

Predicting Sequential Design Decisions Using the Function-Behavior-Structure Design Process Model and Recurrent Neural Networks

Molla Hafizur Rahman

Department of Mechanical Engineering,
University of Arkansas,
Fayetteville, AR 72701
e-mail: mbrahim@email.uark.edu

Charles Xie

Institute for Future Intelligence,
Natick, MA 01760
e-mail: charles@intofuture.org

Zhenghui Sha¹

Mem. ASME
Department of Mechanical Engineering,
University of Arkansas,
Fayetteville, AR 72701
e-mail: zsha@email.uark.edu

In engineering systems design, designers iteratively go back and forth between different design stages to explore the design space and search for the best design solution that satisfies all design constraints. For complex design problems, human has shown surprising capability in effectively reducing the dimensionality of design space and quickly converging it to a reasonable range for algorithms to step in and continue the search process. Therefore, modeling how human designers make decisions in such a sequential design process can help discover beneficial design patterns, strategies, and heuristics, which are essential to the development of new algorithms embedded with human intelligence to augment the computational design. In this paper, we develop a deep learning-based approach to model and predict designers' sequential decisions in the systems design context. The core of this approach is an integration of the function-behavior-structure (FBS) model for design process characterization and the long short-term memory unit (LSTM) model for deep learning. This approach is demonstrated in two case studies on solar energy system design, and its prediction accuracy is evaluated benchmarking on several commonly used models for sequential design decisions, such as the Markov Chain model, the Hidden Markov Chain model, and the random sequence generation model. The results indicate that the proposed approach outperforms the other traditional models. This implies that during a system design task, designers are very likely to rely on both short-term and long-term memory of past design decisions in guiding their future decision-making in the design process. Our approach can support human-computer interactions in design and is general to be applied in other design contexts as long as the sequential data of design actions are available. [DOI: 10.1115/1.4049971]

Keywords: computer-aided design, data-driven design, decision theory, design process, deep learning, systems design

1 Background Introduction and Research Overview

Design involves an iteratively searching process of design space for desired solutions. In such a search process, designers make sequential decisions that involve the selection of design actions and determination of design parameters so that they can get the best output. Since the search process always accompanies uncertainties, the tradeoff strategies and the decisions on when and where to explore and exploit design space are essential to the quality of design outcomes and the resources needed to achieve the objectives. For example, in our previous study on sequential design behaviors [1], we integrated the Markov chain and unsupervised clustering methods to explore designers' sequential behaviors. We observed that designers follow specific design patterns, such as high frequent design synthesis-related operations, in systems designs that often have many coupling design variables and exhibit significant design uncertainties.

These sequential decision-making strategies are often the key features that differentiate expert designers and novice. Therefore, these strategies compose the essence of human intelligence in

design. This is also the reason why sometimes human designers are effective in reducing the dimensionality of solution space of complex systems design and quickly converge it to a reasonable range for the search. For example, human designers show very different strategies compared with computational algorithms [2], and they are more effective than algorithms searching for promising design candidates in certain situations, such as in early-stage systems design [3].

Modeling and machine learning of human sequential design decisions would significantly impact the engineering design process and design automation in three aspects. First, successful modeling of sequential design decisions can help discover quantifiable design patterns that are useful to improve the existing computational design by encoding human intelligence. Second, the discovery of beneficial design patterns from expert designers can be used to train novice designers. Third, computational modeling of sequential design decisions can help build artificial design agent, which can be used in computer-aided design (CAD) systems to work collaboratively with human designers by reducing unnecessary design iterations and reinforce useful iterations.

These benefits are significant to socio-technical systems engineering and design. This is because human-system interaction plays a vital role in determining whether desired design solutions can be successfully achieved [4] in the open design platform or distributed innovation. For example, in the study of participants' behaviors on GrabCAD, a community-based design innovation

¹Corresponding author.

