls due to its low permeability. Nevertheless, the pressure-driven percolation effect may form discrete pathways for the fluid through the walls which cause damage in the cavern. Zill et al. [11] created a model which shows good agreement with the pressure-driven percolation of hydrofracturing experiments.

Based on the researches mentioned above, the advantage of training of neural networks to learn percolation is unambiguous. Here we focus on using two types of neural nets, the multi-layer-perceptron (MLP) and the convolutional neural network (CNN) with approximately the same number of trainable parameters, train them for the classification of percolation lattices, and compare the accuracies of these nets. After that, we investigate the effect on the accuracy of different training datasets.

The outline of this paper is as follows. In Sec. 2 we introduce the basic concept of percolation theory and the algorithm we used to generate the percolating lattices for classification. Sec. 3 contains the general background for those kind of artificial neural networks that were used in this study. After that, we describe the training dataset and the parameters of the neural nets in Sec. 4. Sec. 5 and Sec. 6 contain the results and the conclusion, respectively.

2. PERCOLATION THEORY

Percolation theory and percolation models have been developed for a differing range of applications such as fractal diffusion, the conductivity of semiconductors, or modeling fluid propagation through a porous medium. Percolation theory studies the properties of lattice clusters. The simplest problems in percolation theory are site-percolation and bond-percolation whose schematics can be seen in Fig. 1.

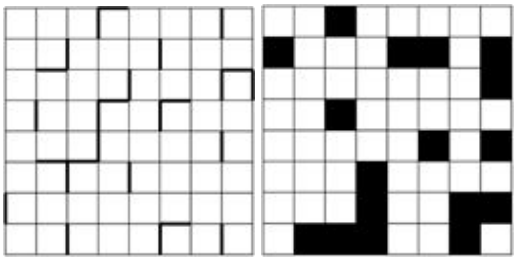


Figure 1. Bond percolation (left) and site percolation (right).

These percolation models deal with lattices in which a site/bond can be occupied with probability p (occupation probability). A group of neighboring occupied sites/bonds is called a cluster. A cluster that connects the top and bottom sides of the lattice is called spanning cluster. The existence of a spanning cluster means that the modeled fluid percolated through the lattice. Fundamental questions in percolation theory is whether a spanning cluster exists and how the existence of spanning clusters is dependent on the occupation probability p. There is a percolation threshold (p_c) which refers to the critical occupation probability. Under this value, the probability that a spanning cluster exists is zero in infinite lattices. On the other hand, for $p > p_c$, the probability that a spanning cluster exists is one. Thus, percolation theory can be thought of as the study of a geometrical phase transition, since at a critical fraction of addition the network of small, disconnected clusters merge into a spanning cluster (visual representation can be seen in Fig. 2).

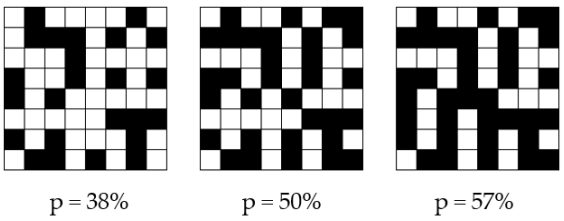


Figure 2. Lattices with different occupation probabilities.

3. ARTIFICIAL NEURAL NETWORKS

Feed-forward artificial neural networks stem from biological neural networks, but the similarities should be examined only from a sufficiently distant view. As it was mentioned before, neural nets are widely used due to their flexibility which ensures the fact that the architecture can be constructed from dif-

ferent types of building blocks. In this research, the multi-layer perceptron and the convolutional layer are used in the network architecture and the theory behind these building blocks is detailed here.

3.1. Multi-layer perceptron

The basic calculation unit of such a network is the perceptron which receives different signals on its input, multiplies each input with a corresponding weight, sums them up, and applies a nonlinear function on the summation. The nonlinear function is called activation function in the machine learning terminology and its output is the output of the perceptron. The weights quantify the importance of the specific input values regarding to the output and the activation function ensures that the summed output of multiple perceptrons can mimick the output of general functions. The schematic of a standalone perceptron is shown in Fig. 3.

