ARINET: USING 3D CONVOLUTIONAL NEURAL NETWORKS TO ESTIMATE ANNUAL RADIATION INTENSITIES ON BUILDING FACADES

Jung Min Han[1,2], Chih-Kang Chang[3], and Ali Malkawi[1,2]
[1]Harvard Graduate School of Design, Cambridge, MA
[2]Harvard Center for Green Buildings and Cities, Cambridge, MA
[3]Harvard John A. Paulson School of Engineering and Applied Sciences, Cambridge, MA

ABSTRACT

Artificial intelligence and data-driven modeling are becoming more prominent in the building, and construction sectors. Physics-based models usually require significant computational power and a considerable amount of time to simulate output. Therefore, data-driven models for predicting the physical properties of buildings are becoming increasingly popular. The objective of this research is to introduce Artificial Neural Networks (ANNs) methods as a means of representing the physical properties of buildings. Achieving this goal will illustrate the future capacity of integrated neural networks in building performance simulations. The Annual Radiation Intensity Neural Network (ARINet) demonstrates the feasibility of using a 3D convolutional neural network to predict the surface radiation received by building façades. The structure of ARINet is composed of 3D convolution, fully connected, and 3D deconvolution layers. In this research, it was trained on 1,692 datasets and validated by 424 datasets generated by a physical simulator. ARINet showed errors in 0.2% of the validation sets.

INTRODUCTION

In the present of the Big Data era, it is becoming more and more common to employ data-driven models, especially when physical models may not fully explain the operational environment (Simon, 2019). In building physics, models are useful for clarifying a building's physical properties and when making inferences about the future, as well as for providing feedback on design changes and facilitating optimization. With recent increases in computing power and the substantial availability of data sources, the combined use of both modeling techniques is likely to be essential to the future of building performance simulation (BPS). Physics-based models designed to examine the surface of the earth with conservation laws. Unlike conservation laws, models used empirical methods are mostly inductive and based on observable phenomena (Goldstein & Coco, 2015). For example, building a physical sky model requires a certain empirical model to calculate the local impact of diffuse solar radiation on a horizontal surface. However, it may also require a number of assumptions to predict the surrounding natural phenomena (Han, Malkawi, & Gajos, 2019).

As more data are made available, it is becoming increasingly difficult to incorporate all available sources and fewer assumptions into a single predictor. It can be argued that the empirical parameterization of numerical models should be conducted using ANNs methods because this type of tool is designed to operate on large, multi-dimensional datasets (Goldstein & Coco, 2015).

Machine learning has attracted attention in predicting surface solar radiation (Mohandes et al., 1998; Yadav & Chandel, 2014; Voyant et al., 2017) and building energy (Goldstein & Coco, 2015; Amasyali & El-Gohary, 2018). Artificial neural networks (ANNs) can provide innovative ways of solving design problems, allowing designers to receive instantaneous feedback on the effects of proposed changes to a building's design.

Unlike computational fluid dynamics, solar radiation analysis is scalable but still requires computational power to simulate the cumulative radiation values received on a building's surface throughout the year. Therefore, an ANN model embedded in the design process has the potential to encourage performance optimization and design exploration by reducing performance simulation time.

In both practice and academia, physical sky models (Bird, 1981; Perez, 1993) have widely been used to evaluate daylighting performance and energy consumption. However, using ANNs to create virtual environments is new to both. The benefits include reproducibility and scalability; the former means a reduction in the time complexity related to calculating optimized design solutions during iterative simulation processes, and the latter facilitates the free exploitation of local information obtained from sensors to represent local conditions in global systems and models.

LITERATURE REVIEW

Surrogate models have been used for decades as high-performing function approximators. Examples of commonly used surrogate modeling techniques in BPS include linear regression (Gratia, 2002; Jaffal, Inard, & Ghiaus, 2009; Hygh, Decarolis, Hill, & Ranjithan, 2012; Catalina, Iordache, & Caracaleanu, 2013; Geyer & Schlüter, 2014; Yi, Ritter, 2015), Bayesian networks (Heo, Choudhary, & Augenbroe, 2012; Chong & Menberg, 2018), evolutionary algorithms (Machairas, Tsangrassoulis, & Axarli, 2014), and ANNs (Kalogirou, 2000; New, Ridge, & Parker, 2017; Ascione, Bianco, Stasio, Maria, & Peter, 2017; Singaravel, Suykens, & Geyer, 2018). Since currently available research on and tools for ANNs are not specifically designed to represent architectural geometry and its geometrical relationships, a review of different neural network architectures for representing building geometry in ANN modeling is offered below.