Contributed by the Design Automation Committee of ASME for publication in the JOURNAL OF MECHANICAL DESIGN. Manuscript received June 23, 2020; final manuscript received January 19, 2021; published online March 18, 2021. Assoc. Editor: Michel-Alexandre Cardin.

system, Sha et al. [5] developed a network-based approach and used online design challenge data to identify the important system factors (e.g., incentive structures and prize amount) that can significantly influence participants (designers') behaviors, which in turn, impact how successfully the GrabCAD system can solicit design solutions through design contests. Similarly, with the establishment of the design collaboration network of OpenIDEO—a design and innovation online platform that uses a human-centered, collaborative approach to solving complex issues—Fuge et al. [6] found that the core-periphery structure in OpenIDEO's collaborative network is at risk of decreasing its design exploration ability. The authors proposed possible interventions that can prevent this issue, for example, by encouraging core members to collaborate with periphery nodes, and increasing the diversity of the user population. In other socio-technical systems, such as autonomous vehicle and human-robot manufacturing, the ability to predict operators' next move is even more critical because it directly affects the system performance and human safety.

However, modeling and predicting sequential human decisions is challenging. First, human decisions are a result of a mental process that is hidden, implicit, and sometimes tacit [7] because they are difficult to be transferred in an explicit way like writing or verbalizing. Second, sequential design data have unique features that are different from the sequential data in other fields, such as in natural language or human daily routine activities where machine learning approaches can yield satisfactory performance. In design, designers perform conceptual design at the beginning stage and involve design actions that would be very different from the ones in the embodiment design stage. This indicates the design actions taken in various phases of design would be very different, while in the same phase, a particular set of actions could be more preferred. Third, design decisions are intricate, particularly in systems design scenarios where the design process often spans over a long period, and designers' decisions involve multiple interdependent variables; thus, their decisions are highly correlated in time sequence. So, it is very challenging for traditional sequential learning models such as the Markov model (MM), autoregressive integrated moving average (ARIMA), etc., to discover prominent design patterns for prediction. To address these challenges, the objective of this study is to computationally model and predict human designers' sequential decisions in support of an in-depth understanding of designers' thinking in the systems design context.

Artificial neural network (ANN), which mimics the human brain, has been proved to be capable of machine learning sequential behavioral patterns in various fields such as natural language processing [8], healthcare [9], image recognition [10], finance [11], etc., and recently in the field of engineering design (see Sec. 2 for detail). Yet, there is still a gap of using deep learning-based approaches in understanding designers' design processes and sequential decision-making, which requires integration with design theory, and particularly design process models. Therefore, the questions we are interested in this study are: how a design process model can be used in support of modeling and understand designer's sequential decision-making? And what is the performance of the neural network-based models in predicting sequential design decisions? To answer these questions, we develop a deep learning-based approach integrating recurrent neural network (RNN) and function-behavior-structure (FBS).

In this paper, we focus on the systems design problems that involve parametric and configuration design decisions. In real design scenarios, from product design to production pipeline, configuration and parametric design are crucial where designers make decisions on which design components to use and what design parameter values to choose for the desired design outcome that satisfies the given requirements. These decisions can be recorded continuously in real time as a sequence of actions taken. The scope and the design context of this research are put under the design decisions made in a CAD process for the ease of data collection, but our approach is generally applicable in any design situations as long as designers' sequential actions can be collected.

The contributions of this work are twofold: (1) A new approach that integrates the FBS-based design process model and RNN to model and predict human sequential design decisions in systems designs. (2) The new knowledge about the capability of deep learning models in predicting sequential design decisions and supporting design thinking research, benchmarking conventional models. The remaining paper is organized as follows. In Sec. 2, we discuss the state-of-the-art design research on sequential design decision-making and deep learning of sequential design data. In Sec. 3, we present our research approach and briefly introduce the technical backgrounds on different types of ANN models and the FBS based design process model. In Sec. 4, two case studies are presented to demonstrate the approach in predicting designers' sequential decisions in a solar energy system design context. The details of the design experiment and data collection are provided in this section too. In Sec. 5, the validation based on a comparative study of different models for sequential design decisions is performed; the results are presented and discussed. Finally, we conclude the paper with closing insights and discuss our future work in Sec. 6.

