

Research Article

Cite this article: Rahimian M, Duarte JP, Iulo LD (2021). A real-time predictive software prototype for simulating urban-scale energy consumption based on surrogate models. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* **35**, 353–368. <https://doi.org/10.1017/S0890060421000184>

Received: 12 May 2021

Revised: 23 August 2021

Accepted: 31 August 2021

First published online: 28 December 2021

Key words:



Artificial neural networks; simulation; surrogate modeling; urban-scale energy modeling

Author for correspondence:

Mina Rahimian,

E-mail: mxr446@psu.edu

A real-time predictive software prototype for simulating urban-scale energy consumption based on surrogate models

Mina Rahimian¹ , Jose Pinto Duarte²  and Lisa Domenica Iulo³

¹Stuckeman School of Architecture and Landscape Architecture, The Pennsylvania State University, University Park, PA, USA; ²Stuckeman Center for Design Computing, Stuckeman School of Architecture and Landscape Architecture, The Pennsylvania State University, University Park, PA, USA and ³Hamer Center for Community Design, Stuckeman School of Architecture and Landscape Architecture, The Pennsylvania State University, University Park, PA, USA

Abstract

This paper discusses the development of an experimental software prototype that uses surrogate models for predicting the monthly energy consumption of urban-scale community design scenarios in real time. The surrogate models were prepared by training artificial neural networks on datasets of urban form and monthly energy consumption values of all zip codes in San Diego county. The surrogate models were then used as the simulation engine of a generative urban design tool, which generates hypothetical communities in San Diego following the county's existing urban typologies and then estimates the monthly energy consumption value of each generated design option. This paper and developed software prototype is part of a larger research project that evaluates the energy performance of community microgrids via their urban spatial configurations. This prototype takes the first step in introducing a new set of tools for architects and urban designers with the goal of engaging them in the development process of community microgrids.

Introduction

In “Redesigning the Metropolis the Case for a New Approach” (1989), Johnathan Barnett discusses the dynamic evolution of cities and urban areas. In this paper, Barnett emphasizes the necessity of updating urban design and planning techniques as cities face constant changes in their environmental, developmental, and political settings. Taking the environmental aspect of built environments into consideration along with rapidly changing climatic conditions (Andreson and Bausch, 2006), researchers suggest that reaching energy self-sufficient built environments requires moving past building-scale analysis and adopting up-to-date and innovative energy-related measurements for urban design and planning (Gossop, 2011; Davila *et al.*, 2016).

Addressing energy issues at an urban scale brings more complexity than those of a single building. This is mainly due to the wide range of stakeholders, as well as the large number of energy-relevant variables and features involved in urban-scale projects. With this comes extended and obscurant power relations making urban issues ill-defined and multi-faceted, especially when it comes to the inherently political nature of energy-related issues. In an era where the causes and effects of climate change have been a topic of dispute among politicians and scientists, the goal of reaching low carbon and energy self-sufficient communities and cities has become more urgent than before¹. Research and planning communities have come to the conclusion that a new understanding of how urban planning impacts the energy dynamics in cities and communities is required (Cajot *et al.*, 2017). Reaching low carbon and energy self-sufficient communities entails reducing fossil fuel consumption and combating greenhouse gas emissions by taking actions in pursuit of building resilient communities and cities which are less pollutant and less energy demanding. A main action item for reaching this goal is the development of community microgrids which support the local supply and demand of clean energy in neighborhood-scale urban settlements (Amin and Wollenberg, 2005; Farhangi, 2010). To develop low carbon and energy self-sufficient community microgrids, the role of urban planning in improving the energy performance of these power-grid-independent territories is considered essential. This is done by adding a spatial dimension in evaluating the energy performance of community microgrids, a topic that previously has been only marginally addressed in the research and academic communities but has remained intact in practice.

© The Author(s), 2021. Published by Cambridge University Press. This is an Open Access article, distributed under the terms of the Creative Commons Attribution licence (<https://creativecommons.org/licenses/by/4.0/>), which permits unrestricted re-use, distribution, and reproduction in any medium, provided the original work is properly cited.

Problem statement

The US department of energy refers to microgrids as a “group of interconnected loads and distributed energy resources within clearly defined electrical boundaries that acts as a single controllable entity with respect to the grid. A microgrid can connect and disconnect from the grid to enable it to operate in both grid-connected or island-mode” (Ton and Smith, 2012). Different types of microgrids exist depending on their use and purpose of development. Community microgrids [in this study] are neighborhood-scale microgrids contextualized in cities and urban areas which are typically developed for environmental purposes and their loads consist of various mixes of residential and nonresidential buildings.

With this definition, community microgrids are essentially *building-integrated energy systems*. As with any other *energy system*, the efficiency of a community microgrid’s energy performance is evaluated by comparing the energy that is inputted into the system to the energy that is outputted from the system (Fernandez and Blumsack, 2010). The energy input comes from on- and off-site sources of energy, and the energy output is in the form of the energy used for building operations, the stored energy in microgrids’ storage devices, and the wasted energy during the conversion or distribution process. With community microgrids being *energy systems*, improving their energy performance has been typically associated with advocating *technological advances* enhancing the limited supplies of local energy and addressing the constantly growing demands of the loads² (Siderius, 2004; Wouters, 2015). However, an exclusive focus on technologically improving a community microgrid’s *infrastructure* without considering the characteristics of its *superstructure* limits a holistic understanding of these energy systems being *building-integrated*.

This research argues that for a comprehensive understanding of energy performance in community microgrids, one needs to study and evaluate its superstructure – as of the spatial characteristics of its urban form – in addition to the technicalities of its infrastructure. This argument is supported by two sets of studies:

- The first set are those studies that claim focusing on technological innovations alone wouldn’t solve the current energy issues in the built environment. These studies verify that despite the high rates of technological enhancements in energy systems, per capita energy consumption specifically in residential buildings has been gradually increasing since the 1980s (Ewing and Rong, 2008).
- The second set of studies are those from 1960s onwards which have studied the impact that urban form has on the energy required for space heating and cooling in buildings, as well as the feasibility of adopting on-site renewable energy generators such as Photovoltaic (PV) panels and wind turbines in urban areas (Steadman, 1977; Owens, 1986; Grosso, 1998; Steemers, 2003; Cajot *et al.*, 2017).

The motivation behind this research is to understand why the spatial dimension of improving community microgrids’ energy performance is currently being neglected in practice despite all the researches that support this argument. More broadly, what is causing the existing practical disengagement of the design sector in the development process of community microgrids?

²A high-energy performance community microgrid ensures the extended duration of energy self-sufficiency while not receiving energy supply from the larger power grid.

Focus of this paper

The results of our research suggest two main reasons driving the above-mentioned disengagement:

1. The first reason is the dearth of information that architects and urban designers need to comprehend on how urban form and geometry impact the energy performance of cities at large. Typically, in practice, the energy implications of individual buildings are considered and those of urban scales are easily neglected. This could be due to the spatial complexity associated with urban form and the fact that studies to date were not able to capture the multidimensional influence of urban form on community-scale energy performance. Measuring and understanding the energy implications of an urban area, unlike common knowledge, goes beyond summing up the energy performance of each individual building within a certain regional boundary, hence the associated complexity with urban-scale energy studies. Researchers suggest that the spatial characteristics of urban form have a major impact on changing the local wind patterns and trapping heat in urban areas and therefore creating micro-climates (Santamouris *et al.*, 2001; Reinhart *et al.*, 2013; Chatzidimitriou and Yannas, 2015; Silva *et al.*, 2017). Therefore, when considering the energy implications of urban areas, the existing micro-climates, as well as the energy performance of individual buildings, need to be studied. Adding the feasibility study of accessing renewable energy to this relational pattern brings the understanding of how urban form impacts energy performance in community microgrids to another level of complexity. This, previously, has not been offered to the building and urban design communities.
2. The second reason, concluded by our research, is the lack of urban-scale energy modeling and simulation tools that capture the presented complexity (Rahimian *et al.*, 2018). Existing urban-scale energy simulation tools use the summed amount of each building’s energy performance within a region as the value of energy performance of the urban region of study. Based on the reasonings provided above, this returned value is not an accurate reflection of the real-world energy performance of an urban area. Additionally, the hardcoded backend of these tools makes running urban-scale energy simulations a tedious task and an expensive computation, practically impossible to use for real-world projects.

In a recent publication by the authors of this paper, the first reason for this disengagement has been thoroughly discussed (Rahimian *et al.*, 2020). The analysis offered in this paper captures the complexity of urban form by understanding the multidimensional impact that the spatial structure of urban form has on the amount of energy consumed for community building operations. Choosing San Diego County as a case study, this paper described benefiting from artificial neural networks (ANN) to factor in the multitude spatial dimensions of urban form, and to explore their combined effect on community-wide net energy consumption. To do so, 19 energy-relevant indices of urban form were selected from past studies and measured for all zip codes in San Diego County along with their monthly values of energy consumption, from 2012 to 2018, which were acquired through the county’s utility company, SDG&E. Inference on the resulting predictive model was done using Shapley values showing that the most influential characteristics of urban form on energy consumption are related to compactness, passivity, shading, and diversity of a

community in the context of the case study. Through this work, the strategic role of developing energy-efficient urban settlements by looking beyond building-scale energy efficiency had been acknowledged.

