Neural network for indoor airflow prediction with CFD database

Q Zhou¹ and R Ooka²

¹ Department of Architecture, The University of Tokyo, Tokyo, Japan
² Institute of Industrial Science, The University of Tokyo, Tokyo, Japan

E-mail: qizhou@iis.u-tokyo.ac.jp

Abstract. Energy efficiency and indoor thermal comfort are both important in built environment, making it necessary to simultaneously take into consideration of the two aspects, building energy performance and indoor environmental quality, at the design stage. Coupled simulation between building energy simulation (BES) and computational fluid dynamics (CFD) enables providing each other complementary information with regard to building energy performance and detailed indoor environment conditions; however, the main drawback of CFD in computational cost limits its application. Neural networks (NNs) are considered as promising alternatives for CFD due to their advanced modelling abilities and high-speed computational powers. This research aims to confirm the feasibility of NN for indoor airflow prediction, which extends previous studies from two-dimensional to three-dimensional indoor space for more realistic conditions. The NN receives boundary conditions as input and outputs corresponding velocity and temperature distributions. Comparisons were made between NN predictions and CFD simulations regarding accuracy and time consumption on testing cases. The results show that the NN reproduces indoor airflow and thermal distributions with relative errors less than 12%. Time consumption for predicting the testing cases is reduced by 80% with the NN. The feasibility of NN for fast and accurate indoor airflow prediction is confirmed.

1. Introduction

Energy efficiency and indoor thermal comfort are both important in built environment, making it necessary to simultaneously take into consideration of the two aspects, building energy performance and indoor environmental quality, at the design stage. As building energy simulation (BES) and computational fluid dynamics (CFD) are powerful tools for energy consumption calculation and fluid heat and mass transfer computation, respectively, the integration of BES and CFD enables providing each other complementary information about building energy and environmental conditions, which is an accurate and effective tool to address the issue [1]. The coupled simulation of BES and CFD has successful applications in building design regarding envelope performance evaluation [2], air-conditioning system control [3], and local climate impact [4].

Owing to large differences in the time scale of heat transfer between solid and air, CFD generally requires much more time than BES in the coupled simulation. Moreover, to acquire informative and accurate CFD simulation results, fine discretization in space and time, with sophisticated turbulent models and high order schemes are preferred, leading to high computational cost and time consumption. In this regard, new approaches have been proposed to accelerate CFD calculation speed or to work as surrogates for CFD [5-8]. However, issues with simulation accuracy, calculation speed,
and implementation are still challenging. More research efforts are necessary to develop a fast and accurate indoor airflow prediction method.

In view of this consideration, neural network (NN) models which have advanced modeling abilities and high-speed computation capabilities are viewed as promising alternatives. In our previous studies [9, 10], NN models were implemented to reproduce two-dimensional indoor airflows which originally simulated by CFD. The flow and thermal characteristics were well represented via NN models with errors less than 10% for most cases. Meanwhile, the calculation time required for one case was significantly reduced by NN models. However, owing to the two-dimensional scenario, the complexity of airflow and thermal distributions are limited. In practical applications, flows in three-dimensional space are more common, thus the feasibility of NN for indoor airflow prediction in a three-dimensional room requires investigation. For this purpose, in the current study, we attempt to extend our previous studies from an ideal two-dimensional to a three-dimensional indoor space for more realistic conditions.

2. Methods

2.1. CFD database

Database is necessary to train and test the NN. In this study, the database consists of the velocity and temperature distributions in a three-dimensional room with specific boundary conditions, which are generated by CFD simulations. The target domain is three-dimensional with a size of 2.44 m in x, y and z directions, as shown in Figure 1(a). The airflow inlet with a size of 2.44 m (x) × 0.03 m (z) is located at the corner near the ceiling. The outlet has dimensions of 2.44 m (x) × 0.08 m (z), located at the diagonal position to the inlet, near the floor. In addition, there is a heated obstacle at the center of the room, with dimensions of 1.22 m (x) × 1.22 m (y) × 1.22 m (z). The configuration of the target domain is identical to that in reference [11], of which representative airflow characteristics such as jets, separations, and thermal plumes in an enclosed environment are included.