2 Literature Review of the Research on Sequential Design Decision

Several studies have been done in the engineering design field to explore the sequential patterns, optimize the sequence of design tasks, and finding heuristics from sequence learning. Particularly, a large number of studies have been conducted based on the Markov chain model. For example, in order to compare designers' sequential design behaviors in three different domains, including architecture, software design, and mechanical design, the function-behavior-structure ontology and the first-order Markov chain [12] were adopted. The second-order Markov chain model was also used to explore the effect of previous experience and design knowledge on design sequence [13]. In order to study designers' sequential learning strategies, McComb et al. [14] used a Markov chain model in a truss design problem. Their results indicate that the first-order Markov chain better represents designers' action sequences. In a later study, they used the hidden Markov model (HMM) to study the patterns of sequential design state in the same design problem. They found four hidden states in the configuration design and observed that designers used the first two states to topology operation, the third state to spatial, and the fourth state to parameter operation. The trained HMM model was then utilized to compare the design processes of the high-performing group and low-performing group [15].

To computationally model designers' sequential search process, there have been studies based on Bayesian optimization (BO) framework. For example, to mimic the human searching strategy, Sexton and Ren [3] developed a searching process using the BO algorithm, which can replace human solvers from a design process. Sha et al. [16] also integrated the Weiner process BO with game theory to study designers' sequential decisions in a one-on-one competition for monetary reward. Some studies have also used the Gaussian process-based model [17] and descriptive models based on expected utility maximization [18] to understand human design strategies in the sequential information acquisition decision-making (SIADM) scenario. In order to quantify the impact of designers' domain knowledge and problem framing, Shergadwala et al. [19] developed a SIADM framework incorporating expected improvement maximization and optimal one-step look-ahead strategy. The framework is applied to a motor track design problem and found that problem framing impacts designers' knowledge as well as their performance. Later, the framework is extended as a Strategic-SIADM model to understand the influence of competitors' past performance on individuals' design behavior and outcomes [20].

Prior studies on sequential design processes have also been focused on project task level in support of product development

and project management. For example, the design structure matrix [21] has been used to study task sequencing for identifying the sequence that minimizes expected project completion time. Some other work has been grounded in theoretical processes. For example, Miller et al. [22] use multi-objective formulations to study the design process sequentially advancing through to smaller sets of alternatives using models of increasing fidelity. In addition, optimization approaches, such as the expected value of perfect information [23], genetic algorithm [24], and optimal learning [25], have been utilized in studying optimal design sequences. However, these studies are fundamentally different from the presented work in that they formulate a design problem and cast it into a sequential decision process to be optimized with normative models. In this study, however, we focus on the sequential decision-making of human designers. It is about the actual actions that designers sequentially take. By modeling and analyzing such a design sequence at a fine-grained resolution, it is expected that insights and new knowledge regarding the designers' thought process can be obtained.

In recent years, deep learning techniques have shown their promise in the design field to solve different design problems, including design optimization, design ideation, and design behavioral modeling. For example, Raina et al. [26] developed a two-step deep learning framework. A convolutional neural network (CNN) based auto-encoder is used in the framework to map the images of design to a low-dimensional embedding to generate design without specific design operation (i.e., adding any particular design component). In the second step, the derived embedding and a rule-based image processing inference algorithm are used to output the operation, construct the structure, and iteratively improve the design. The resulting design is found to have a better factor of safety and strength-to-weight ratio over human designs. Oh et al. [27] developed a framework where topology optimization and boundary equilibrium generative adversarial network (BEGAN) is used iteratively to generate new designs. The proposed method is applied to a case study on a 2D car wheel design. Stump et al. [28] developed a method for optimizing the structure and attributes of sailboat design. By embedding spatial grammar in character recurrent neural network (char-RNN), the structure of the sailboat is optimized in a physics-based game engine, Unity3D. In a similar case study, reinforcement learning (RL) is used to minimize the time of the sailboat's travel path. Table 1 shows a summary of these relevant studies.