This paper, however, takes the above-mentioned study to the next stage by addressing the second reason for the described disengagement. In the current paper, a solution is suggested for developing an urban-scale energy simulation tool that not only captures the multidimensionality of measuring community-scale energy consumption but also reduces the computing time to close to real time.

Aim and significance: surrogate models for running urban-scale energy simulations

The adoption of computer simulation codes and programs has been extensively used in many science and engineering fields as a flexible way to study complex and real-world phenomena under controlled environments (Gorissen *et al.*, 2010). Compared to expensive physical experimentations, computer simulations are economically more efficient and improve the quality of engineered products and services. However, the downside of simulation activities, especially when the problem of interest reaches a certain level of complexity, is the high cost of computing a simulation that may take hours and days to perform (Forrester *et al.*, 2008).

An example of a complicated phenomenon for simulation purposes is measuring the building and urban-scale energy performance where the influence of various internal and external factors needs to be considered. These factors include but are not limited to environmental factors, such as weather conditions, building-specific factors, and characteristics such as materials and structure, building operational factors and components such as HVAC system, as well as occupational factors such as the energy behavior of the building occupants. For simulating a complex phenomenon as such, researchers suggest three different methods for measuring energy performance in the built environment including (Magoules and Zhao, 2016; Silva *et al.*, 2017; Seyedzadeh *et al.*, 2018)

- *Engineering methods or white-box models*: which use engineering and physics-based principles to calculate the energy performance of the built environment at different scales. The basis of this method is to precisely calculate the thermal dynamics and physical performance of buildings based on their structural and operational characteristics, environmental factors, and sublevel building components which ultimately results in complex mathematical models.
- *Statistical methods or grey-box model*: which combine physical and engineering methods with data-based, statistical modeling (Tardioli *et al.*, 2015). Gray-box models usually have very particular analysis methods including linear correlation, regression analysis, stepwise regression analysis, logit models, ANOVA, *t*-test, factor analysis, panel data, structural equation models, and cross-tabulation (Silva *et al.*, 2017) with the aim of correlating energy indexes with influencing variables.
- *Data mining methods or black-box models*: which extract “implicit, previously unknown, and potentially useful knowledge from data” by “applying data analysis and discovery algorithms that produce a particular enumeration of patterns (or models) over the data” (Fayyad *et al.*, 1996; Tsui *et al.*, 2006). Data mining techniques originated a branch named machine

learning which is the “science and art of programming computers so that they can learn from data” (Géron, 2017) in which “the ‘machine’ is able to identify and generalize patterns” from large datasets without being explicitly programmed (Samuel, 1959; Chen *et al.*, 2000; Silva *et al.*, 2017)

Over the past 60 years, hundreds of building energy simulation programs have been developed utilizing engineering methods (white-box models) with different levels of complexity depending on the number and type of parameters they incorporate for running energy performance measurements (Crawley *et al.*, 2008). Common to all existing energy performance tools – whether it runs analysis at the building scale or city scale, is stand-alone such as TRNSYS³, or integrated into computational design workflows such as Ladybug tools⁴ and DIVA⁵ – is that their backends are all hardcoded physical laws for the derivation of building energy performance (Crawley *et al.*, 2008; Magoules and Zhao, 2016; Tamke *et al.*, 2018). Although the simulation results of these tools are effective and accurate, in practice, they bear some difficulties: firstly, they require lots of input parameters about the building and its environmental context which might not be accessible to all users. Secondly, the hardcoded backend results in extremely time-consuming computing processes make running simulations a tedious task to perform. This is especially accurate when simulating buildings’ energy performance at a city scale. In such cases, enormous amounts of time and resources need to be dedicated to creating building energy models for hundreds or thousands of buildings across a city in order to run urban-scale energy simulations effectively.

Due to these reasons, researchers have utilized methods other than physics-based and engineering methods to estimate building energy performance in a less time- and resource-consuming way. One way to deal with this problem is to construct simpler approximation models in order to develop a relationship between input and outputs and predict performance. When the “simpler approximation model” is properly constructed, it could mimic the behavior of a simulation program quite accurately while being computationally cheaper for running evaluations. Different methods exist for constructing such approximation models. This paper is focused on the use of compact surrogate models, also known as metamodels (Simpson *et al.*, 2008), which are data-driven approaches that incorporate either statistical analysis or machine learning methodologies capable of mimicking the behavior of a simulation program.

In a statistical approach, buildings’ historical data are used to run statistical analysis to correlate energy performance with oversimplified variables and predict future performance. In a machine learning approach, which some argue may fall under statistical methodology (Seyedzadeh *et al.*, 2018), computer algorithms are trained to learn from data without being explicitly programmed. In the paper “Machine learning for estimation of building energy consumption and performance: a review” (2018), the authors provided a sufficient review of the application of different machine learning techniques in forecasting building energy performance such as the use of ANN, support vector machines, Gaussian-based regression, and clustering. The authors conclude that traditional building energy modeling and forecasting using engineering methods are not fast enough to meet the demands of

³<http://www.trnsys.com/>

⁴<https://www.ladybug.tools/>

⁵<https://www.solemma.com/diva>

decision-makers and therefore are not as frequently used by professionals as could be expected. This is while machine learning models have shown great potential for predicting building- or city-scale energy performance quickly and accurately.

Most research on using machine learning for predicting building energy performance has been carried out in engineering as reviewed by Seyedzadeh *et al.* (2018). However, the field of architecture has also been observing the benefits of machine learning for design and for advancing performance-based decision-making in design. In the paper “Machine Learning for Architectural Design: Practices and Infrastructure”, Martin Tamke *et al.* (2018) discusses the different potentials that machine learning can bring into architectural practices throughout the design and construction process. Currently, most research on the applications of machine learning in architecture has been focused on image-based design generation and shape recognition such as in the work of Hu *et al.* (2020), Chaillou (2019), and Huang and Zheng (2018). The value and importance that Tamke *et al.* (2018) brings into play are suggesting emergent practices in architecture that can benefit from machine learning in creative use and synthesis that goes beyond current design generation utilization. Five novel and practical streams of applications are suggested by the authors, including using “machine learning for short-circuiting simulation”.

As the term suggests, trained machine learning-based surrogate models can be used for simulating building energy performance rapidly in a time frame very close to real time. Short-circuiting buildings’ performance simulation – whether it is energy, structural, mechanical, thermodynamics, or other – can help with understanding the behavior of architecture well before its construction. Since current energy performance simulation tools are based on physical laws, running these simulations is time consuming and computationally intensive and therefore is not often used within the design process and as a tool to drive design decisions. Using trained surrogate models as the backend of such tools for predicting simulation results, in a very short amount of time can aid with integrating simulation into the design process and advancing performance-based design methodologies.

In this paper, we provide a proof-of-concept software prototype that benefits from trained machine learning models as its backend. The development of an urban-scale energy simulation software prototype is described herein that predicts the monthly value of the energy consumed by any inputted community design scenario by evaluating and measuring its urban form. Unlike existing building and urban-scale energy simulation tools, this prototype does not operate on a hardcoded backend that takes *hours* to run; it rather uses predictive trained models to estimate the monthly value of energy consumption for any designed community scenario in *real time*.

Methodology

This section describes the different steps taken for developing the described software prototype based on surrogate models. This prototype has been developed for Rhinoceros⁶ which is a common 3D modeling tool extensively used among architects and designers. The tool takes 3D community design scenarios and instantly outputs predictive estimations of the design’s monthly energy consumption. Therefore, there are two aspects to this tool: a user-facing component in which the 3D communities

are designed, and a simulation engine that outputs the energy consumption values.

The next three sections describe the preparation of these two components of the prototype and how they’re merged towards a functional prototype.

A machine learning-based simulation engine

The dataset in which the machine learning models were trained entails measurements of urban form and monthly energy consumption values for 110 zip codes in San Diego county. Part of the dataset is shown in Figure 1; each row has the zip code number, followed by 19 numbers representing the urban form⁷, then 12 monthly values of energy consumption and then total energy consumed in one year. This is repeated seven times for each zip code each row representing one year of data from 2012 to 2018. More information on how this dataset has been processed and cleaned can be found in “A Machine Learning Approach for Mining the Multidimensional Impact of Urban Form on Community Scale Energy Consumption in Cities” by the authors (Rahimian *et al.*, 2020).

The goal of the proposed software prototype is to use any inputted 3D community design scenario and estimate its monthly values of energy consumption. This requires having ANN trained on monthly values of energy consumption and using the resulting predictive models as the backend of the software prototype. The approach used here is to train separate neural networks for each month of the year; the reason behind this logic is to avoid unnecessary complexity by training one neural network over a dataset with 12 months as outputs. In this regard, 12 subsets of the dataset were extracted – each has the same predictor variable (19 indices of urban form), but the response variable changes based on the specific month of the year on which the dataset has the energy information.

After establishing the 12 datasets, the next step was to train specific machine learning models on the datasets. ANN were used as the specific machine learning method to model and forecast community energy consumption. Finding the best performing artificial neural network architecture is an empirical process (Rahimian *et al.*, 2020). Usually, different architectures are tested and the one which yields the best accuracy is selected. Empirically finding the best performing artificial neural network for each month of the year was a time-consuming procedure. As a solution for simplifying this process, we developed a code for performing a grid search operation so that for each month’s dataset the algorithm tests different artificial neural network architectures and selects the ones with the highest performance (highest performance means lowest mean squared error). The variables instituting the different permutations of the grid search were:

- *Optimizer*: adam, nadam
- *First layer size*: 512, 1,024, 2,048
- *Number of layers*: 6, 7, 8

Some parameters along with their values were also selected to be used throughout all the neural network variations. For example, the activation function for all neural networks was set to Relu, dropout layers were added at the rate of 0.2⁸, number of

⁷It is assumed that in all these 7 years of study, the urban form has majorly remained the same.