![Figure 1. Illustrations of (a) three-dimensional target domain; (b) partition of training data (red square) and testing data (solid blue circle).](image)

As for CFD simulations, the computational domain is discretized to hexahedral cells with non-uniform size (Figure 2(a)). Most boundary conditions are set the same as those in experiments [11] and kept constant in different cases, as listed in Table 1. However, supply air temperature and wall surface temperature vary with case in the range [289, 299] (K) and [273, 308] (K), respectively, to account for various airflow and thermal distributions. Considering combinations of the supply temperature and wall temperature, there are 48 cases were generated as testing data, as demonstrated in Figure 1(b). For training data, cases with maximum, median and minimum supply temperatures and wall temperatures were selected, resulting in 9 training data generated. The steady-state Reynolds-averaged Navier–Stokes (RANS) simulation with the RNG k–ε turbulence model was applied in CFD
for generating training and testing data. Buoyancy effect due to temperature difference was treated by the Boussinesq approximation. The simulations were run on a Linux platform using open-source CFD software OpenFOAM 6.

CFD simulation results of velocity and temperature distributions at grid resolution (raw data, Figure 2(a)) were converted to block resolution (8 (x) × 8 (y) × 8 (z)) through averaging the grid values within each block (Figure 2(b)). The data manipulation removes redundant flow information but preserves the main features of the flow field. For each case, boundary conditions and the corresponding velocity (vector components in the x, y and z directions) and temperature distributions at the block resolution constitute one labeled input-output pair in the database.

![Figure 2. Demonstrations of (a) discretization of the computational domain in CFD simulations; (b) data manipulation for the database.](image)

| Table 1. CFD simulation conditions. |
|------------------------------------|
| **Mesh**                           | Hexahedral          |
| **Count**                          | 45,449              |
| **Inlet**                          | U=1.366 (m/s), T ∈ [289, 299] (K) |
| **Ceiling**                        | T=298.8 (K)         |
| **Floor**                          | T=299.9 (K)         |
| **Walls**                          | T ∈ [273, 308] (K)  |
| **Obstacle**                       | T=309.7 (K)         |
| **Turbulence model**               | RNG k-ε model       |
| **Buoyancy effect**                | Boussinesq approximation |

2.2. **NN model**
In this study, NN model with multiple hidden layers in feedforward structure is implemented, as illustrated in Figure 3. The input layer receives the input vector, i.e., boundary conditions, while at the output layer the temperature and velocity values of the 512 blocks are outputted. In each hidden layer, neurons convert signals from a previous layer and transfer them to the next layer. The output vector is converted to temperature and velocity tensors to represent three-dimensional distributions of temperature and velocity in the target room. During the training process, a gradient descent-based algorithm is applied to update NN parameters to minimize training error. As for hyperparameter settings, it is difficult to determine an optimal setting from scratch. Here, different hidden layer numbers and neuron numbers are taken into account for hyperparameter tuning with grid search, leading to 15 scenarios for comparison, which will be demonstrated in Section 3. The detailed hyperparameter settings are listed in Table 2.

Since the training process is nonconvex and iterative, it is critical to determine the merit for terminating the training. In the current study, an early-stop technique is applied, of which the
mechanism is illustrated in Figure 4. Here, training is terminated as the monitored metric, validation error, has stopped reducing in consecutive 5000 epochs. The training at this stage is considered to reach convergence. After the training is terminated, parameters at the early-stopping point are retrieved. Then, the NN model is tested by testing data for performance evaluation.