Recurrent neural networks (RNNs) and convolutional neural networks (CNNs) are prevalent. DNNs deal with vast datasets for real-world problems. RNNs have been shown to excel at modelling sequential data if they have access to the previous context and any time-dependencies. However, one current limitation of the RNN structure is that input is explicitly required to be single-dimensional, meaning any multi-dimensional data must be preprocessed and flattened before being fed into the RNN model's architecture (Connor & Atlas, 2002). A CNN is an example of a DNN capable of using multidimensional data (i.e., images). However, CNNs lose the ability to learn from long-term memory and involve significant increases in computational cost as the input data increase, due to their multi-dimensionality. Increasing the dimensionality of the data and network architecture of an ANN causes the computational time to increase greatly, due to the multitude of layers and parameters involved in tuning the proposed architecture. One method proposed in the literature is to extend the functionality of RNNs to multi-dimensional data (Graves et al., 2006). The proposed method extends the dimensionality of the data input into the RNN architecture, while avoiding the extreme scaling issues experienced by CNNs. Multi-dimensional recurrent neural networks alter the architecture of conventional RNNs to expand the number of recurrent connections and forget gates, such that there is only one for each dimension. This could be an interesting architecture for use in investigating 3D physics-based simulations for specific multidimensional spatial and temporal problems (e.g., 3D buildings with values that change over time).

Robotics researchers have also analyzed point cloud data, such as in "VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition" (Maturana & Scherer, 2015). Point cloud data are sets of points with coordinates within a 3D space measured by LiDAR or an RGBD camera. Scherer (2015) proposed that VoxNet, a basic 3D convolutional neural network (3DCNN) architecture, could be applied to create fast and accurate object class detectors for 3D point cloud data. VoxNet is composed of an input layer, two 3D convolutional layers, a maxpool layer, a fully connected (FC) network, and an output layer. In order to cover objects of different scales, (e.g., a truck or traffic sign) multiresolution VoxNet can be achieved by combining two networks with identical VoxNet architectures, each receiving occupancy grids at different resolutions. The information from both networks can then be fused by concatenating the outputs of their respective FC (128) layers. Along with other pre-processing and training techniques, VoxNet is able to achieve surprisingly good results from such a simple structure.

DATA PROCESSING

The generic workflow in the present research consisted of three parts: data generation, preprocessing, and modeling and validation. Conventional modeling and simulation software were used in all three steps. To make the process discrete, Python3 and MATLAB were employed at the same time.

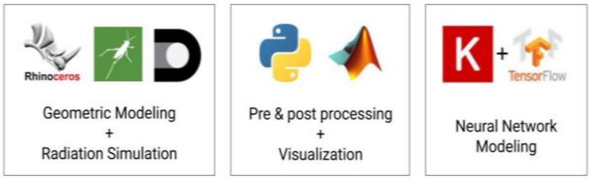
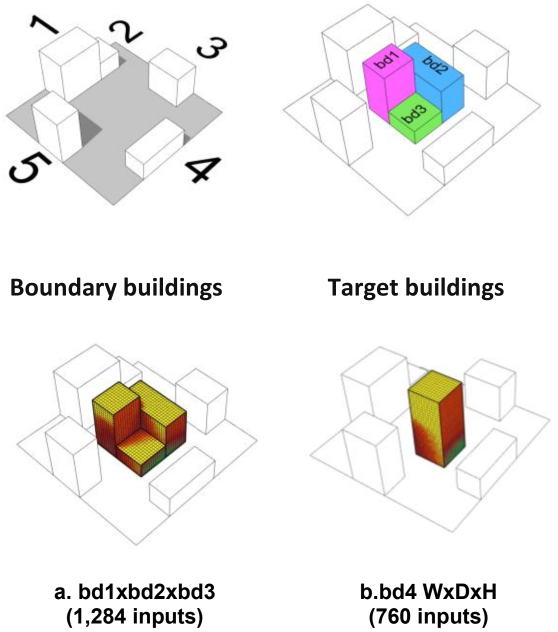