⁸Dropout was used on the first hidden layer as a regularization method to avoid potential overfitting due to the small size of the datasets.

⁶<https://www.rhino3d.com/>

	Zipcode	Floor Space Index	Ground Space Index	Open Space Ratio	Layer	Network Density	Aspect Ratio	Volumetric Compactness	Size Factor	Form Factor	...	kwh-04	kwh-05	kwh-06	kwh-07	kwh-08	kwh-09	kwh-10	kwh-11	kwh-12	kwh-total
0	91901	0.001205	0.000432	0.999568	5.420000e-10	0.000359	487.563149	0.991402	420.056228	416.444609	...	1628495	1696036	1883073	2525349	3142081	2818045	2358746	2573347	3387368	23943788
1	91901	0.001205	0.000432	0.999568	5.420000e-10	0.000359	487.563149	0.991402	420.056228	416.444609	...	3003970	3354659	3620557	4408138	4648182	4520802	5704204	6063564	6933314	51574865
2	91901	0.001205	0.000432	0.999568	5.420000e-10	0.000359	487.563149	0.991402	420.056228	416.444609	...	5842737	8013901	8571344	10996885	11028397	12137559	9076297	8353464	9802152	100882572
3	91901	0.001205	0.000432	0.999568	5.420000e-10	0.000359	487.563149	0.991402	420.056228	416.444609	...	7582986	7945450	9317607	9713128	11549307	10527516	2430824	2291552	2716079	88881788
4	91901	0.001205	0.000432	0.999568	5.420000e-10	0.000359	487.563149	0.991402	420.056228	416.444609	...	907845	946058	1196945	11337912	11261558	8806578	9327471	8418730	9901345	76243126
5	91901	0.001205	0.000432	0.999568	5.420000e-10	0.000359	487.563149	0.991402	420.056228	416.444609	...	6895892	7608524	8983617	11717806	11301569	9665343	8782920	9192398	8680018	109078731
6	91901	0.001205	0.000432	0.999568	5.420000e-10	0.000359	487.563149	0.991402	420.056228	416.444609	...	3237940	3386157	3695892	12251954	11544396	9038465	3845338	4310910	5756585	101844061

Fig. 1. The first several rows of the dataset show urban form and energy consumption data for 7 years for one zip code in San Diego.

epochs were set to 300, and patience value⁹ was set to 100. Note that the choice of the static parameters, as well as the different permutable variables, were all based on the empirical training experience gained in our previously published paper (Rahimian *et al.*, 2020). With this setting for each of the 12 datasets, 2*3*3 = 18 different artificial neural network architectures were trained, and the top 5 best performing models were returned. For each month, the top five (5) returned models were compared against each other based on the value of their mean squared error, the behavior of their learning curve plot, and the accuracy of each predictive model in predicting the values for the observations of the validation dataset. The human intuition in selecting the best performing ANN [among the top 5] is deemed necessary since there's no universal rule on structuring best performing ANN, also since it is a problem highly dependent on the nature of any dataset and the problem of interest.

The downside of performing multiple trainings in a grid search run one after another is that the computer's GPU memory does not clear after each run and there's a possibility that the training information from one model may leak to another training procedure. Therefore, another layer of investigation and fine-tuning were performed on the 12 selected ANN. In this investigation phase, each of the neural network architectures was re-built and re-trained one-by-one on their designated datasets and the resulting learning curves were plotted in Tensorboard¹⁰. One of the advantages of Tensorboard is that metrics, such as loss and accuracy, could be tracked and visualized for each model. This is especially useful in the case of comparing the performance of different neural networks trained on the same dataset. When a model's learning curve was not performing as expected, a fine-tuning and slight modification was performed on the code and the new model's performance was plotted on Tensorboard ready for comparison and, ultimately, selecting the best performing one. In this process, some of the selected original neural network architectures were modified and some architectures remained the same. Table 1 shows the variables in the selected model architectures for each month from the grid search procedure, as well as the final variables after fine-tuning the selected architectures.

Figure 2 takes the month of February as an example to demonstrate how the final ANN [trained on the month's dataset] was structured based on the "final variables" identified in Table 1. An ANN consists of several neurons that are organized into three types of layers known as the input, hidden, and output layers. The input layer is used to introduce the dataset to the

network with no computation performed. The shape of the dataset is defined in this layer by identifying the number of inputs or features of the dataset; the neurons placed and defined in this layer are responsible for passing the dataset's input information to the hidden layers. In the example shown below, the input layer has 512 neurons.

Hidden layers are placed between the input and output layer; their quantity and the number of neurons in each of them can be as many as desired (in the demonstrated ANN, six hidden layers with different numbers of neurons are specified). Defining the hidden layers is done manually and does not follow any specific logic. Typically, different structures are tested and the one that yields the best results for the desired problem of the study is selected. These layers are named "hidden" because the outputs of their computation remain in the network and are not available outside the neural network. The "black box" perception of neural networks is due to the abstract nature of the computations happening in hidden layers. The hidden layers are responsible for performing all sorts of computation on the features entered from the input layer and then transferring the results to the output layer; for instance, the ANN architecture demonstrated herein consists of six hidden layers. Finally, the output layer puts forward the information learnt by the network to the outside of the neural network. The number of neurons in the output layer directly corresponds to the number of outputs or response variables of the dataset. In this example, we are interested in outputting one number which represents the amount of energy consumed for February therefore, we have one neuron placed in the output layer.

ANNs are normally fully connected implying that the neurons of the adjacent layers are fully connected to each other, and each connection has an associated weight (Ciaramella *et al.*, 2015). These neurons operate in correspondence with their associated weight, bias, and activation function in the following procedure: each neuron sums the result of multiplying each input by its associated weight of the input connection and after adding a bias, it applies a function to the result. So, if a neuron is considered to be $X = \sum (\text{weight} * \text{input}) + \text{bias}$, a function is applied to the value of X whose functionality is to decide whether a neuron should be activated or not. Referred to as the *activation function*, its purpose is to add non-linearity to the output of the neurons assuring its learning capabilities. Different variants of activation functions exist including but not limited to Sigmoid, Rectified Linear Unit (Relu), Tanh, and Leaky Rectified Linear Unit (Leaky Relu). Knowing certain characteristics of the problem of interest can help with choosing appropriate activation functions that lead to faster training and more accurate results. For the example shown in Figure 2, Relu demonstrated the best performance as the activation function of this ANN.

⁹Patience value is the number of epochs to wait before early stopping if no progress on the validation set was made.

¹⁰<https://www.tensorflow.org/tensorboard>

Table 1. Original and modified variables in the ANN architecture after retraining process and its resulting MSE

Table of original and modified variables in the ANN ^a training and retraining procedure							
	Variables in the selected model architectures from grid search			Final variables after retraining			Final returned MSE ^b value
	# of layers	1st layer size	Optimizer ^c	# of layers	1st layer size	Optimizer	
January	6	512	Adam	6	512	Nadam	0.009350555156575199
February	6	512	Adam	6	512	Adam	0.010599181247866586
March	8	512	Nadam	8	512	Adam	0.008406761498070537
April	7	512	Adam	7	512	Adam	0.008348045845998105
May	8	512	Nadam	8	512	Nadam	0.011946498677007852
June	7	2,048	Nadam	7	512	Adam	0.008610573696633138
July	6	1,024	Adam	7	512	Nadam	0.008045464576725394
August	7	512	Nadam	7	512	Adam	0.00773768380128509
September	7	512	Nadam	7	512	Adam	0.00706806445608078
October	6	512	Nadam	6	512	Adam	0.007876764878022083
November	7	2,048	Nadam	7	2,048	Nadam	0.009284132136650806
December	6	1,024	Nadam	7	512	Adam	0.008146409050588875

^aArtificial neural network.

^bMean squared error.

^cOptimizers are methods or algorithms used to change the attributes of ANN such as weights and biases.

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 512)	7680
dropout (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 256)	131328
dense_2 (Dense)	(None, 128)	32896
dense_3 (Dense)	(None, 64)	8256
dense_4 (Dense)	(None, 32)	2080
dense_5 (Dense)	(None, 16)	528
dense_6 (Dense)	(None, 8)	136
dense_7 (Dense)	(None, 1)	9
Total params: 182,913		
Trainable params: 182,913		
Non-trainable params: 0		
0.010599181247866586		
[512, 6, 'relu', 'adam', True]		

Fig. 2. The architecture of the ANN that was trained on February dataset.