![Figure 3. Outline of NN architecture and structures of input and output data.](image)

| Table 2. NN Hyperparameter settings. |
|--------------------------------------|
| Number of hidden layers              | 1 / 3 / 5                          |
| Number of neurons per hidden layer    | 20 / 40 / 80 / 120 / 200            |
| Activation function                  | Tanh                               |
| Gradient descent algorithm           | Adam                               |
| Learning rate                        | 0.001                              |
| Batch size                           | 1                                  |

![Figure 4. Illustration of training and validation error variations with epochs and determination of early-stopping point.](image)

2.3. Evaluation criterion
As for hyperparameter tuning, mean Euclidean distance between NN predictions and CFD simulation results of temperature or velocities is used as criterion for comparison, as defined in Equation (1):
Ed(\(NN, CFD\)) = \sqrt{\frac{\sum_{i=1}^{n} (NN_i - CFD_i)^2}{n}}

where, \(NN_i\) is NN prediction of temperature or velocity of the \(i\) th block; \(CFD_i\) is the true value of temperature or velocity of the \(i\) th block generated by CFD. \(n\) is total block number, i.e., 512 for each case.

To quantitatively evaluate NN performance on the 48 testing cases, relative error is used to calculate the discrepancy between NN predictions and CFD simulation results of velocity and temperature, as defined in Equation (2):

\[
Er(\(NN, CFD\)) = \sqrt{\frac{\sum_{i=1}^{n} (NN_i - CFD_i)^2}{\sum_{i=1}^{n} (CFD_{max} - CFD_{min})^2}}
\]

where, \(NN_i\) and \(CFD_i\) are NN predictions and CFD simulations of temperature or velocity of the \(i\) th block, respectively; \(CFD_{max}\) and \(CFD_{min}\) are maximum and minimum temperature or velocity values from CFD simulations for each case. \(n\) is total block number.

### 3. Results and discussion

To confirm the feasibility of NN for indoor airflow prediction for more realistic conditions, comparisons were made between NN predictions and CFD simulations regarding accuracy and time consumption on 48 testing cases, which will be demonstrated in subsections 3.1 and 3.2, respectively.

#### 3.1. NN generalization capability

NN prediction accuracy on unknown data essentially reflects its generalization capability, the crucial performance index for the NN model. As 15 scenarios with different hidden layer numbers and neuron numbers are considered for hyperparameter tuning, we implemented 15 NN models and their generalization capabilities are compared.

Figure 5 shows Euclidean distances between NN predictions and CFD simulations for scenarios with different hidden layer and neuron numbers. Each box represents the statistics of the Euclidean distances on the 48 testing cases. Notwithstanding different hidden layer and neuron numbers, the Euclidean distance distributions are similar in all scenarios regarding temperature and velocities in \(x\), \(y\) and \(z\) directions. Given the 15 combinations, the variation in hidden layer and neuron numbers has limited impact on NN performance. This is probably due to over-parameterization of the NN model, of which parameters are significantly more than training data. The redundant parameters indicate the capacity of the NN model is large enough for containing all the patterns in the training data and thus some increase or decrease in hidden layer and neuron numbers may have little influence [12].

In this regard, one of the 15 NN models, e.g., NN with 3 hidden layers and 40 neurons at each hidden layer, is selected as a representative for detailed evaluation of generalization capability. Figure 6 shows distributions of temperatures and velocity vectors of two testing cases. For the simplification of data visualization, two planes at vertical (\(X = 1.22\) m) and horizontal (\(Z = 0.61\) m) directions are clipped, and temperature (in contours) and velocity (in streamlines) distributions are illustrated on the two planes. In general, distributions of temperatures and velocity vectors predicted by the NN model are consistent with those simulated by CFD. The temperature stratification due to buoyance effect in the \(z\) direction is well reproduced. Meanwhile, vortexes near the corner and flow separation due to blockage are appropriately represented as well.
Figure 5. Comparisons of Euclidean distances between NN predictions and CFD simulations regarding scenarios with different hidden layer and neuron numbers.

Figure 6. Distributions of temperatures and velocity vectors of two testing cases in (a) vertical plane; (b) horizontal plane.
The calculated relative errors of NN predictions of temperature and velocity of the 48 testing cases are shown in Figure 7. Relative errors of each case show some differences; however, there is no significantly high value. The maximum relative errors are 10.5%, 12%, 6%, and 10% for NN predictions of temperature and velocities in x, y and z directions, respectively. All the flow and thermal patterns in the testing cases are well reproduced. Compared to the results in two-dimensional space with relative errors less than 10% for velocity prediction and less than 6% for temperature prediction [10], the NN demonstrates comparable performance in prediction of three-dimensional airflow. In fact, there is still room for improvement in NN performance since the training data and pre-defined space for hyperparameter tuning are both limited in this study. On the one hand, it is estimated that with more trials in different hyperparameters including learning rate, batch-size, etc, the generalization capability of the NN could be further improved. On the other hand, given more training data, the prediction accuracy of the NN could be improved since more flow pattern is learned. However, there should be a limit in generalization capability improvement as the training dataset volume increases and the time required for generating more training data should be considered as well.