Although deep learning methods serve as the core of the research approaches in several studies in the design field, the research objectives are fundamentally different from the presented study. Current deep learning-based methods aim at improving or optimizing a design output or generating new designs by training a neural network that learns from existing design artifact data. For example, Raina et al. [26] uses 2D images of the designs as their input and uses the CNN model to learn truss design structures that can yield high design performance. There are similarities between Raina et al.'s work and this study in the ideas of using deep learning techniques to study human sequential design behaviors. But the research goal and the deep neural networks adopted by Raina et al. are different from the current work. For example, while

the other work's ultimate goal is to generate and optimize new designs, this paper aims at understanding designer's thinking in an engineering system design process and learning the beneficial design sequences for an in-depth understanding of their thought processes. Because of this motivation, the integration of a higher level of abstraction of the design process (often known as *design ontology*) is needed to transform the design action space to the design thinking space.

3 Research Approach and Technical Background

In this section, we first introduce our research approach. Then, we present the technical background regarding the deep learning models and the FBS design process model adopted in our approach.

3.1 The Research Approach. The approach starts with the raw data collection of designers' sequential design decisions from different sources such as the action logger of CAD software, interviews of designers, design documents, etc. The raw data contains the details of human design behaviors (i.e., design actions) as well as design artifact's information, such as values of design parameters, simulation results, etc. In this study, we only extract the actions which are only design-related; for example, in a CAD environment, these actions could be adding a new component or editing that component. Designers act based on the given design requirements and constraints; thus, those design actions can reflect designers' thinking and strategies in searching the design space. Next, we apply a design process model to convert the design actions into design process data. The design process model consists of a series of design stages that characterize a design process. This treatment transforms the action space into a design process space. This treatment helps better interpret and understand designers' design thinking and reduce the dimensionality of the sequential action data (see Sec. 4.2 for details). Then, we use the sequential design process data to train deep learning models and predict the next immediate design action category (i.e., the design stage defined by a design process model) based on the trained models. In this study, we use ANN models, particularly the feed-forward neural network (FNN) and RNN models, to implement the deep learning approach as these two models can properly handle sequential data [29]. Finally, we evaluate the predictive performance of these models and compare them with those commonly used models using different metrics, such as testing accuracy, precision, recall, F1 score, and area under the receiver operating characteristics (AUROC) curve, at both aggregated level and design process stage level (see Sec. 5 for details). Figure 1 depicts a schematic diagram of the overall approach used in this study.

3.2 Feed-Forward Neural Network and Recurrent Neural Network. In an FNN architecture, the information follows one direction from input to output with no back loops. In addition to the input layer and output layer, FNN may have single or multiple hidden layers. When FNN has only one layer between input and

Table 1 The summary of the relevant literature on deep learning in engineering design

References	Data type	Used deep learning model	Research objective	Design context	Number of available design operations
Raina et al. [26]	Images of sequence of design artifacts	Convolutional neural network	Generative design and optimization	Truss design	9 design operations
Oh et al. [27]	Images of sequence of design artifacts	Generative adversarial network	Topology optimization	Car wheel design	N/A
Stump et al. [28]	3D CAD geometry	Char-recurrent neural network	Structure and attribute optimization	Sailboat	N/A
McComb et al. [15]	Text data sequence of design actions	Hidden Markov model	Identify beneficial design heuristics	Truss design and cooling system design	9 operations for truss design and 7 for cooling system design