Figure 1 Generic workflow and related software

Data generation

For the initial attempts at solving problems with the given datasets, sequential information was excluded. Thus, RNNs were no longer beneficial. The 3DCNN was then chosen to serve as the baseline architecture design, in order to predict the annual radiation exposure received by a building façade. With regards to data generation, we utilized both modeling and simulation tools to collect surface radiation values throughout the year. Specifically, the parametric modeling tools Grasshopper and Rhinoceros were employed. For the physical radiation simulation, a plug-in for Rhino called DIVA (Jakubiec & Reinhart, 2011) was utilized. In total, 2,044 datapoints were collected, including height variations for multiple target buildings and volume variations for a single building. Figure 2 shows the two main types of variation in the building geometry and the datasets exploited for the modeling and simulation process.



Boundary buildings

Target buildings

a. bd1xbd2xbd3 (1,284 inputs)

b.bd4 WxDxH (760 inputs)

Figure 2 Datasets obtained from physical simulations (DIVA)

The data, including the annual average exposure to surface radiation of the building, were initially collected on a monthly basis; however, for simplicity, only annual values were utilized to model the 3DCNN. The different locations of the five boundary buildings were fixed in order to simulate a target building with surrounding conditions, and the various options for the target buildings were evaluated via a radiation simulation for the building façades. We split the 2,044 datapoints into 1,635 training and 408 validation sets to evaluate the performance of the proposed 3DCNN.

Data processing

The initial problem with converting the extracted data to a 3D voxel representation was matching the different coordinates to their boundary conditions. Because the output of DIVA for Rhino's grid system cannot evenly distribute the local coordinates of the building façades, pre-processing was necessary to control the points and values representing all radiation values equivalent to each sub-cube. Figure 3 shows the uneven distances between each pair of datapoints when facing the boundary conditions. Therefore, the edge values and voxel map created to represent the 3D information for the radiation received were ignored.

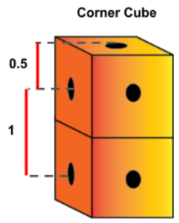


Figure 3 Basic unit for preprocessing of the target building's dataset.

After processing the edge data, the model input, denoted with an X, was padded with binary information (i.e., 1s and 0s). In our voxel representation of the 3D space, 1 represented a building and 0 indicated air. The predicted values that comprised the model output, denoted with a Y, could then be the surface radiation values for every coordinate. The structures of X and Y were as follows:

•X.shape = (number of example, xdim, ydim, zdim)
•Y.shape = (number of example, xdim, ydim, zdim)
(MINMAX normalized)

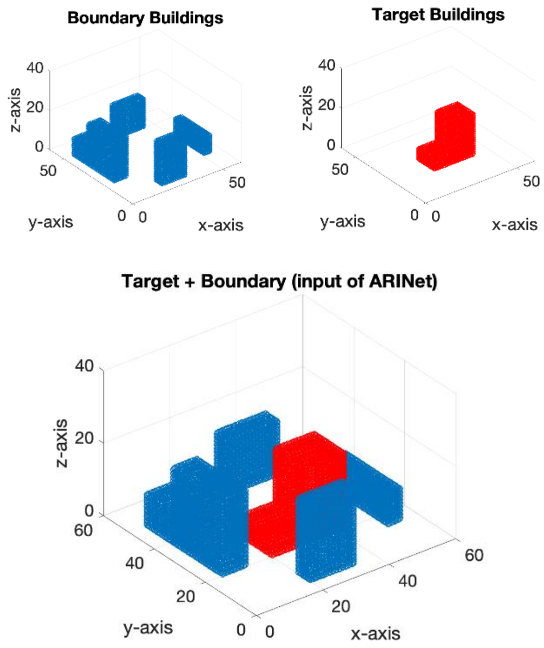


Figure 4 Preprocessing of the combined boundary and target buildings

After processing each voxel shape, the boundary and target buildings were combined to serve as the input for ARINet. MATLAB was used for this process, due to its computational efficiency and the instant visual feedback it provides. The MATLAB code output a numpy array, allowing Python to normalize the data for the model.