![Figure 7. Relative errors of NN predictions of temperature and velocity of the 48 testing cases.](image)

### 3.2. Calculation time

In addition to prediction accuracy, calculation time is another critical point to concern. In this study, the calculation time refers to the total time consumed for predicting temperature and velocity distributions of the 48 testing cases. As for CFD, the calculation time is straightforward, which equals to the time for executing CFD simulations. However, for NN model, owing to the necessity of training data which is generated by CFD, and the training process, the calculation time is the sum of time consumed for generating training data and that for NN training. The time required by the NN model to output predicted values are less than 1 s, thus it is ignored in the calculation time. CFD simulations are carried out on a Linux server with Intel Xeon E5-2667 v4 3.20GHz CPU. Considering parallel computation with multiple cores and processors, it requires about 10 min per case, resulting in 9 times
faster than single processor. On the other hand, NN implementation and training are performed on a Windows desktop with Intel Core i7-6700 3.40GHz CPU, which requires about 12 min to train the NN model.

Comparison of calculation time for predicting the 48 testing cases between the NN model and CFD is demonstrated in Figure 8. CFD requires about 8 h for the 48 testing cases, while for NN prediction, 1.5 h are consumed for generating 9 training cases and 0.2 h for training, resulting in about 80% reduction in total time consumption as compared to CFD simulations. The term ‘fast’ is reflected in two aspects. On the one hand, time consumption is significantly reduced for CFD simulations since only a few cases are calculated for generating training data; on the other hand, predictions of unknown cases are accomplished instantly by the well trained NN.

![Comparison of time consumption for predicting the 48 testing cases between NN model and CFD using parallel computation with 8 cores 16 processors.](image)

**Figure 8.** Comparison of time consumption for predicting the 48 testing cases between NN model and CFD using parallel computation with 8 cores 16 processors.

### 4. Conclusions

This investigation aims to confirm the feasibility of NN for fast and accurate indoor airflow prediction, which extends previous studies from two-dimensional to three-dimensional indoor space for more realistic conditions. For this purpose, a NN model is implemented, which is utilized for predicting flow and thermal fields in a three-dimensional space. Comparisons are made between NN predictions and CFD simulations regarding accuracy and time consumption on 48 testing cases. With the relative errors less than 12% for temperature and velocity predictions, and reduction of the calculation time by about 80% as compared to CFD simulations, the feasibility of NN for fast and accurate prediction is confirmed.

### References

1. Tian W, Han X, Zuo W and Sohn M 2018 *Energy Build.* 165 184–99
2. Gijón M, Xamán J, Álvarez G and Serrano J 2013 *Build Environ* 68 22–34
3. Tian W, Sevilla T, Zuo W and Sohn M 2017 *Build Environ* 122 269–86
4. Allegrini J and Carmeliet J 2018 *Urban Clim.* 24 340–59
5. Zuo W and Chen Q 2009 *Indoor Air* 19 33–44
6. Khan M, Delbosc N, Noakes C and Summers J 2015 *Build Simul.* 8 405–14
7. Li K, Su H, Chu J and Xu C 2013 *Build Environ* 60 150–7
8. Zhang W, Hiyama K, Kato S and Ishida Y 2013 *Build Environ* 63 89–96
9. Zhou Q and Ooka R 2020 *Build Simul.* 13 1409-1423
10. Zhou Q and Ooka R 2020 *Energy Build.* 230 110525
11. Wang M and Chen Q 2009 *HVAC&R Res.* 15 1099–119
12. Frankle J and Carbin M 2019 arXiv: 180303635v5. 1–42.