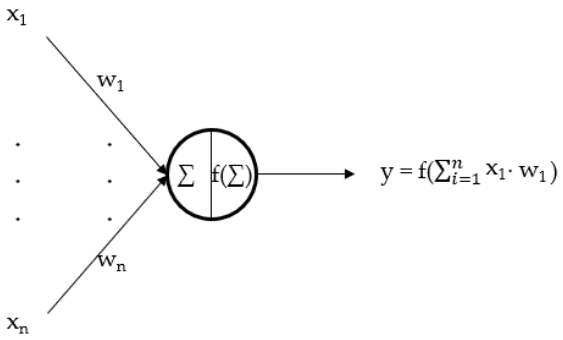


Figure 3. Schematic representation of a perceptron.

In order to enable the higher level feature and pattern recognition with a neural network, the perceptrons are arranged into layers and layers are stacked into a network which is called multi-layer perceptron. Such a network is shown in Fig. 4.

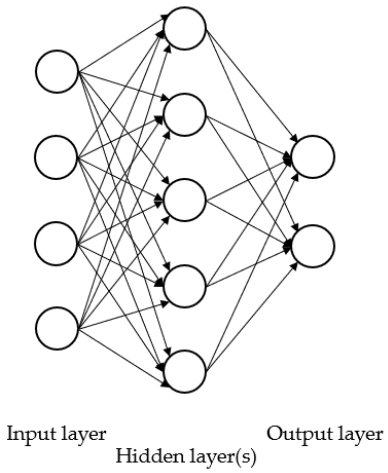


Figure 4. Example of a multi-layer perceptron.

With this arrangement, the network is able to learn abstract features from the data, which is desirable in real-world applications. The first layer of a network is the input layer, thereafter come the so-called hidden layers and at the end of the network is the output layer. If every neuron inside a specific layer is connected to every neuron in the previous layer (in other words: if the output signal of the neurons in the preceding layer is fed to the input of all the neurons in the actual layer), the two layers are densely or fully connected.

To evaluate the accuracy of a neural net, a loss function (or cost function) is needed, which expresses the difference between the ground truth and the output calculated by the network regarding to a specific input. The loss function can be minimized by finding the optimum set of weights for the perceptrons in the ANN. As the ANN builds up from elementary activation functions (like the sigmoid or the tanh function), the contribution of each perceptron to the overall error on the output can be calculated by the backpropagation of the error calculated with the loss function. This process is called training in the machine learning terminology. Since calculating the gradient with respect to the loss is cheap, even traditional gradient seeking methods like gradient descent are suitable for training ANNs.

3.2. Convolutional neural network

Convolutional layers are able to capture the features of the data with less layers compared to MLPs if the relevant features are small compared to the dimensionality of the input which means the size of the input of a layer (or the network). The most obvious example is image recognition where the elementary building blocks like edges are small compared to the full image.

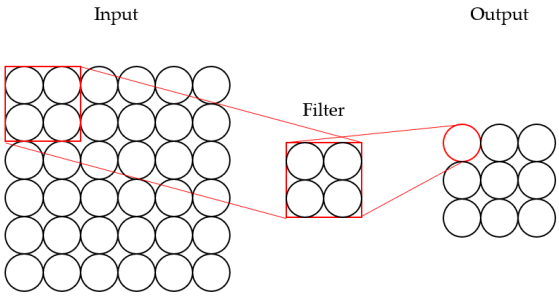


Figure 5. Schematic of the convolutional filter.

The convolutional layer applies a filter on the input where the filter size is smaller than the input size. The output of the layer is the convolution of the filter and the input or in other words it is the element-wise product of the filter and the appropriate parcel of the data where the filter is applied which leads to the reduction of the dimension as seen in Fig. 5.

This structure allows the filters to learn smaller features that are reoccurring in the input regardless on their exact location. This property makes them

ideal for image recognition tasks where topologies built on convolutional neural networks dominate the state-of-the-art in the recent years [12, 13, 14].