MODEL ARCHITECTURE

This section describes the ARINet structure used to predict radiation intensity on a building's façade. The output of the network was the radiation intensity (i.e., a numerical value); thus, mean squared error (MSE) was preferred as the loss function. However, the simulation to generate the dataset could only determine the radiation intensity received by the building's surface. The loss function was then modified to account only for the MSE on the building's surface.

When building ARINet, VoxNet, a 3DCNN for real-time object recognition (Maturana & Scherer, 2015), was referenced for the baseline architecture. Voxnet is a 3DCNN that can be applied to create fast and accurate object class detectors for 3D point cloud data. Due to the simplicity of Voxnet's architecture, a relevant model was obtained from the literature and the dataset was generated as voxel points (0s and 1s in space).

Using ARINet, a latent variable containing the hidden information from the input was obtained. By having latent variables as a part of the training models, we were able to use the autoencoder architecture to map the latent variables back to the 3D space in which the results of the radiation resided. In order to increase the range of captured information (i.e., handle the shadow issue), we used more layers in ARINet than in VoxNet.

ARINet assumed the world to be a 51 x 51 x 51 grid in which both the target and boundary buildings existed. Based on this assumption, superimposed binary output matrices for all of the buildings in the world were produced as part of the data processing stage. Figure 5 illustrates the model architecture of the 3DCNN, beginning with the 51 x 51 x 51 voxel grid. The proposed ARINet consisted of two 3D convolutional layers before the max_pooling layer. The architecture retrieved the network by passing two additional 3D deconvolutional layers. Basically, 3D image models were mapped onto latent spaces and later reshaped by calculating the difference values for radiation after passing into loss functions in the proposed 3DCNN architecture.

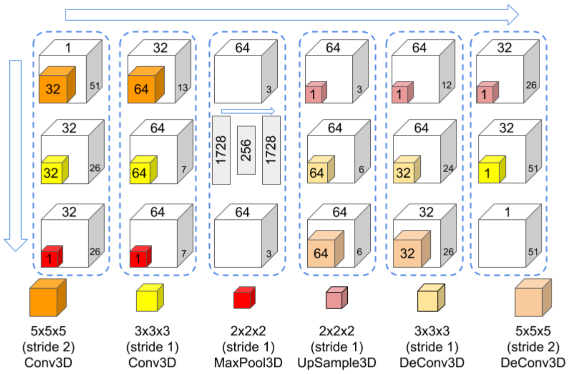


Figure 5 3DCNN architecture of the project

RESULTS AND DISCUSSION

ARINet was trained with a batch_size of 32 and adam optimizer; 10 epochs were used to minimize losses. Figure 6 shows both the training and validation losses after 10 epochs, which reached a 0.002 error on average.

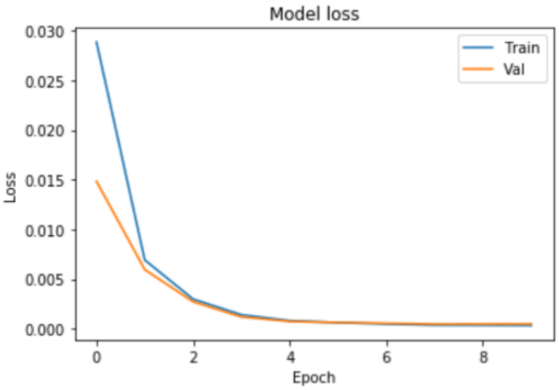


Figure 6 Loss graph per epoch

Predictions for validation sets

To avoid overfitting and estimating the future feasibility of the trained model, completely new datasets were used for validation and testing. For the validation, the same type of building geometry was used. Figure 7 illustrates two visualizations of the validation sets: three buildings and a single building; these include the original values of the radiation (top), predicted radiation values (middle), and errors for the given datapoints (bottom) for the validation sets, which were never used for training. The results were visualized to offer a closer look at the predictions for each building. It was determined that overall they were quite accurate, but there were significant errors in some of the data at the boundaries of the three buildings (e.g., around `y = 32.5` and `x = 22.5`), even for the less complicated building shapes where the adjacent buildings were of the same height. It can be assumed that ARINet learned some specific rules about the boundaries of the buildings, as this type of building comprised the majority of the training data.

