AN URBAN BUILDING ENERGY SIMULATION METHOD INTEGRATING PARAMETRIC BIM AND MACHINE LEARNING

Forecast The Impact Of Surrounding Conditions On Solar Accessibility

JONG BUM KIM[1], SEONGCHAN KIM[2] and JAYEDI AMAN[3]
[1,3]*University of Missouri Columbia.*
[2] *Western Illinois University*
[1]*kimjongb@missouri.edu, 0000-0001-8762-9554*
[2]*s-kim@wiu.edu, 0000-0002-0780-383X*
[3]*jayediaman@mail.missouri.edu, 0000-0002-6128-8293*

Abstract. This research investigates a method of urban building energy simulation (UBES) by integrating Building Information Modeling (BIM), building simulation, and algorithm-based prediction to forecast the impact of surrounding conditions. In the urban context, building energy performances are determined not only by the individual building design but also by the building's surrounding context. Many energy performances are sensitive to outdoor and surrounding building conditions, such as neighbouring building volumes, heights, and spaces between buildings. However, such surrounding conditions were overlooked because they can exponentially increase the complexity of urban modeling and simulation. In that regard, the research sought to investigate a novel framework to take advantage of accurate performance simulations and algorithm-based fast predictions. This paper presents our UBES method implemented from three research phases: (i) building a parametric urban model in BIM to provide simulation inputs, (ii) creating a parametric simulation interface to produce training and validation data, and (iii) creating a prediction interface using a Support Vector Machine (SVR) algorithm. Lastly, the paper elaborates on the findings from the prediction results.

Keywords. Urban Energy Simulation, Solar Accessibility, Surrounding Conditions, Parametric BIM, Machine Learning, Support Vector Machine, Sustainable Cities and Communities

1. Introduction

This research investigates a method of urban building energy simulation (UBES) to forecast the impact of surrounding conditions on solar accessibility in urban development scenarios. The proposed method integrates parametric Building Information Modeling (BIM), building simulation, and Machine Learning (ML)-based prediction.

HUMAN-CENTRIC, Proceedings of the 28th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA) 2023, Volume 1, 665-674.

Substantial UBES approaches have explored building-physics-based simulation for accuracy and historical data-driven estimation for urban-scale expeditious prediction. They are distinctive in simulation cost, accuracy, scalability, level of detail, and predictability for unbuilt development scenarios. Previous studies presented that the inaccuracy or ambiguity in UBES originates from excessive abstraction and neglect of the surrounding condition effects in individual building simulation (Nutkiewicz et al., 2021).

The research focuses on the surrounding conditions in the urban context. Building energy performances are determined not only by the building design but also by the building's surrounding context. Many energy performances are sensitive to outdoor and surrounding conditions, such as neighboring building volumes, heights, and spaces between buildings. However, such surrounding conditions were simplified or not fully addressed in many UBES methods because they can exponentially increase the complexity of energy modeling. Consequently, this simplification could make the simulation results less applicable to the real world.

In that regard, the research seeks a novel framework to take advantage of accurate building simulations and algorithm-based predictions. This paper presents our UBES method implemented in three research phases: (i) building a parametric urban model in BIM to provide physical attributes of surrounding conditions, (ii) creating a parametric simulation interface to produce training and validation data, and (iii) creating a prediction interface using ML algorithms. First, the research collected the modeling information from the recently adopted urban design regulations, such as the urban design regulation variables and the geography information of the surrounding blocks. Second, the simulation module analyzed the solar accessibility of individual parcels, iterated a large number of simulations to test the surrounding conditions, and produced training and validation data for ML prediction. For ML, the research explored Linear Support Vector Machine (SVM), a widely applied algorithm for energy simulation and sensitivity analysis.

2. Research Context

2.1. SURROUNDING CONDITION SCENARIOS OF THE PLANNED BUILT ENVIRONMENT

Surrounding conditions in this research refer to heterogeneous physical conditions of individual blocks, parcels, and buildings that can impact the environmental context of individual sites. Surrounding conditions variate the environmental context, such as neighboring building height, setbacks, streetscapes, traffic lanes, landscape, etc. The energy performance in the urban context is determined by how individual buildings perform, whereas individual performances are influenced by their surrounding conditions and urban morphology.