4. METHODOLOGY

4.1. Datasets

The dataset for training and evaluating the classifier is based on lattices generated by the Hoshen-Kopelman algorithm [15] . 5000 8-by-8 lattices were generated with different occupation probabilities (from 0 to 1 with stepsize 0.01) and they were labeled according whether they are percolating or not. The duplicates were filtered thereafter in order to avoid biased data that yielded 220996 unique lattices. The filtered dataset is referred to as the full dataset. The distribution of the percolating and the non-percolating lattices in the full dataset are seen in Fig. 6.

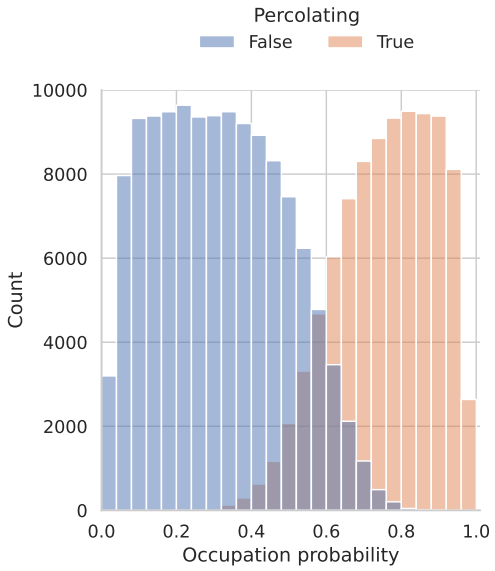


Figure 6. Histogram of the percolating and non-percolating lattices in the full dataset.

For 2D lattices the critical occupation probability is ≈ 0.593. Fig. 6 shows that non-percolating lattices are over-represented where occupation probability is subcritical and percolating lattices are over-represented for supercritical occupation probabilities (the drops for occupation probabilities near 0 and 1 are due to the much smaller number of lattices resulting from the removed duplicates).

The dataset was split to training, validation, and test sets in three different ways to examine the capabilities of the ANNs in case of different data distributions. These cases are referred to as *case A*, *case B*, and *case C*. As the aim of the study was to compare the multi-layer perceptron and the convolutional neural network in terms of prediction accuracy, the number of training samples was kept low compared to the test samples in each of the cases.

In *case A*, training data were selected as the portion of the full dataset with an uniform random distribution. The 1% of the full data were allocated for training and an additional 1% of the data were allocated for validation. The rest of the data were used purely for testing the accuracy of the trained models. *case A* models such cases where samples of all kinds (all occupation probability) are available for training; hence, it is considered as an easy case to learn.

In *case B*, lattices generated by occupation probabilities smaller than 10% and greater than 90% were allocated for training and validation while the rest of the data were allocated for testing. The training-to-validation ratio was 4 : 1. In this case, the model was trained on sparsely and densely occupied lattices only and was tested on almost every kind of lattices thereafter. Thus, it is considered a harder case to learn than *case A*.