However, there is still the potential to train neural network models for physics-based phenomena in the real world. If the problem with boundary surrounding conditions could be fixed, the model would predict the radiation received by a building much better than what was possible in the current research. Furthermore, more options regarding the increments of the training datasets would provide more intuitive results and better predictions. This is important if the approximation of radiation received by a building is to be accurate enough to estimate the internal heat gain through the façade.

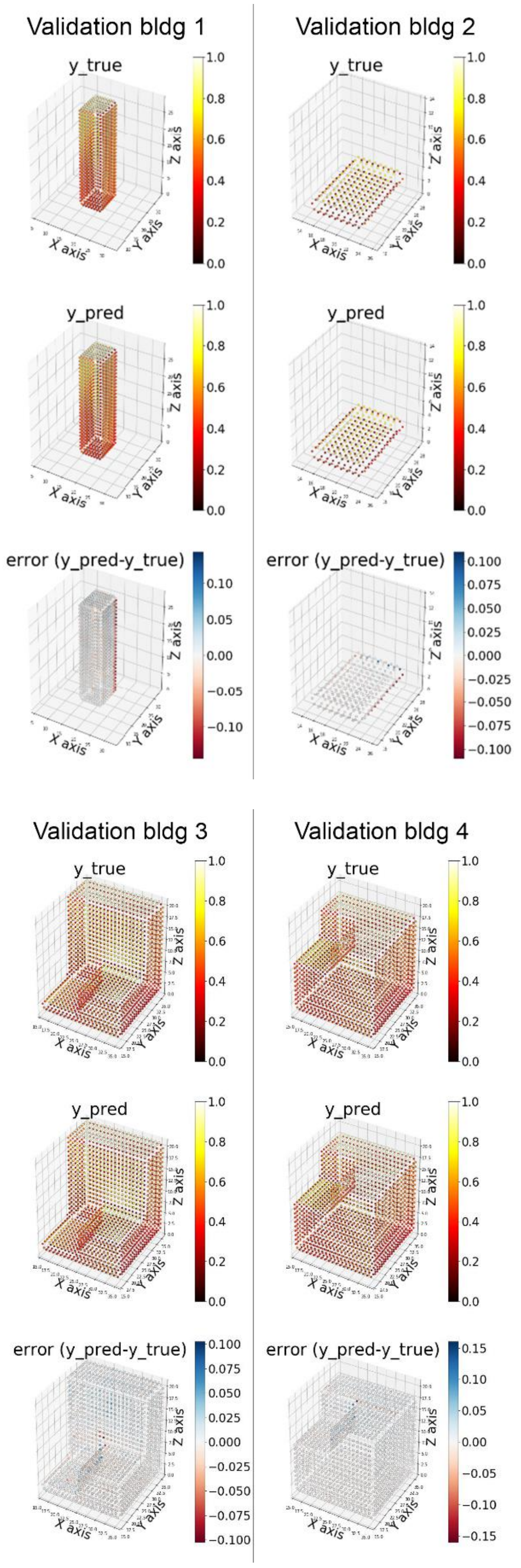


Figure 7 Results plot for four validation options

Predictions from the test sets

This section describes the results of the radiation received by the building façades in the completely new test sets. Three alternative buildings were selected to demonstrate the results and serve as the subject of a detailed analysis. Figure 8 illustrates three types of building geometry: rectangular with a horizontal overhang, round, and cube-shaped with an internal empty core.

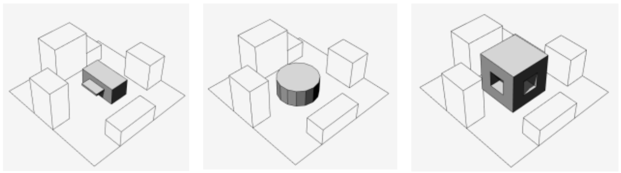


Figure 8 Reference geometries for the completely new buildings with boundaries

After providing these three options for the test sets, we also modified additional options with no boundary buildings. Six options were tested for predicting the radiation received. As ARINet was fully trained using building geometries of a specific type (i.e., box-shaped) with boundary buildings, the test sets were not anticipated to be perfectly predicted. Figure 9 shows that the results for both the shading device and building offered relevant visualization output, leading to a low error rate (i.e., 0.0309) for the boundary buildings.