For the most part, information on the existing conditions is collectible from a geospatial dataset or on-site data collection. When considering growing demands for urban retrofitting, urban planning and design guidelines are critical datasets, such as neighborhood master plans, comprehensive plans, and land-use and zoning regulations. This research collected the existing condition information from GIS and the planning information from Smart Growth regulations. Smart Growth regulations are fast-growing urban planning and design methods developed by U.S. Environmental Protection Agencies (EPA). Smart Growth regulations control building form and design with several tools. The Transect is a classification tool in Smart Growth to assign the type of design requirements to individual parcels and blocks. The idea is to provide diverse housing and workspace options with gradually changing building use, density, and design characteristics.

Figure 1 shows two development plans of the current Smart Growth regulations. Two examples show how Transect is used in the recent planning regulations and how it diversifies individual parcels' surrounding conditions. The block colors indicate the Transect types, and other annotation symbols designate the entrance locations, setbacks, front façade locations, etc. P1 and P2 in the left map have similar parcel dimensions and orientations, but they have different surrounding conditions, including different transects of south-facing parcels, the main traffic lane location, and the block orientations. P3 and 4 in the right-side map have opposite directions of the front façade. Even though two parcels belong to the same Transect and need to follow the same building design standards, their different surrounding conditions may change their environmental context.

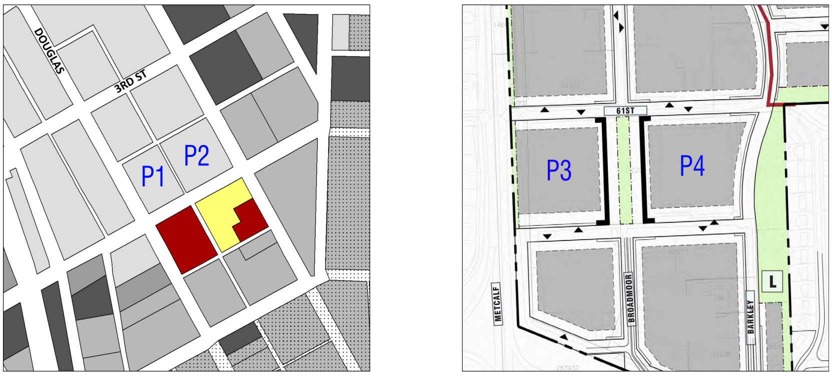

Figure 1 Various surrounding conditions in Smart Growth regulations. Notes: (Left) West Gateway Form-based Code, Kansas (Right) Lee's Summit Comprehensive Plan, Missouri

Transect-based planning promotes consistent styles and homogeneous functions within the same transect zones. However, where multiple transects are assigned, the pattern of surrounding conditions becomes more complex. In other words, the individual blocks within the same Transect have varying surrounding conditions. In UBES research, this random surrounding condition can quickly increase the range of what-if scenarios, making the modeling and simulation drastically time and effort-intensive or less viable. It increases complexity in UBES research because they yield bi-directional and dynamic energy flow between buildings, parcels, and blocks.

3. A UBES METHOD FOR SURROUNDING CONDITION SIMULATION

This research proposed a UBES method to address the surrounding conditions in energy modeling and simulation. The proposed method fused parametric BIM, physics-based simulations, and algorithm-based prediction to mediate the above problems and challenges. This research was built on our previous works on parametric BIM modeling, multi-criteria simulations, and immersive visualization (Kim et al., 2021).

- Parametric BIM: The research collected the modeling and geography information from the recently adopted urban design regulations and GIS data. Then, parametric urban models were built using spatial operations in BIM and a set of custom parameters that can change the BIM objects' geometry.
- Simulation iteration: Second, the simulation module analyzed solar accessibility of individual parcels, iterated a large number of parameters to test the surrounding conditions, and produced training and validation data for the ML prediction.
- SVM model: Lastly, for ML development, the research explored Linear SVM to perform prediction of the impact of surrounding conditions on the solar radiation.

The research implemented the method with parametric modeling in BIM and software prototype development. For parametric modeling, we used Autodesk Revit, its custom object modeling interface, and custom parameters to manipulate the object geometry. For energy simulation, we built simulation interfaces for solar radiation analysis. The research used crossover simulation tools such as Ladybug tools and TT Toolbox written in Grasshopper to take advantage of rich simulation interfaces and validated simulation engines. To connect Revit and Grasshopper, Rhino.inside was used. For algorithm-based prediction, the research explored SVM with an open-source library such as scikit-learn (Pedregosa et al., 2011.) and Python scripting.

This research explored SVM for regression analysis using the training data obtained from physics-based energy simulations. The parametric BIM modeling, simulation module creation, and parametric BIM/ simulation integration were built on our prior works. The following sections describe the methods and processes of the SVM model creation.