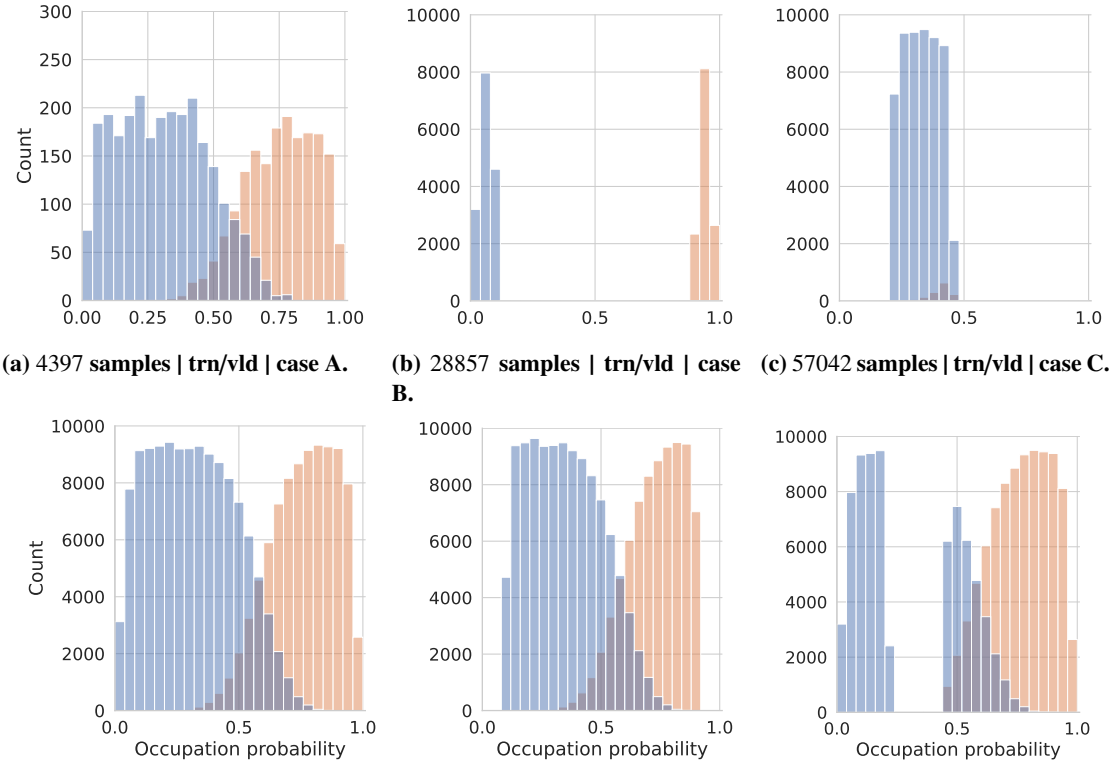
Lattices belonging to most of the occupation probabilities were excluded from the training data in *case B*. This difficulty was tried to increase in *case C* by including lattices with such occupation probabilities that yield to an imbalanced training dataset. Training data were selected according to occupation probability between 20% and 45% which yielded a skewed data distribution. The rest of the data were used for testing, the training-to-validation ratio was again 4 : 1.

Both in cases *B* and *C* the validation data were selected according to a uniform random distribution from the data allocated for training and validation. The data distributions are visualized in Fig. 7 with the number of samples indicated in the captions of the histograms.

4.2. Neural network architectures

There are many practical applications that could benefit from a classifier that can decide whether a 2D lattice percolates or not. However, there is no best practice on creating such classifiers; hence, two ANNs built from different kinds of layers were investigated in this study to find out whether CNNs are outperforming MLPs in this task as they do in computer vision problems. In order to do a fair comparison, the architectures were created in such a way that the number of the trainable parameters – that refers to the expression capacity of the ANN – were almost the same for the two networks. Moreover, the activation functions of the artificial neurons were rectified linear units (ReLU) in both networks. The lattices were fed to the classifiers as 2D arrays and the classifiers gave a probability $p \in [0, 1]$ on whether the lattice is percolating in return. The error of the prediction was quantified with the binary cross-entropy during the training and this error was backpropagated to the individual neurons.

The lattices were flattened in the first place in the MLP and the 64 input dimensions were fed to 2 hidden layers. Both layers utilized 4 neurons with ReLU activations and the prediction was given by a



(a) 4397 **samples | trn/vld | case A.** **(b)** 28857 **samples | trn/vld | case B.** **(c)** 57042 **samples | trn/vld | case C.**

(d) 216599 **samples | tst | case A.** **(e)** 192139 **samples | tst | case B.** **(f)** 163954 **samples | tst | case C.**

Figure 7. Histograms of the training/validation data (trn/vld) and the test data (tst) for the different cases.

single neuron activated by the softmax function.

The CNN was able to handle the lattices without transformation by nature; hence, the input was directly fed to the convolution layers. 3 convolution layers were utilized in sequence with 3 filters of size 2x2, and a convolution layer with only 1 filter of the same size as before. The output of the last convolution was flattened yielding a 16 dimensional latent representation of the lattice. This representation was fed to 2 dense layers with 8 and 4 neurons in sequence. The output of the last dense layer was fed to a classifying neuron utilizing the softmax function as its activation as in the case of the MLP.