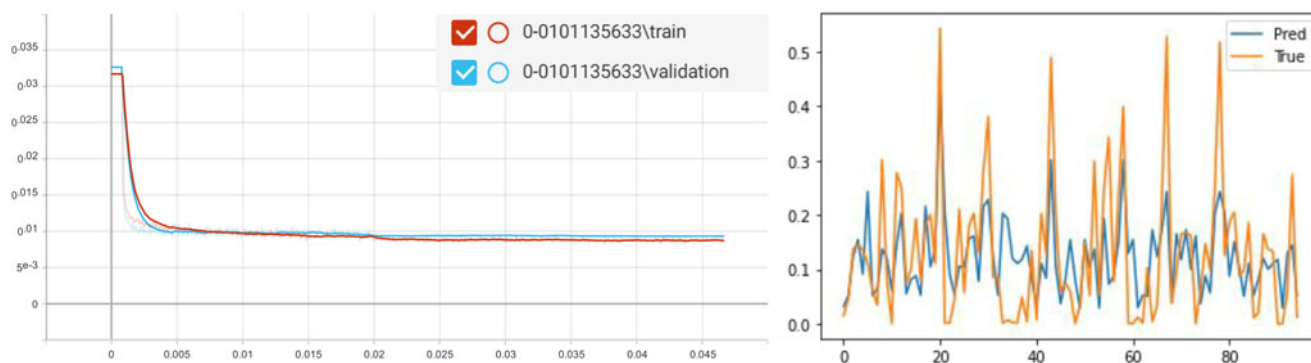


Fig. 3. January's learning curve (left) and prediction accuracy plot (right).

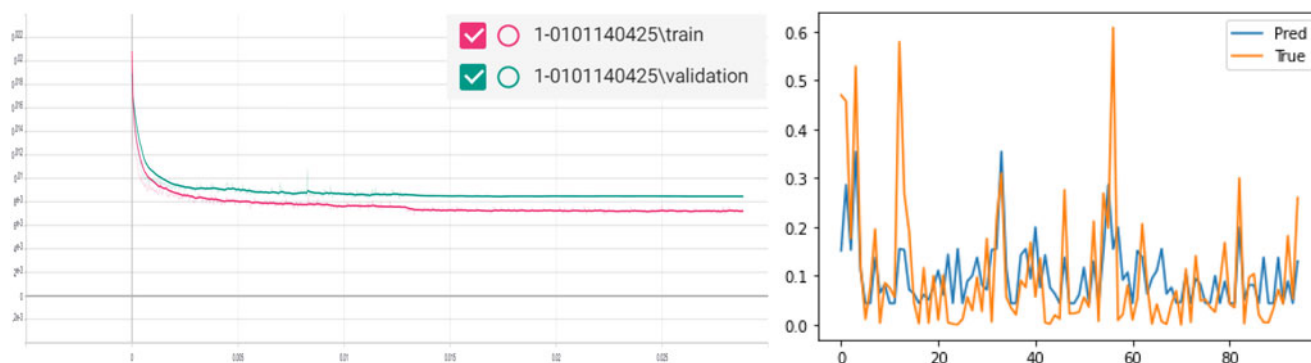


Fig. 4. February's learning curve (left) and prediction accuracy plot (right).

Figures 3–14 show each month's learning curve on the left (x -axis: number of epochs, y -axis: epoch loss), and prediction accuracy plot on the right where predictions are made for the validation dataset and the results are compared against the validation true values¹¹.

After fitting and evaluating the neural network models on each dataset, the finalized models are run on held-back test datasets – which the models have never seen before – to verify the models' performance. The test results of the month of September are shown in Figure 15 as an example. The resulted mean squared error from the testing procedure is reported as 0.046819390610493866. In Figure 15, the orange line shows the true energy consumption values from the test dataset, and the blue line shows the predicted value of energy consumption for each of the observations in the test dataset.

Comparing the model's performance on the test dataset with the model's performance on the training dataset, a performance mismatch is observed; there is a promising performance when evaluating the model on the training dataset and a poor performance when evaluating it on the test dataset. One of the reasons behind model performance mismatch is training on a small and unrepresented dataset. This means that the examples in the training set do not effectively cover the cases observed in the broader domain. Another reason is the stochastic nature of machine learning algorithms resulting from the random initial weights in an

artificial neural network, the shuffling of data, etc. This means that with the same artificial neural network architecture run on the same dataset, different sequences of random numbers are used which in turn return models with different performances. This could potentially be problematic in small datasets, such as the ones used in this research, where each data point or observation counts towards training a model; there might be hard-to-learn observations which sometimes are in training and sometimes are in the validation or test datasets as a result of shuffling. The remedy is often to enrich the dataset to become larger and more representative, a solution that was not possible in the course of this research. Therefore, while being cognizant of this problem, the final trained models are used towards developing the predictive software prototype.

After the training and testing procedure, 12 predictive surrogate models are prepared which can predict each month's value of energy consumption for any unseen and new values of the urban form if the unseen data follow the generalization principle. The generalization principle indicates that trained machine learning models can provide valid predictions for new data as long as they are drawn from the same distribution as the original dataset that was used for training. Honoring this principle and the fact that our trained models are limited to San Diego, it is important to note that the produced surrogate models are unable to estimate valid values of energy consumption for "any" designed urban form scenario; the unseen values of urban form that will be given to the trained models for prediction purposes should fall in the same range of urban form values as the training dataset, representing hypothetical communities as if they were constructed in San Diego.

¹¹For an advanced and high fidelity set of analysis on the training and testing procedures, please visit a previously published paper by the authors of this paper, titled as "A Machine Learning Approach for Mining the Multidimensional Impact of Urban Form on Community Scale Energy Consumption in Cities" (Rahimian *et al.*, 2020).

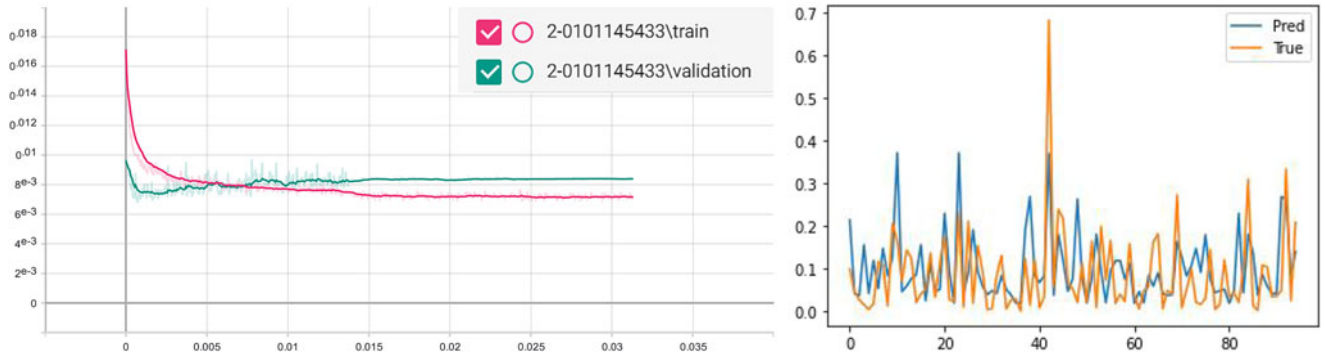


Fig. 5. March's learning curve (left) and prediction accuracy plot (right).

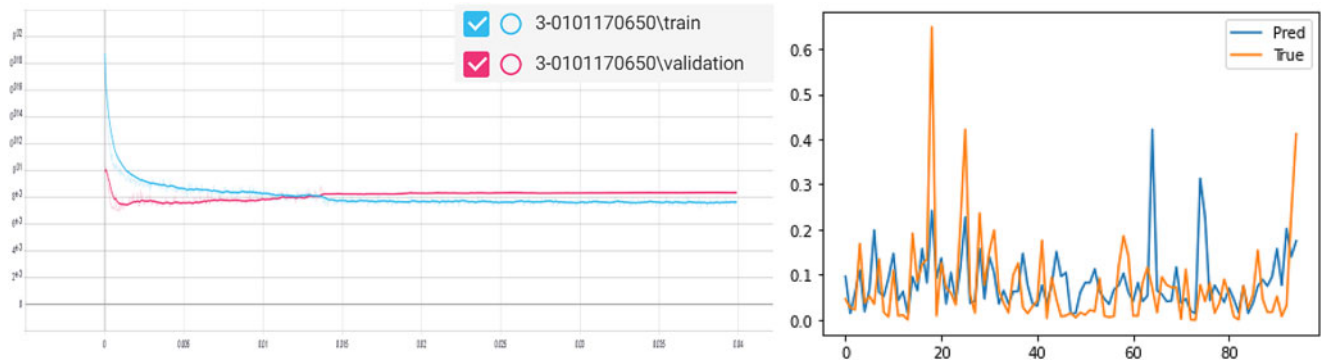


Fig. 6. April's learning curve (left) and prediction accuracy plot (right).

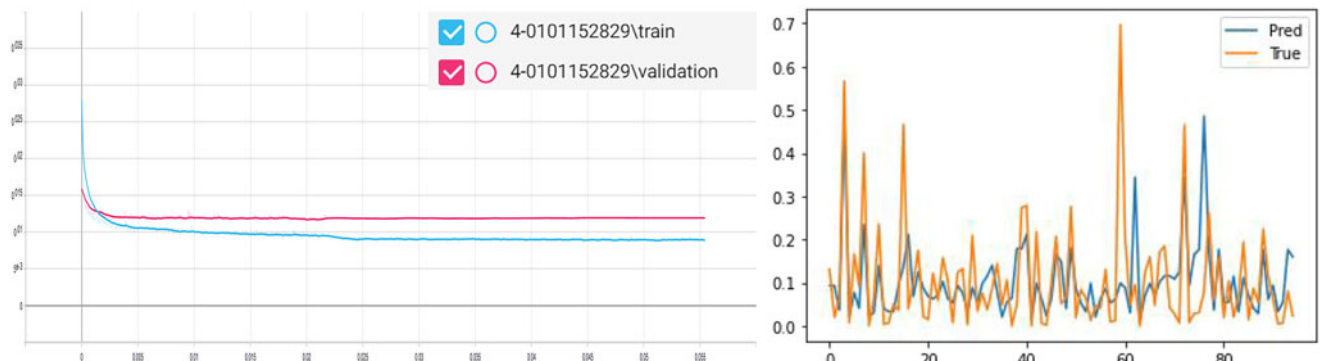


Fig. 7. May's learning curve (left) and prediction accuracy plot (right).