3.1. SVM FOR ENERGY SIMULATION

For years, machine-learning approaches have been tested in the energy simulation sector (Chen et al., 2020). Support Vector Machine (SVM), developed by Vapnik with colleagues in 1995, is a supervised learning technique for data classification and regression analysis. Two major applications of SVMs are data classification and regression analysis. In classification analysis, SVMs assign a given dataset to the binary linear classifier, distributes data points in high-dimensional space, and maximize the distance (margin) between the classified data groups. In regression, SVMs map the input dataset into kernel-induced feature space to place the data points inside a tube space of a given radius. The kernel is an essential concept of SVMs. When the dataset in low-dimensional space does not present linear separable patterns, SVMs can transform this low-dimensional space into high-dimensional space. These mapping rules between two spaces are the kernel function. Widely used functions include linear, nonlinear, polynomial, radial basis, and sigmoid. The selection of kernel functions considerably affects the SVM results.

Because of powerful data classification capabilities, SVMs have been widely used in text categorization, handwriting character recognition, bioinformatics, etc. In addition, SVMs are increasingly used in building energy simulations. Example areas include building energy consumption prediction (Dong et al., 2005; Jung et al., 2015), annual electric load forecast (Azadeh et al., 2008), estimating heating energy (Paudel et al., 2014), PV system performance (Ogliari et al., 2013), and wind power system (Chen et al., 2013). Studies show that SVMs can solve nonlinear problems with a small number of training data (Zhao et al., 2012). Other studies presented that SVM is more capable than different algorithms such as Artificial Neural Networks (Li et al., 2010). Several studies described SVR's benefits over decision tree (DTR) and random forest (RFR). Ma et al. (2019) claimed that the SVR could correctly predict building energy use with a significant nonlinear capacity, improving accuracy and stability in energy performance forecasting. Furthermore, Others explained that the SVM outperforms other techniques when limited samples are available (Chen et al., 2020).

SVR's regression function is summarized as follows:

$$f(x) = w \cdot \phi(x) + b$$

Where f(x) is the forecasting value, x represents the input variable, w represents the weight coefficient, b represents the deviation value, and (x) represents the high dimensional feature space. In addition, we used the R-squared value and mean absolute error (MAE) to evaluate the SVR model as follows:

$$R^2 = 1 - \frac{\sum_{i=1}^{n} |y_i - \hat{y_i}|^2}{\sum_{i=1}^{n} |y_i - \bar{y}_i|^2}$$

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y_i}|$$

Where y_i is the simulation results, (y_i)^ indicates the prediction results, y ̅_i is the average of the simulation results, and n represents the sample numbers of the testing dataset.

3.2. DATA FLOW

Figure 2 explains the workflow of surrounding condition simulations using the SVR model. First, we produced inputs, including regulation variables as IVs and energy simulation results as DVs. This research simulated the solar radiation levels from a building-physics-based simulation (S1). The input dataset was then split into training

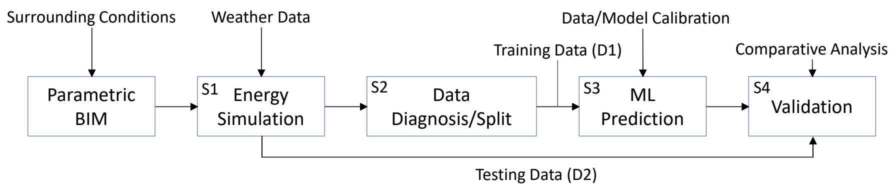

Figure 2. Workflow of the surrounding condition simulation

(D1) and testing datasets (D2). In this phase, the data process was performed to normalize the variables before the SVR model building. Then we fed the training IVs and DVs into the training module (S3). The SVR then projected the testing IVs dataset into kernel-induced feature space. Lastly, the SVR results were compared with the testing dataset selected from simulation results (S4) for validation.

4. Case Study

The research performed case studies of two study areas with different surrounding conditions. Two sites were selected from the Downtown Form District of Overland Park Unified Development Plan. The City of Overland Park is a southern neighborhood of the Kansas City metropolitan area. The City adopted the master plan and Smart Growth regulations to revitalize the downtown area and the Metcalf corridor, a historical corridor connecting southern cities with the Kansas City downtown area. First, we analyzed the regulating map, transect types, and building design standards. Then we built two block models using Revit and its custom modeling interface. GIS data provided the district subdivision information. The building design standards in the regulation provided modeling information of the building sections (Table 1).