The number of trainable parameters were 284 and 282 for the MLP and the CNN, respectively. These numbers are so small that no overfitting of the ANNs was suspected; hence, no attempt was taken in order to regularize the weights during the training. The weights were aligned according to the error calculated on the training data and the out-of-sample accuracy of the network was calculated on the validation data. The training was stopped after the validation accuracy did not rose for 30 consecutive epochs while the weights of the classifier were saved when the validation accuracy reached its maximum.

5. RESULTS

The MLP and the CNN were trained multiple times for each cases to examine the effect of random weight initializations. As the loss surface can have many local minima, different weight initializations can lead to different results even if the topology and the other hyperparameters are kept the same.

This disadvantageous behavior could be observed in *case B* and in *C*, but could not be observed in *A*. Hence, the trainings were repeated three times for *case B* and for *case C* for both topologies and the results were averaged over the repetitions.

The trainings were similar regarding the training and validation losses and the accuracies; hence, only a typical learning curve is depicted in Fig. 8. The closeness of the training and the validation curve indicates that omitting the regularization of the trainable parameters was a safe choice for this study.

In order to compare the accuracy of the different architectures in the different cases, the model weights were loaded back after each training and the model performance was evaluated on the test data. The results are visualized as confusion matrices in Fig. 9, while the corresponding data distributions are to be found in Fig. 7.

A good classification accuracy was achieved with both topologies in *case A*. This case was the only one wherein the training and the testing data shared the same distribution. Despite the size of the training data was greatly reduced compared to the other cases, it was not a difficulty for neither of the ANNs to achieve a recall over 90% for percolating and non-percolating lattices.

Case B presented a harder task and this fact is reflected in the recall metrics as well. Both topologies'

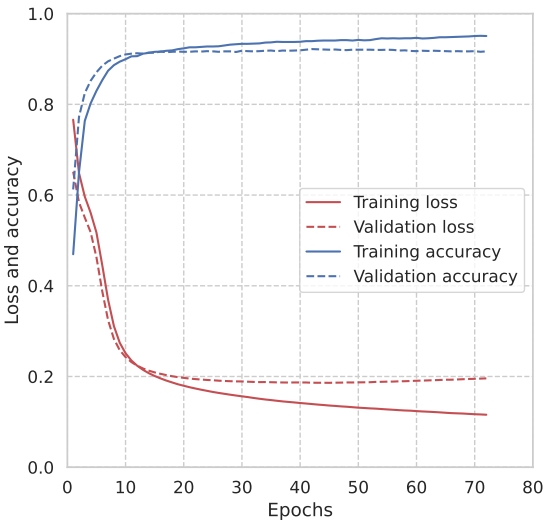


Figure 8. Learning curves of a typical training.

performance degraded in classifying non-percolating lattices but the MLP showed a worse performance than the CNN. However, the MLP performed better in classifying percolating lattices in contrast. Data distribution explains the bias of the recall towards the percolating lattices: these kind of lattices are a bit over-represented in the training and validation dataset compared to the non-percolating ones.

While the recalls provided by the MLP and the CNN were comparable in *B*, the CNN clearly outperforms the MLP in *case C*. The training and validation dataset contained almost twice as much lattices than in *case B* but the data distribution was strongly skewed as depicted in Fig. 7. While the MLP was able to achieve a recall of almost 80% for the underrepresented kind of data (percolating lattices), the CNN achieved a recall of 93.9% for the same lattices.

6. SUMMARY

The capabilities of ANNs were examined in this study on the classification of percolating lattices. The focus was on the comparison of the MLP and the CNN layer types as the latter represents the state-of-the-art for common image recognition tasks.

Two different classifiers were constructed with similar modeling capacity based on the two layer types and a comprehensive dataset was generated including more than 220000 unique 2D lattices with occupation probabilities spanning the range from 0 to 1. Three different test cases were set up to find out the behavior of the classifiers when their training data suffers from different difficulties.