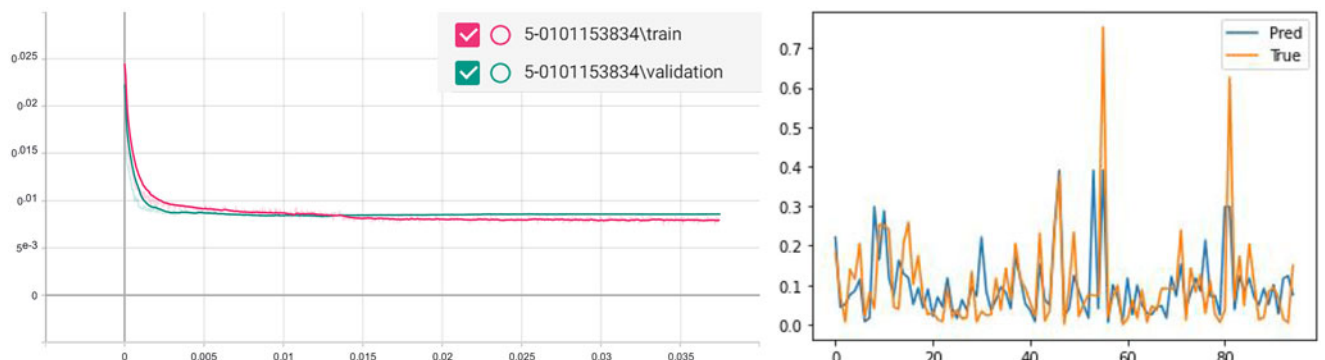


Fig. 8. June's learning curve (left) and prediction accuracy plot (right).

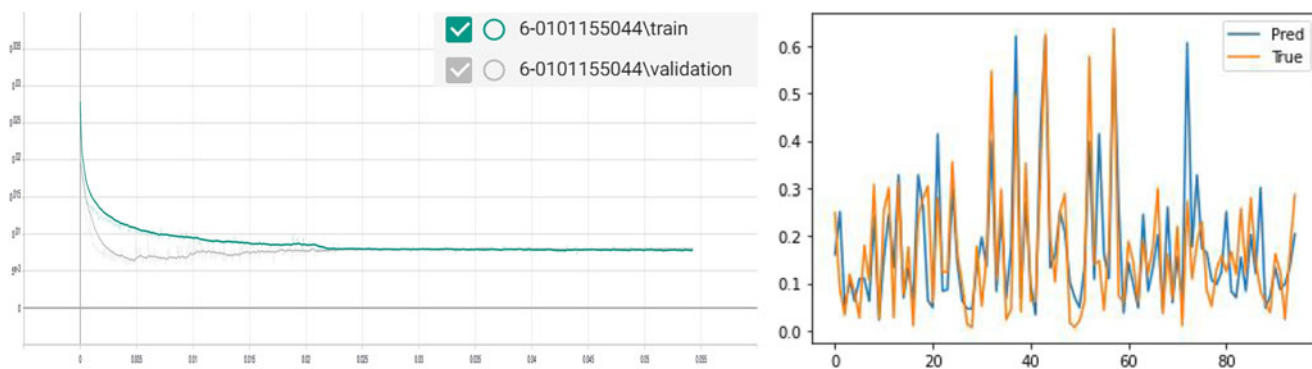


Fig. 9. July's learning curve (left) and prediction accuracy plot (right).

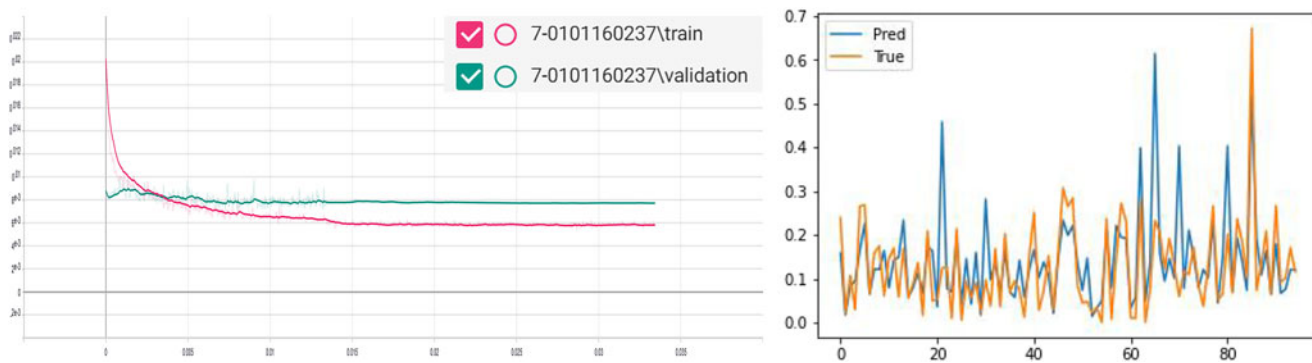


Fig. 10. August's learning curve (left) and prediction accuracy plot (right).

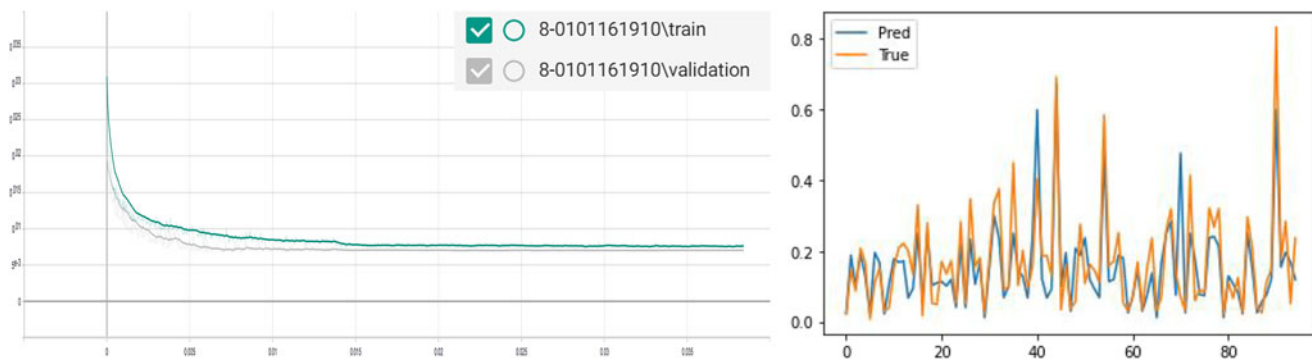


Fig. 11. September's learning curve (left) and prediction accuracy plot (right).

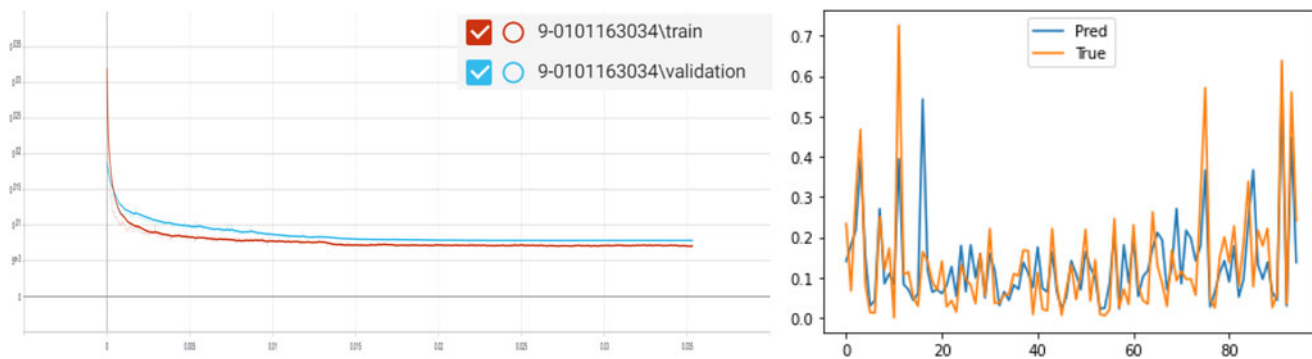


Fig. 12. October's learning curve (left) and prediction accuracy plot (right).