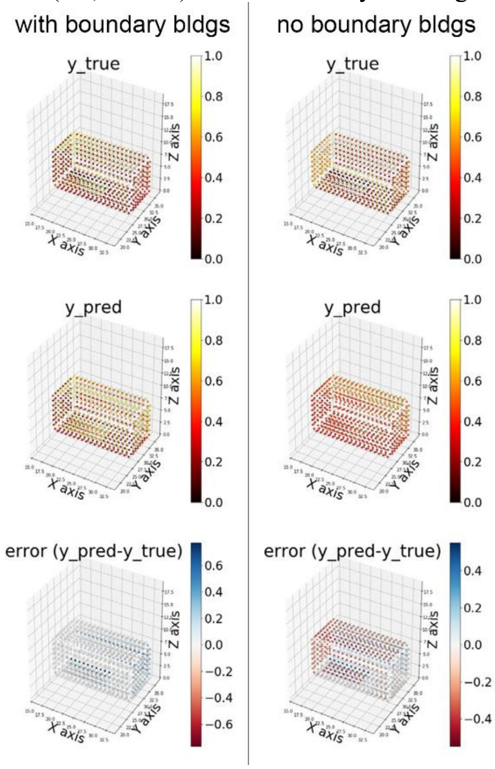


Figure 9 Analysis and comparison results (shading)

However, the absence of the boundary buildings resulted in a higher error rate (i.e., 0.061) and invalid estimation of the radiation intensities received by the buildings.

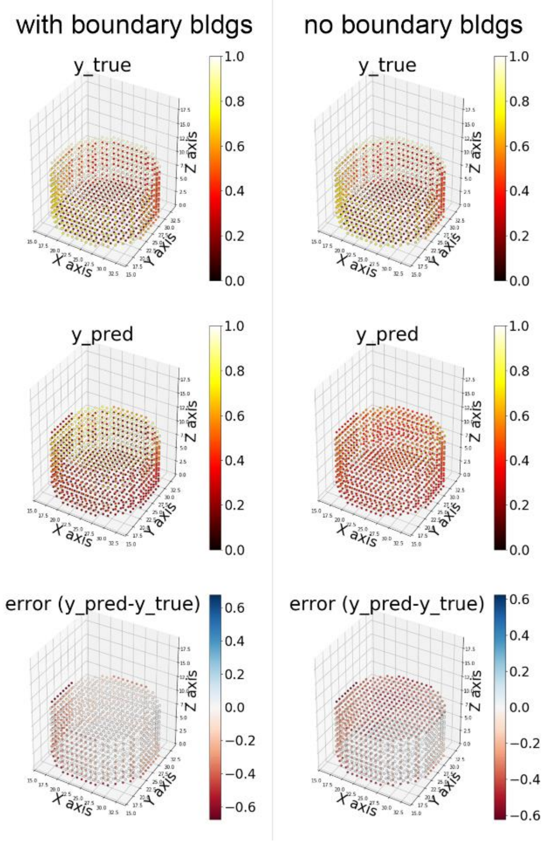


Figure 10 Analysis and comparison results (round)

The model was able to predict the values for the round building (see Figure 10). However, a higher error rate was observed for the vertical façades on the round shapes. This is because there were no options for rotated façades in the training sets. Also, given that the model was initially trained with boundary buildings, the result with no boundary also produced a greater error (i.e., 0.0702) than did the other option (i.e., 0.0338). Therefore, the results did not realistically represent the radiation received by the building façades, due to the information missing for the hidden properties of the surrounding buildings. However, this could be enhanced by training with different boundary buildings, which is a fixed property in the current training dataset. Increasing the rotation options for façades in the training dataset may improve the accuracy of predictions regarding vertical façades.