Both the MLP and the CNN performed well when they were trained on a reduced dataset, achieving a recall above 90% for each kind of lattices. Although the CNN recognized non-percolating lattices a slightly better than the MLP, the difference was considered marginal.

The second test case contained a slightly imbalanced dataset but with training samples selected with extreme occupation probabilities that were close either to 0 either to 1. The CNN classifier gave slightly more accurate predictions for the underrepresented type of lattice than the MLP but both of the ANNs showed a degraded accuracy on recognizing that kind of lattice compared to the other.

The third case contained a larger but strongly imbalanced dataset than the second case. The CNN clearly outperformed the MLP in this case by giving a decent (above 90%) recall on both the under- and the overpresented kind of lattices.

The overall high accuracy of the CNN-based classifier and its superiority compared to the classifier based on MLP suggests that CNNs should be considered in the first place for the recognition of percolating lattices. This decision is also supported by the fact that real-world datasets are often imbalanced as is the case e.g. with the recognition of surface cracks in rocks [16]. When the imbalanced data cannot be cured with data augmentation, CNN-based classifiers can still deliver acceptable accuracies according to the findings of the present study.

REFERENCES

[1] Cybenko, G., 1989, “Approximation by superpositions of a sigmoidal function”, *Mathematics of Control, Signals, and Systems*, Vol. 2 (4), pp. 303–314.

[2] Zhang, W., Liu, J., and Wei, T.-C., 2019, “Machine learning of phase transitions in the percolation andXYmodels”, *Physical Review E*, Vol. 99 (3), p. 032142.

[3] Yu, W., and Lyu, P., 2020, “Unsupervised machine learning of phase transition in percolation”, *Physica A: Statistical Mechanics and its Applications*, Vol. 559, p. 125065.

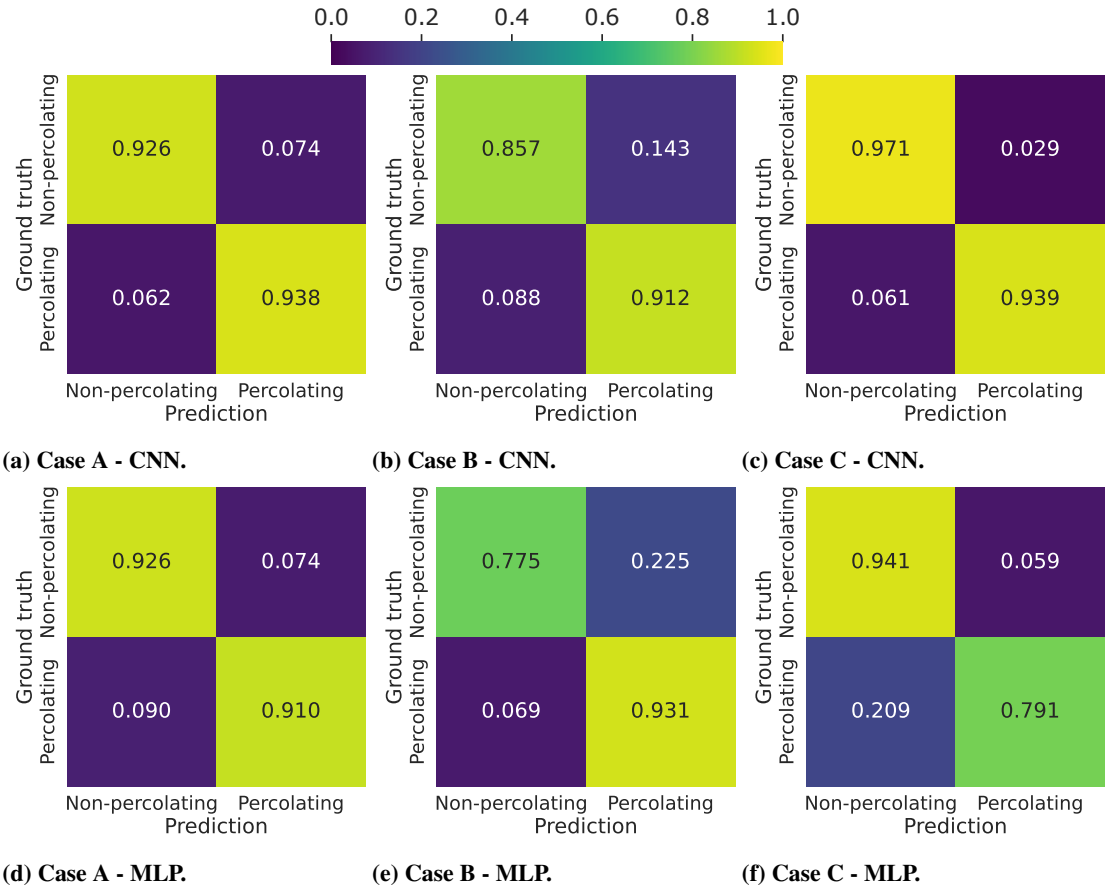
[4] Kamrava, S., Tahmasebi, P., Sahimi, M., and Arbabi, S., 2020, “Phase transitions, percolation, fracture of materials, and deep learning”, *Physical Review E*, Vol. 102 (1), p. 011001.