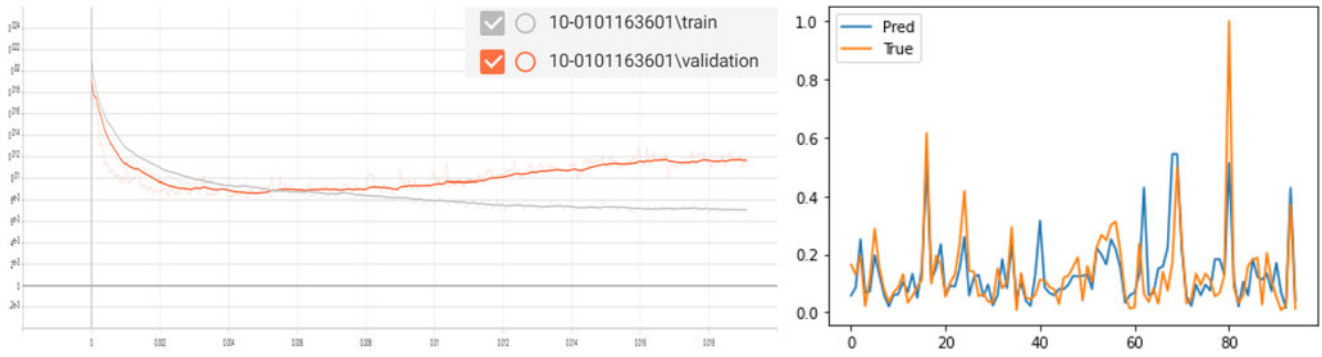


Fig. 13. November's learning curve (left) and prediction accuracy plot (right).

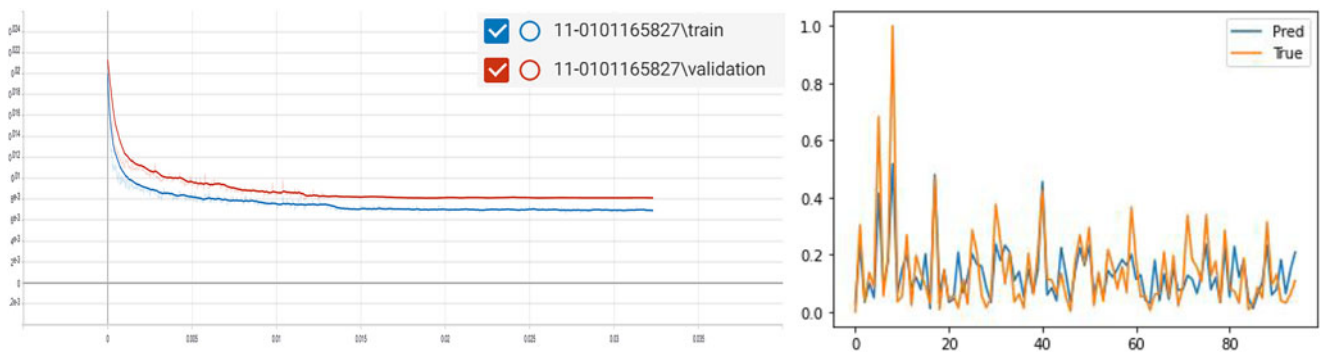


Fig. 14. December's learning curve (left) and prediction accuracy plot (right).

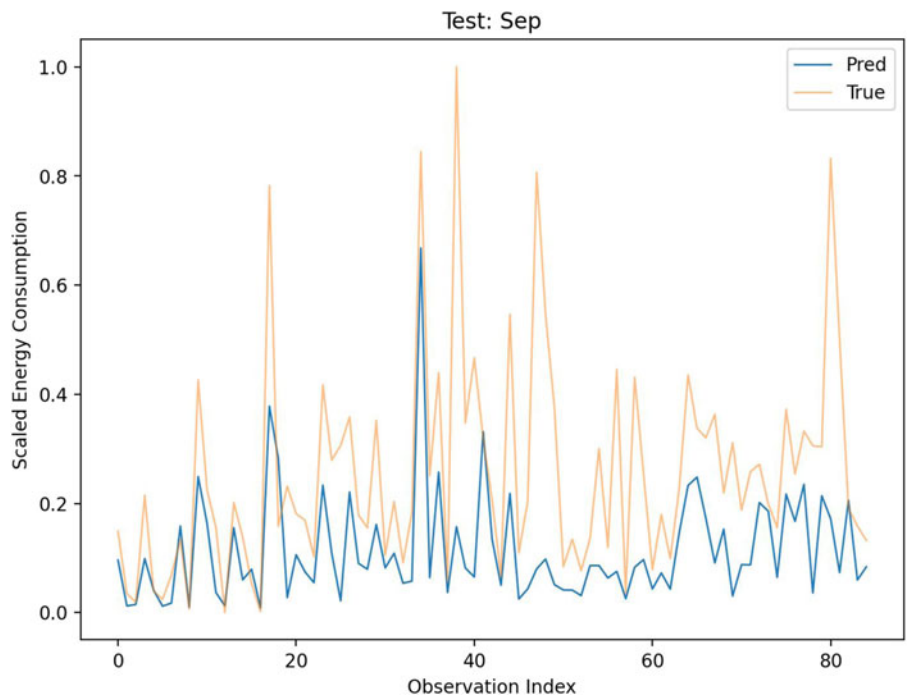


Fig. 15. Plot showing the model's performance on the test dataset for the month of September.

Therefore, to make sure that the software prototype is estimating values of energy consumption for valid design scenarios, a generative urban design algorithm has been developed as part of the software prototype. The generative algorithm generates 3D community design scenarios following San Diego's urban

form, the county's principles of urban planning, and zoning standards. By this, any community designs generated by this algorithm will be similar to the existing urban fabric of San Diego and consequently, their derived measurement of urban form follows the same distribution as the original dataset. Then by adding

the trained surrogate models to the generative algorithm, monthly energy consumption values will be estimated for those generated designs. The next section offers a brief description of the generative algorithm.

A generative algorithm for generating communities following San Diego's urban form

Generative design is a design automation technology based on different artificial intelligence techniques ranging from shape grammars and procedural rule-based systems to genetic algorithms and advanced machine learning-based processes (Gu *et al.*, 2010). In generative design, instead of laboriously designing a single artifact, a design system is programmed which starts with a set of design goals, constraints, and variables – often stemmed from the designer's past experiences and the environment where the design is situated – and then innumerable possible permutations of a solution are explored.

Although overlaps and similarities exist among the different generative design techniques, some appear to be more suitable for certain design and automation tasks (Gu *et al.*, 2010). For example, rule-based systems, like shape grammars, tend to be used when there is strong domain knowledge; Stochastic systems like genetic algorithms are used when there is weaker domain knowledge. To elaborate, methods of generative design can be classified into being explicit or implicit depending on the availability and complexity of data:

- *Explicit or Strong AI methods* – teaching an AI by feeding it human-readable information related to what the programmer/designer thinks the generative system needs to know for generating design options. Rule-based systems are an example of explicit methods.
- *Implicit or Weak AI methods* – where raw data is fed into an AI so the algorithm can analyze and construct its own implicit knowledge about the design such as different machine learning-based



Fig. 16. A portion of San Diego's map from 1979. Source: www.sunnycv.com.



Fig. 17. Two samples of community designs generated by the generative tool.

generative models that is, autoregressive models, variational auto-encoders, and generative adversarial neural networks.

The generative urban design algorithm constructed herein is developed upon explicit methods of generation, namely shape grammars. Shape grammars were used to extract the rules and patterns forming the structure of San Diego's urban typologies leading to its urban form. The extracted shape grammars were then codified into a generative algorithm using Python¹² and Grasshopper¹³, capable of producing various community design scenarios following San Diego's urban form¹⁴. Figure 16 shows a map of San Diego studied for the shape grammars and Figure 17 demonstrates two samples of communities generated by this algorithm which are all spatial representatives of communities in San Diego.

Implementation and software architecture

With the generative algorithm and the twelve-monthly surrogate models established, the next step was to tie these two main components to develop and finalize the experimental software prototype. For each generated community design scenario, this tool is intended to measure its urban form values and estimate the community's monthly estimate of energy consumption accordingly.

For measuring the 19 indices of urban form, several different equations and formulas were added to the generator based on

their measurement metric outlined in Figure 18. By this, the generator can compute the urban form value for energy generated design and output it as a data tree.

The important point to note here is that these 19 indices needed to be sorted and fed into the trained models in the same order that was initially used for training the ANN. A Python communicator was then scripted in Grasshopper which takes the produced 19 values of urban form as a data tree, converts it into a list, and sends it to the server¹⁵. The server is a virtual computer scripted in Python language which uses websockets and Tensorflow¹⁶ (Keras¹⁷) to load all 12 surrogate models (in .h5 format) from the database where they are stored. By loading the models, the server takes any set of 19 numbers and inputs them to the surrogate models as urban form values and uses that to predict and output twelve-monthly values of energy consumption. The urban form values are inputted to the server via the Python communicator. By receiving these values, the server uses the loaded surrogate models to *instantly* estimate its relevant twelve-monthly values of energy consumption and to send the outputs back to the Python communicator through a local network. A visualization script is added to the Python communicator which takes the received predicted values from the Python communicator and visualizes them as a bar chart on the Rhino viewport. When the generator generates a community design scenario, it takes only a fraction of a second to simulate and visualize the design's monthly value of energy consumption. A screenshot of the output of the software prototype is shown in Figure 19.

¹²<https://www.python.org/>

¹³<https://www.rhino3d.com/6/new/grasshopper/>

¹⁴This paper is not going into the details of developing the generative algorithm; more information on the extracted rules and shape grammars can be found in "A Grammar-Based Generative Urban Design Tool Considering Topographic Constraints: The Case for American Urban Planning" (Rahimian *et al.*, 2019).