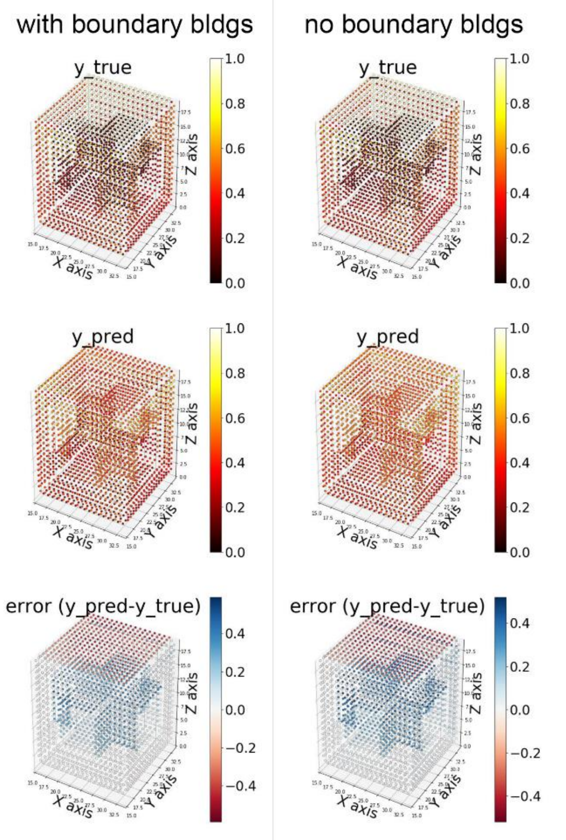


Figure 11 Analysis and comparison results (inner core)

The last test option was cube-shaped with an internal empty core (see Figure 11). This option offered the lowest prediction accuracy, even with boundary buildings (i.e., 0.0538). In this case, the predictions for the horizontal rooftop and internal core surfaces were quite different from the expected results. This was because our model did not count the radiation bouncing off the opposite surface; thus, the model miscalculated the rays bouncing from the sun. It can easily be seen that the function of the ambient bounce in DIVA was very low, Level 1 or 2 for our model, due to the traits of the training sets. To fix this problem, radiation prediction for internal spaces such as floors with large windows should be used in training. In addition, the accuracy for this building without boundary buildings was better than for the round building (i.e., 0.0637). From the visualization, we could see that the vertical facade actually had a high level of accuracy, since the buildings for the training set included these properties.

CONCLUSION

This research proved the feasibility of converting a physics-based model into a data-driven model with some limitations. The total training time of 10 epochs took only 17 minutes and 30 seconds, which was yielded reasonable results for the test sets. Once ARINet was trained, the model could easily be utilized to give instant solar feedback for building façades. It is more feasible to apply instant feedback during the early design decision-making process, since this requires relatively low accuracy but high efficiency. Furthermore, by using this model, designers and consultants can practically optimize building geometry based on local solar information and contextual data such as boundary buildings. Early design decision support requires manifold design options with relevant performance feedback. In such cases, ARINet can be exploited widely during the design process.

Greater data generation and more input that accurately represents physical phenomena in real situations are recommended for increasing the accuracy of ARINet and its usability. Furthermore, the hyperparameters for ARINet, such as its optimizer, loss function, etc., should be explored. Additionally, the integration of ARINet into RNNs for time series-based predictions of radiation should be investigated in terms of monthly and daily resolutions. It is highly likely that this new type of approach to radiation modeling in buildings and architectural modeling environments that utilize ARINet could be a unique and prominent research opportunity for applied artificial intelligence in architecture.

ACKNOWLEDGMENTS

The authors would like to thank Dabin Choi and Timothy Lee for their invaluable collaboration in data processing. It is always a pleasure to work with such engaging people. This project would not have been possible without the support of Pavlos Protopapas and the Harvard Data Science 109/209B class.

REFERENCES

Amasyali, K., & El-Gohary, N. M. (2018). A review of data-driven building energy consumption prediction studies. Renewable and Sustainable Energy Reviews, 81(2016), 1192–1205.

Ascione, F., Bianco, N., Stasio, C. De, Maria, G., & Peter, G. (2017). Artificial neural networks to predict energy performance and retrofit scenarios for any member of a building category : A novel approach. Energy, 118, 999–1017.