[5] Bayo, D., Honecker, A., and Römer, R. A., 2021, “Machine learning the 2D percolation model”, `2111.13116`.

[6] Bak, B. D., and Kalmár-Nagy, T., 2017, “Porcolation: An Invasion Percolation Model for Mercury Porosimetry”, *Fluctuation and Noise Letters*, Vol. 16 (01), p. 1750008.

[7] Vizi, M. B., Mizsák, P. Á., and Kalmár-Nagy, T., 2018, “Saturation in Regular, Exotic and Random Pore Networks”, *Fluctuation and Noise Letters*, Vol. 17 (03), p. 1850024.

[8] Bak, B. D., and Kalmár-Nagy, T., 2021, “A Size-Perimeter Discrete Growth Model for Percolation Clusters”, *Complexity*, Vol. 2021, pp. 1–16.



(a) Case A - CNN. **(b) Case B - CNN.** **(c) Case C - CNN.**

(d) Case A - MLP. **(e) Case B - MLP.** **(f) Case C - MLP.**

Figure 9. Accuracies reached on the test data sets. Numbers indicate the recall: the number of predictions related to the number of lattices in the corresponding ground truth category.

[9] Bottiglione, F., Carbone, G., and Mantriota, G., 2009, "Fluid leakage in seals: An approach based on percolation theory", *Tribology International*, Vol. 42 (5), pp. 731–737.

[10] Zhao, Y., Khandagale, P., and Majidi, C., 2021, "Modeling electromechanical coupling of liquid metal embedded elastomers while accounting stochasticity in 3D percolation", *Extreme Mechanics Letters*, Vol. 48, p. 101443.

[11] Zill, F., Lüdeling, C., Kolditz, O., and Nagel, T., 2021, "Hydro-mechanical continuum modelling of fluid percolation through rock salt", *International Journal of Rock Mechanics and Mining Sciences*, Vol. 147, p. 104879.

[12] Krizhevsky, A., Sutskever, I., and Hinton, G. E., 2012, "ImageNet Classification with Deep Convolutional Neural Networks", F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, Curran Associates, Inc., Vol. 25, pp. 1–9, URL `https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf`.

[13] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A., 2015, "Going deeper with convolutions", *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, pp. 1–9.

[14] He, K., Zhang, X., Ren, S., and Sun, J., 2016, "Deep Residual Learning for Image Recognition", *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, pp. 770–778.

[15] Hoshen, J., and Kopelman, R., 1976, "Percolation and cluster distribution. I. Cluster multiple labeling technique and critical concentration algorithm", *Physical Review B*, Vol. 14 (8), p. 3438.

[16] Alzubaidi, F., Mostaghimi, P., Si, G., Swietojanski, P., and Armstrong, R. T., 2022, "Automated Rock Quality Designation Using Convolutional Neural Networks", *Rock Mechanics and Rock Engineering*.