¹⁵A server is a computer designed to process requests and deliver data to another computer over the internet or a local network.

¹⁶<https://www.tensorflow.org/>

¹⁷<https://keras.io/>

Attribute	Index	Metric
Density	Floor Space Index	$(\text{Gross floor area}) / (\text{Total ground area})$
	Ground Space Index	$(\text{Gross area of building footprint}) / (\text{Total ground area})$
	Open Space Ratio	$(\text{Gross area of unbuilt ground}) / (\text{Total ground area})$
	Layer	$(\text{Average number of floors}) / (\text{Total ground area})$
	Network Density	$(\text{Length of the street network}) / (\text{Total ground area})$
Compactness	Aspect Ratio	$(\text{Average building height}) / (\text{Average street width})$
	Volumetric Compactness	$(\text{Envelope surface area}) / (\text{Building volume})$
	Size Factor	$\sqrt[3]{\text{Building volume}}$
	Form Factor	$\text{Envelope surface area} / \sqrt[3]{\text{Building volume}^2}$
Diversity	Mixed Use Index	$(\text{Gross residential floor area}) / (\text{Gross floor area})$
Green Areas	Green Space Density	$(\text{Gross green space area}) / (\text{Total ground area})$
	Green Area Geometry	$(\text{Gross green space perimeter}) / (\text{Gross green space area})$
Orientation	Community Building Orientation	$(\text{Sum of all buildings}^{\wedge} \text{ orientation}) / (\text{Number of buildings})$
	Street Orientation	$(\text{Sum of all streets}^{\wedge} \text{ orientation}) / (\text{Number of buildings})$
Shading	Urban Horizon Angle (UHA)	$(\text{Average height of the opposite skyline}) / (\text{Canyon width}) = \tan (\text{UHA})$
	Obstruction Sky View (OSV)	$(\text{Average height of the opposite skyline}) / (\text{Canyon width}) = \cos (\text{OSV})$
	Sky View Factor	$\text{if average building height} > 25 \text{ then } \text{SVF} = 1.56 - (0.00572 * H * D)$ $\text{if average building height} < 25 \text{ then } \text{SVF} = 0.9502 + (0.00042 * H) + (0.0198 * D) - (0.0065 * H * D)$ where H is the average height of the community buildings and D is the ground space index
Passivity	Passivity Ratio	$(\text{Net perimeter of all passive zones in community buildings}) / (\text{Net perimeter of all nonpassive zones in community buildings})$
	Plan Depth	$(\text{Net volume of all community buildings}) / (\text{Net area of all exposed walls in community})$

Fig. 18. Energy-relevant indices of urban form along with their measurement metric.

A diagram is provided in Figure 20 which shows the prototypes architecture for clarification. To summarize the diagram:

1. Database: is a folder on the computer storing all trained machine learning models in. h5 format.
2. Back-end is a server responsible for calling and loading the trained machine learning models from the database, receiving and computing requests, and delivering data to other computing programs.
3. Front-end: is the interface the user works with. In this prototype, the front-end is the urban generator in Grasshopper where the user can change certain parameters and have the algorithm generating various different 3D community-scale urban scenarios based on San Diego's principles of urban planning. The front-end is also used to visualize the energy simulation results.
4. Data: are the values being inputted to the server (19 values of urban form) and the data outputted from the server (twelve-monthly values of energy consumption). When the user selects its custom community design through the front end, the generator algorithms automatically extract 19 features of urban form and through a python code send that to the server. The server immediately predicts the monthly values of energy consumption as outputted by the trained models and sends it back to Grasshopper. The front-end instantly visualizes the predicted energy values as a bar chart in the Rhino environment.



Fig. 19. A picture screen of the output of the software. The urban setting that was created by the tool is shown on the right, its values of urban form are shown on the top left, and the predicted values of energy consumption are shown on the bottom left.

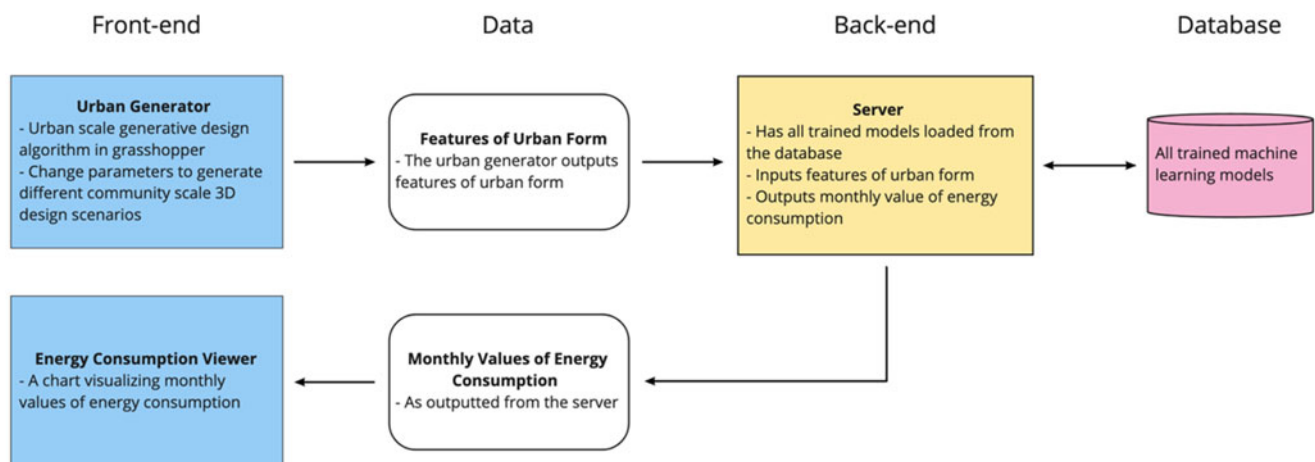


Fig. 20. The prototype's software architecture.

Discussion

One important point drawn from this study is the importance of having urban-scale energy simulation programs that do not rely on complex mathematical models and physics-based computations. In order to encourage architects and urban designers to design energy self-sufficient communities, the first step is to provide them with tools that enable them to do so. This research provides a prototype of a software that runs urban-scale energy simulations in real time which can be used for community designs in the context of San Diego. While this prototype is specific to San Diego, but its development process provides a guideline for developing similar tools as well offering practical avenues for more advanced developments.

This prototype does not provide exact numbers of energy consumption for any designed community but rather offers a “predictive” number. An imprecise simulation of energy performance provided in real time while designing an urban area is more practical and useful for an architect and urban designer than a software that takes hours to run a simulation but provides a more accurate number. This is particularly useful when a project is in its schematic phase or early stages of design; with a real-time tool, the designer would be able to constantly change the design and see the simulation results instantaneously and thus produce design scenarios with improved energy performances. With a time-consuming simulation tool that runs on hardcoded formulas which requires heavy computing processes, it would be difficult to

be used in a practical and efficient way by designers. Firstly, because the designs need to be fairly detailed and elaborated to initiate the simulation process and thus the tool cannot be reasonably used in the schematic phase of a design project. Secondly, since the simulation process takes an extensive amount of time to be processed, it becomes unlikely for the designer to delicately change their designs if their intended energy targets are not met in their initial rounds of iterations. Moreover, for urban-scale energy planning and design, an approximate number is sufficient to guide the designs from a high level towards more energy-conscious solutions. This prototype, and the process of developing it, sheds light for future researchers to develop a similar tool that can run urban-scale energy simulations for other – or any – climatic regions.

Conclusion

As mentioned previously, the motivation behind this research is the practical disengagement of the design sector in the development process of community microgrids given that the importance of this involvement has been marginally highlighted among research and academic communities. Our research shows that one reason driving this disengagement is the illiteracy among architects and urban designers on the impact of urban form and urban geometry on the energy performance of cities at large. This is due to the spatial complexity associated with urban form and the fact that studies to date were not able to capture the multidimensional influence of urban form on community and urban-scale energy performance.

The second reason behind the above-mentioned disengagement is the lack of urban-scale energy modeling and simulation tools that capture the presented complexity. As explained in “A Review of Predictive Software for the Design of Community Microgrids” from the authors of this paper (Rahimian *et al.*, 2018), the quantity outputted from existing urban-scale energy simulation tools are the summed amount of each building’s energy performance in an area, which is not an accurate reflection of the real-world energy performance of cities. Moreover, the hardcoded backends of these tools make running urban-scale energy simulations a tedious task and of expensive computation, practically impossible to use for real-world projects. The software prototype described in this paper provides a preliminary example of a tool used by architects and urban designers for spatially designing energy-conscious microgrid-connected communities well before their construction. This is done by helping the users of this tool to reach the required energy targets for their designs within a much lesser time compared to any similar and commercially existing tool. A tool as such helps architects and urban designers in using passive design solutions for spatially designing and/or retrofitting urban settlements towards communities with improved rates of energy performance. Then when a microgrid system is to be established at the community’s infrastructure level, the number of active systems required for electrification will be minimized, resulting in high-energy performance community microgrids.

By setting the path to take ownership of designing energy self-sufficient community microgrids, this study enables architects and urban planners to carefully and profoundly address the pressing issues related to the local supply and demand of clean energy within their profession. This means that planners could be more and more involved in the technical conversation of developing community microgrids by not only handling the esthetics and

quality of life in urban communities but also to get quantitatively concerned with energy system design and engineering. In this regard, this research has reached its main goal of adding a spatial dimension to the development of community microgrids where architects and urban planners get more involved in the development process of these local energy systems.