Arsenio, E. (2012). Book review. Journal of Transport Geography, 25, 163–164.

Bird, R. E., & Hulstrom, R. L. (1981). Simplified clear sky model for direct and diffuse insolation on horizontal surfaces.

Catalina, T., Iordache, V., & Caracaleanu, B. (2013). Multiple regression model for fast prediction of the heating energy demand. Energy & Buildings, 57, 302–312.

Chong, A., & Menberg, K. (2018). Energy & Buildings Guidelines for the Bayesian calibration of building energy models R. Energy & Buildings, 174, 527–547.

Connor, J., & Atlas, L. (2002). Recurrent neural networks and time series prediction. 5(2), 301–306. https://doi.org/10.1109/ijcnn.1991.155194

Fernandez, A. G. S., & Faustino-Gomez, J. S. (2006). Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks.

Geyer, P., & Schlüter, A. (2014). Automated metamodel generation for Design Space Exploration and decision-making – A novel method supporting performance-oriented building design and retrofitting. Applied Energy, 119, 537–556.

Goldstein, E. B., & Coco, G. (2015). Machine learning components in deterministic models: Hybrid synergy in the age of data. 3, 1–4.

Gratia, E. (2002). A simple design tool for the thermal study of an of ® ce building. 34, 279–289.

Han, J. M., Malkawi, A., & Gajos, K. Z. (2018). Eabbit 1.0: New Environmental Analysis Software for Solar Energy Representation. In 16th IBPSA International Conference and Exhibition, Rome.

Haykin, E. S. (2001). Recurrent neural networks for prediction: Wiley series in adaptive and learning systems. Image Processing, 4(2), 127–140.

Hygh, J. S., Decarolis, J. F., Hill, D. B., & Ranjithan, S. R. (2012). Multivariate regression as an energy assessment tool in early building design. Building and Environment, 57, 165–175.

Jaffal, I., Inard, C., & Ghiaus, C. (2009). Fast method to predict building heating demand based on the design of experiments. 41, 669–677.

Jakubiec, J. A., & Reinhart, C. F. (2011). DIVA 2.0: Integrating daylight and thermal simulations using rhinoceros 3D, DAYSIM and EnergyPlus. Proceedings of Building Simulation 2011: 12th Conference of International Building Performance Simulation Association, 2202–2209.

Kalogirou, S. (2000). Artificial neural networks for the prediction of the energy consumption of a passive solar building. Energy, 25(5), 479–491.

Mohandes, M., Rehman, S., & Halawani, T. O. (1998). Estimation of global solar radiation using artificial neural networks. Renewable Energy, 14(1), 179–184.

New, J. R., Ridge, O., & Parker, L. (2017). Constructing Large Scale Surrogate Models from Big Data and Artificial Intelligence Constructing Large Scale Surrogate Models from Big Data and Artificial Intelligence. (June).

Perez, R., Seals, R., & Michalsky, J. (1993). All-weather model for sky luminance distribution: Preliminary configuration and validation. Solar Energy, 50(3), 235–245.

Perez, R., Ineichen, P., Seals, R, & Michalsky, R. S. (1993). Article modeling daylight availability and irradiance components from direct and global irradiance. 44(5), 271–289.

Ritter, F. (2015). Simulation-based Decision-making in Early Design Stages.

Scherer, D. M. and S. (2015). VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition. IEEE/RSJ International Conference on Intelligent Robots and Systems, 922 – 928.

Simon, H. A. (2019). The sciences of the artificial. MIT press.

Singaravel, S., Suykens, J., & Geyer, P. (2018). Advanced Engineering Informatics Deep-learning neural-network architectures and methods : Using component- based models in building-design energy prediction. Advanced Engineering Informatics, 38(June), 81–90.

Voyant, C., Notton, G., Kalogirou, S., Nivet, M., Paoli, C., Motte, F., & Fouilloy, A. (2017). Machine learning methods for solar radiation forecasting: A review. Renewable Energy, 105, 569–582.

Yadav, A. K., & Chandel, S. S. (2014). Solar radiation prediction using artificial neural network techniques: A review. 33, 772–781.