Parameters	Transect 1	Transect 2	Transect 3
Building height	7.3m~18.2m	7.3m~18.2m	7.3m~14.6m
Building depth	18.2m~21.3m	18.2m~21.3m	15.2m~18.2m
Setback	1.5m~3m	1.5m~3m	1.5m~3m
Roof height	2.7m~3.6m	2.7m~3.6m	2.7m~3.6m

Table 1. Regulation variables of three transects for parametric BIM.

Figure 3 shows two block models showing different subdivision shapes and surrounding conditions. The regulating map in Smart Growth indicated the position of the building front façade, street types, setback requirements, etc. The building design standards in Smart Growth regulations showed building use, height, roof shape, and other sectional configurations. The street color indicates the streetscape types and the building types. The maps indicate the allowed building front façade location, making

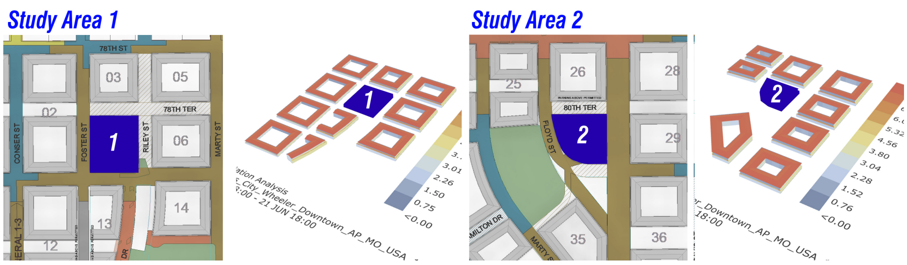

Figure 3. Regulating plans and solar radiation maps of two study areas

the building masses similar to the parcel shapes.

Next, the simulation interface iterated parameter sets and ran 500 simulations for each case. We simulated solar radiation values using the Revit district models and Ladybug tools. The resulting dataset consisted of input variables as IVs and solar radiation values per parcel as DV. The inputs include building height, depth, setback, and roof pitch. The system interface written in Python redefined the result datasets and crosschecked the missing values for data pre-processing.

In the final phase, we split the result datasets randomly into 80% for training and 20% for testing before the SVR model training. For each case, 500 independent data points were divided into 400 for training and 100 for testing. Lastly, we fed the training and testing data into the SVR kernel and produced ML-based prediction models.

5. Results and Findings

The research implemented several calibration methods to improve the reliability and accuracy of input and output of prediction models, including random state, learning curves, and K-fold cross-validation. Then, a comparative study was performed between the simulation results and the ML-prediction results.

Input data diagnosis: To diagnose the simulation dataset, we analyzed multicollinearity, normality, homoskedasticity, and independence. Standardized Regression Coefficients showed that the building depth and setback have the most substantial effect on solar accessibility.

Random State: We used the random permutation function to select the reliable training and data sets from the simulation results. The function generated the pseudo-random number of the selected datasets, calculated the regression coefficient, and selected the most reliable dataset. The pseudo-code made the selected datasets not only trackable but also reusable. The selected dataset presented significance (Case1: R2adj=0.979, Case2: R2adj=0.977).

Learning Curves: Learning curves are used to plot the error of the training set and validation set as a function of training size to measure the model's generalization performance. Overfitting occurs when a model performs well on training data but poorly on validation data. In contrast, if it does poorly on both, it is underfitting. The learning curve for Case 1 is presented on the left, and the learning curve for Case 2 is presented on the right in the figure 4. As seen in the learning curve plots, the line of the training set begins at zero. As additional instances are added to the training set, the model's ability to fit the training data worsens. As a result, the error on the training data increases until it approaches a plateau indicating additional instance has no effect on the average error. For the validation set, when the model is trained on a few instances, it cannot generalize appropriately, so the validation error is initially greater than the training set. The model then learns as more training instances are provided, and validation error gradually decreases until it reaches a plateau close to the training set. The two learning curve plots demonstrate that the SVM models used in this study generalize adequately, without overfitting or underfitting issues.