Acknowledgements. This work is supported by Penn State Department of Architecture, the Stuckeman Center for Design Computing (SCDC), the Hamer Center for Community Design. Special thanks to Jose Beirão at the Design Computing Group (DCG), University of Lisbon, and to Guido Cervone and Seth Blumsack, The Pennsylvania State University.

References

- Amin M and Wollenberg B** (2005) Toward a smart grid. *IEEE Power and Energy* **3**, 34–41.
- Andreson J and Bausch C** (2006) Climate change and natural disasters: scientific evidence of a possible relation between recent natural disasters and climate change. *Policy Department Economic and Scientific Policy* **2**, 1–30.
- Barnett J** (1989) Redesigning the metropolis – the case for a new approach. *Journal of the American Planning Association* **55**, 131–135.
- Cajot S, Peter M, Bahu J, Guignet F and Koch A** (2017) Obstacles in energy planning at the urban scale. *Sustainable Cities and Society* **30**, 223–236.
- Chaillou S** (2019) *AI + Architecture – Towards a New Approach* (Master’s thesis). Graduate School of Design, Harvard University.
- Chatzidimitriou A and Yannas S** (2015) Microclimate development in open urban spaces: the influence of form and materials. *Energy and Buildings* **108**, 156–174.
- Chen LD, Sakaguchi T and Frolick M** (2000) Data mining methods, applications, and tools. *Information Systems Management* **17**, 67–68.
- Ciamarella A, Staiano A, Cervone G and Alessandrini S** (2015) A Bayesian-based neural network model for solar photovoltaic power forecasting. In *International Workshop on Neural Networks*. Cham: Springer, pp. 169–177.
- Crawley DB, Hand JW, Kummert M and Griffith BT** (2008) Contrasting the capabilities of building energy performance simulation programs. *Building and Environment* **43**, 661–673.
- Davila CC, Reinhart C and Bemis J** (2016) Modeling Boston: a workflow for the generation of complete urban building energy demand models from existing urban geospatial datasets. *Energy* **117**, 237–250.
- Ewing R and Rong F** (2008) The impact of urban form on U.S. residential energy use. *Housing Policy Debate* **19**, 1–30.
- Farhangi H** (2010) The path of the smart grid. *IEEE Power & Energy* **8**, 18–28.
- Fayyad U, Piatetsky-Shapiro G and Smyth P** (1996) From data mining to knowledge discovery in databases. *AI Magazine* **17**, 37.
- Fernandez A and Blumsack F** (2010) *Distributing Electric Energy in Rural America Efficiently and Economically: The Micro-Grid Option*. University Park: The Pennsylvania State University.
- Forrester A, Sobester A and Keane A** (2008) *Engineering Design Via Surrogate Modelling: A Practical Guide*. University of Southampton, UK: Wiley.
- Géron A** (2017) *Hands-on Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. Sebastopol, CA: O’Reilly Media.
- Gorissen D, Couckuyt I, Demeester P, Dhaene T and Crombecq K** (2010) A surrogate modeling and adaptive sampling toolbox for computer based design. *Journal of Machine Learning Research* **11**, 2051–2055.
- Gossop C** (2011) Low carbon cities: an introduction to the special issue. *Cities* **6**, 495–497.
- Grosso M** (1998) Urban form and renewable energy potential. *Renewable Energy* **15**, 331–336.
- Gu N, Singh V and Merrick K** (2010) A framework to integrate generative design techniques for enhancing design automation. *New Frontiers: Proceedings of the 15th International Conference on Computer-Aided Architectural Design in Asia (CAADRIA)*. Hong Kong: Association of Computer-Aided Architectural Design Research in Asia (CAADRIA), pp. 127–136.

- Hu R, Huang Z, Tang Y, van Kaick O, Zhang H and Huang H** (2020) Graph2plan: learning floorplan generation from layout graphs. *arXiv pre-print arXiv:2004.13204*. **39**, 118:1–118:14.
- Huang W and Zheng H** (2018) Architectural Drawings Recognition and Generation through Machine Learning. ACADIA, pp. 156–165.
- Magoules F and Zhao H-X** (2016) *Data Mining and Machine Learning in Building Energy Analysis: Towards High Performance Computing*. Hoboken, NJ: John Wiley & Sons.
- Owens S** (1986) *Energy, Planning, and Urban Form*. London: Pion Limited.
- Rahimian M, Iulo LD and Duarte JP** (2018) A review of predictive software for the design of community microgrids. *Journal of Engineering* **2018**, Article ID 5350981, 13 pages. <https://doi.org/10.1155/2018/5350981>.
- Rahimian M, Beirão JN, Duarte JP and Domenica Iulo L** (2019) A grammar-based generative urban design tool considering topographic constraints: The Case for American Urban Planning. *Proceedings of the 37th eCAADe and 23rd SIGraDi Conference – Volume 3: Architecture in the Age of the 4th Industrial Revolution*. Porto, Portugal: University of Porto, pp. 267–276.
- Rahimian M, Cervone G, Duarte JP and Iulo LD** (2020) A machine learning approach for mining the multidimensional impact of urban form on community scale energy consumption in cities. *Design Computing and Cognition (DCC)*. Atlanta, GA: Springer, pp. 627–648.
- Reinhert C, Dogan T, Jakubiec J, Rakha T and Sang A** (2013) UMI – an urban simulation environment for building energy use, daylighting and walkability. *13th Conference of International Building Performance Simulation Association*. Chambéry, France, pp. 476–483.
- Samuel A** (1959) Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development* **3**, 210–229.
- Santamouris M, Papanikolaou N, Livada I and Koronakis I** (2001) On the impact of urban climate on the energy consumption of buildings. *Solar Energy* **70**, 201–216.
- Seyedzadeh S, Pour Rahimian F, Glesk I and Roper M** (2018) Machine learning for estimation of building energy consumption and performance: a review. *Visualization in Engineering* **6**, 1–20.
- Siderius H-P** (2004) The end of energy efficiency improvements = the start of energy savings?!. *2004 ACEEE Summer Study on Energy Efficiency in Buildings*. Pacific Grove, CA: American Council for an Energy-Efficient Economy, pp. 11.165–11.176.
- Silva MC, Horta IM, Leal V and Oliveira V** (2017) A spatially-explicit methodological framework based on neural networks to assess the effect of urban form on energy demand. *Applied Energy* **202**, 386–398.
- Simpson T, Toropov V, Balabanov V and Viana F** (2008) Design and analysis of computer experiments in multidisciplinary design optimization: a review of how far we have come or not. *Proceedings of the 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. MAO, Victoria, Canada.
- Steadman P** (1977) Energy and patterns of land use. *Journal of Architectural Education* **30**, 62–67.
- Stemers K** (2003) Energy and the city: density, buildings and transport. *Energy and Buildings* **35**, 3–14.
- Tamke M, Nicholas P and Zwierzycki M** (2018) Machine learning for architectural design: practices and infrastructures. *International Journal of Architectural Computing* **16**, 123–143.
- Tardioli G, Kerrigan R, Oates M, O'Donnell J and Finn D** (2015) Data driven approaches for prediction of building energy consumption at urban level. *Energy Procedia* **78**, 3378–3383.
- Ton DT and Smith MA** (2012) The US department of energy's microgrid initiative. *The Electricity Journal* **25**, 84–94.
- Tsui KL, Chen V, Jiang W and Aslandogan Y** (2006) Data mining methods and applications. In Pham H (ed.), *Springer Handbook of Engineering Statistics*. London: Springer, pp. 651–669.
- Wouters C** (2015) Towards a regulatory framework for microgrids – the Singapore experience. *Sustainable Cities and Societies* **15**, 22–32.
- Mina (Vina) Rahimian** is earning her PhD in Architecture at the Pennsylvania State University (Penn State) that will be conferred in Spring 2022. Mina also did her MS in Architecture at Penn State and her bachelor's in architectural engineering at University of Tehran. Her research interest lies in technological and data-driven responses for solving problems related to the built environment specially at an urban scale. Mina is currently a senior product manager at Outer Labs and an adjunct lecturer at Penn State.
- José P. Duarte** (Lic Arch UT Lisbon 1987, SMArchS 1993, and PhD MIT 2001) is the Stuckeman Chair in Design Innovation and director of the Stuckeman Center for Design Computing at Penn State, where he is the Professor of Architecture and Landscape Architecture, and Affiliate Professor of Architectural Engineering and Engineering Design. Dr. Duarte was dean of the Lisbon School of Architecture and president of eCAADe. He was co-founder of the Penn State Additive Construction Laboratory (AddCon Lab) and his research interests are in the use of computation to support context-sensitive design at different scales.
- Lisa D. Iulo** is an Associate Professor of Architecture and Director of the Hamer Center for Community Design at Penn State. Her work focuses on research and design related to residential green building and affordable housing, energy efficiency, and strategies for the implementation of renewable energy at the building and community scale. With support from Penn State, NSF, and DOE, she has been collaborating with colleagues to better understand the building/community relationships and opportunities where research, data, and improved decision-making can inform the design of resilient buildings and communities. Lisa has been a member of the architecture faculty at the Penn State since 2003.