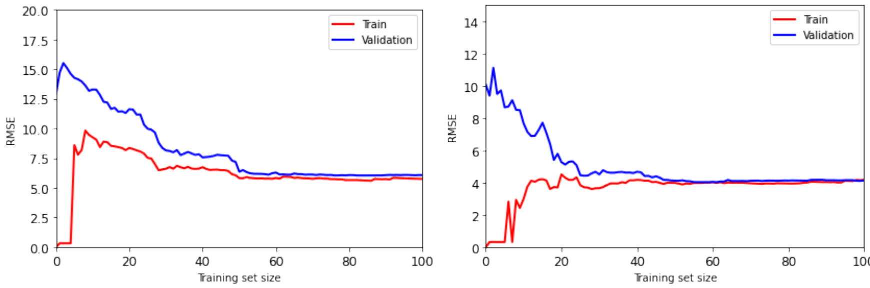

Figure 4. Learning cures (Left: Study area1, Right: Study area2)

K-fold cross-validation: In addition, we conducted K-fold cross-validation to compare and select a regression model with a lower bias. In K-fold, the given dataset is split into a K number of folds. In the first iteration, the first fold is used for testing, and the rest are used for training. In the second iteration, training is done with the second fold data. Then, the cross-validation process is repeated k times, with each of the k subsamples used exactly once as the validation data. The k results can then be averaged to produce a single estimation. The dataset was randomly shuffled using the "Gamma" and "Epsilon" hypermeters of the SVR model. Seven gamma (¥) and six epsilon (€) values together with 42 possibilities resulted in 420 fits of the sample sets. For accuracy testing, we employed R-square and root mean square deviation (RMSE). In case 1, the most fitted ¥ value was 0.1, whereas, in case 2, the most fitted value was 0.5. The € value that fit both instances the best was 1. The advantage of this method over repeated random sub-sampling is that all observations are used for both training and validation, and each observation is used exactly once. 10-fold cross-validation is commonly used [16], but in general k remains an unfixed parameter.

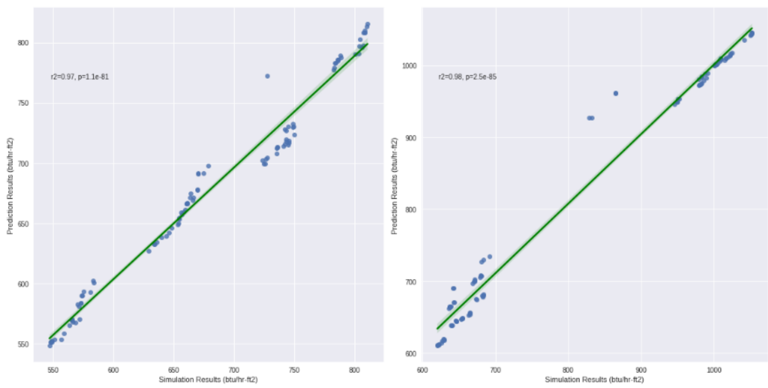

Figure 5. Regression analysis (Left: Study area 1, Right: Study area 2)

Comparative Analysis: Finally, we carried out regression analysis of simulation and prediction results in Figure 5. Strong correlation is observed (r2 is 0.97 and 0.98

respectively), while several datasets presented weak correlations. Table 2 shows a subset of samples to illustrate the percentage variance in the analysis results. Each sample contains 12 IVs as well as solar radiation results generated by the physics-based simulation and ML-based prediction model. For both cases, very few samples show deviations greater than 1%. The results demonstrate the high accuracy and reliability of the proposed method.

Study area 1				Study area 2			
Case No.	Simulation (btu/hr-ft2)	ML-prediction (btu/hr-ft2)	Deviation (%)	Case No.	Simulation (btu/hr-ft2)	ML-prediction (btu/hr-ft2)	Deviation (%)
103	810.19	814.44	0.52	178	682.29	678.77	-0.52
219	640.69	634.51	-0.97	360	1024.45	1017.45	-0.69
422	546.84	549.68	0.52	128	673.83	675.25	0.21
111	668.55	675.29	1.00	144	1041.23	1035.64	-0.54
373	574.18	590.43	2.75	203	1000.86	1000.61	-0.02

Table 2. A comparison of selected samples from simulation and prediction

6. Conclusion

In this paper, we investigated the UBES method to handle many simulation scenarios associated with the surrounding conditions. The proposed approach attempted to strike a balance between the accuracy and the reliability of simulations, simulation size and latency, and the level of detail and scale. To do so, we examined the fusion of parametric BIM, simulations, and ML with SVR.

Algorithm-based prediction can handle a large number of district layout scenarios with less effort and computing time, particularly when multi-criteria performances in urban development are considered. As a result, the substitution of simulations with ML predictions will yield significant benefits in saving cost and time. The software prototype outperformed physics-based parametric simulations in terms of reliability, capacity, and speed. The heterogeneous connection between modeling and simulation platforms required high computing power, but it eliminated redundant and error-prone input processes by using BIM information. The case study tested small-scale urban blocks. In reality, there are infinite potential layout patterns because of the curse of dimensionality and numerous provisions in Smart Growth regulations. As a result, the proposed method still has limitations in generating optimal development scenarios to minimize the environmental impacts.

The simulation results demonstrated that the surrounding conditions significantly impact each block's solar accessibility, which needs to be considered in urban energy modeling and simulations. From case studies, we inferred that SVR predicts different solar radiation patterns according to each parcel's surrounding conditions. However, it is still required to address "black box effect" of algorithm-based prediction. This research used the linear kernel, but it is suggested to investigate different kernel types. In addition, we noticed that SVR is sensitive to the sample size and the sampling data sets. Potential overfitting issues and sensitivity in sampling need to be addressed in future research to gain the validity and credibility of SVR-based surrounding condition predictions.

References

Azadeh, A. & Ghaderi, S.F. (2008). A simulated-based neural network algorithm for forecasting electrical energy consumption in Iran. Energy Policy. 36. 2637-2644. https://doi.org/10.1016/j.enpol.2008.02.035

Chen, J., Zhu, J., Wang, Z., Zheng, X., & Zhang, B. (2013). Scalable Inference for Logistic-Normal Topic Models. NIPS.

Wu, M., Liao, J., Yu, L., Lv, R., Li, P., Sun, W., Tan, R., Duan, X., Zhang, L., Li, F., Kim, J., Shin, K. H., Seok Park, H., Zhang, W., Guo, Z., Wang, H., Tang, Y., Gorgolis, G., Galiotis, C., & Ma, J. (2020). 2020 Roadmap on Carbon Materials for Energy Storage and Conversion. Chemistry, an Asian journal, 15(7), 995–1013. https://doi.org/10.1002/asia.201901802

City of Overland Park, Kansas (2016). Downtown Form-Based Code. Retrieved from http://online.encodeplus.com/regs/overlandpark-ks/doc-viewer.aspx#secid-7075

Dong, B., Cao, C., & Lee, S.E. (2005). Applying support vector machines to predict building energy consumption in tropical region. Energy and Buildings, 37(5), 545-553. https://doi.org/10.1016/j.enbuild.2004.09.009.

Jung, H.C., Kim, J.S., & Heo, H. (2015). Prediction of building energy consumption using an improved real coded genetic algorithm based least squares support vector machine approach. Energy and buildings, 90, 76–84. Elsevier B.V. https://doi.org/10.1016/j.enbuild.2014.12.029.

Kim, J. B., Aman, J., & Balakrishnan, B. (2021). Forecasting performance of Smart Growth development with parametric BIM-based microclimate simulation. Projection, Proceedings of the 26th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA) 2021, Hong Kong. https://doi.org/10.52842/conf.caadria.2021.1.411

Li, R., Chen, X., Balezentis, T., Streimikiene, D., & Niu, Z. (2021). Multi-step least squares support vector machine modeling approach for forecasting short-term electricity demand with application. Neural Computing and Applications, 33(1), 301–320. https://doi.org/10.1007/s00521-020-04996-3

Ma, Z., Ye, C., Li, H., & Ma, W. (2018). Applying support vector machines to predict building energy consumption in China. Energy Procedia, 152, 780–786. https://doi.org/10.1016/j.egypro.2018.09.245

Nutkiewicz, A., Choi, B., & Jain, R. K. (2021). Exploring the influence of urban context on building energy retrofit performance: A hybrid simulation and data-driven approach. Advances in Applied Energy, 3, 100038. https://doi.org/10.1016/j.adapen.2021.100038

Ogliari, E.; Grimaccia, F.; Leva, S.; Mussetta, M. Hybrid predictive models for accurate forecasting in PV systems. Energies 2013, 6, 1918–1929. https://doi.org/10.3390/en6041918

Paudel, S., Elmtiri, M., Kling, W.L., Corre, O.L., & Lacarrière, B. (2014). Pseudo dynamic transitional modeling of building heating energy demand using artificial neural network. Energy and Buildings, 70, 81–93. https://doi.org/10.1016/j.enbuild.2013.11.051.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, & M. Blondel. (2011). "Scikit-Learn: Machine Learning in Python." Journal of Machine Learning Research 12: 2825–2830.

Zhao, H., & Magoulès, F. (2012). A review on the prediction of building energy consumption. Renewable and Sustainable Energy Reviews, 16(6), 3586–3592. https://doi.org/10.1016/j.rser.2012.02.049