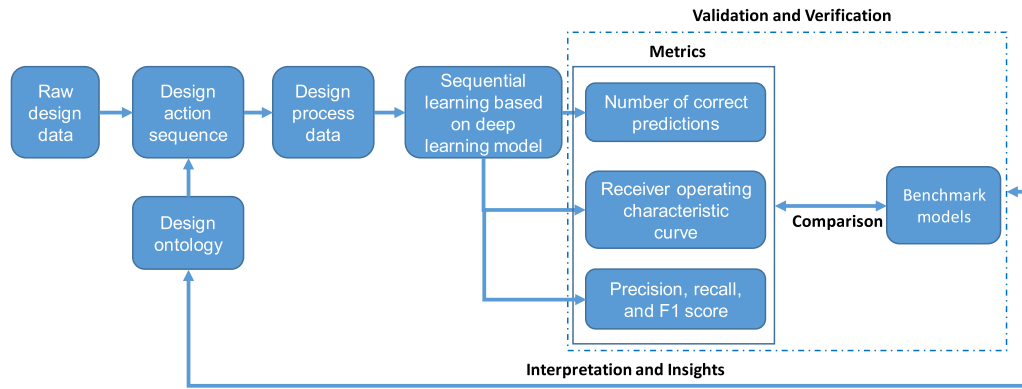


Fig. 1 The overall research approach

output, it is known as the single-layer perceptron. An FNN with more than one hidden layer, including the output layer, is called a multilayer perceptron. An FNN with a single hidden layer between the input layer and the output layer is often sufficient to be universal function approximation [30]. However, deep neural networks with additional hidden layers outperform this shallow model. A standard depth of deep neural network (i.e., the number of hidden layers) may vary from two or three to even one thousand [10]. The structure of the RNN is similar to that of an FNN. The only distinction is that there is no restriction on back loops. So, the information not only passes in one direction forward but also does it flow backward, called *recurrence*. This feature allows RNN to create a hidden state which carries information from the previous time-steps to its current step. Although RNN can be used for capturing long-term dependencies, *simple recurrent units* are not effective in this task due to vanishing gradient problem [31]. In order to solve this problem, Hocreiter and Schmidhuber [32] proposed the long short-term memory units (LSTMs)—a special type of mechanism where information flow is controlled by three different *gates*, namely, input gate, forget gate, and output gate. LSTM is more widely used than simple RNN in many domains for its capability of modeling long-term dependencies. To study to what extent the past decisions of designers' can influence their future decision-making, we adopt LSTM as a representative RNN model in our study.

3.3 The Function-Behavior-Structure Design Process Model. The FBS model [33], a domain-independent design process model, consists of three ontological design variables: Function (F), Behavior (B), and Structure (S). Function describes the purpose of the design and establishes the connection between design goals and measurable effect. Behavior is defined as a design attribute that can be derived from the design structure. Structure (S) is defined as the design component and its interconnected relationship. During the design task, designers establish interconnections among these three variables. The first basic interconnection is constructed by transforming the function into behavior and behavior into the structure, i.e., $F \rightarrow B$ (#1), and $B \rightarrow S$ (#2), as shown in Fig. 2. Here behavior is interpreted as the expected performance in order to achieve the function. However, once the structure is generated, the expected performance may not be achieved. Therefore, the performance from the structure needs to be compared with the expected performance. For this reason, in the FBS design model, the behavior is distinguished into two separate classes of behavior: expected behavior (Be) and behavior derived from the structure (Bs). With these additional variables, the transformations are extended as follows: $F \rightarrow Be$ (#1), $Be \rightarrow S$ (#2), $S \rightarrow Bs$ (#3), and $Be \rightarrow Bs$ (#4), as shown in Fig. 2.

Often designers start their design from the initial requirement and finish the design task by reporting the description of the design. Therefore, two additional design variables: requirements (R) and

description (D) are added to the design process model. The FBS design process model regards requirements (R) as a function (F) generator and defines description (D) as the representation of a design task. The additional transformations are $S \rightarrow D$ (#5) and $R \rightarrow F$ (#1).

During the design task, designers iteratively and incrementally improve the design. Also, designers implement new ideas by removing or changing the existing structures or functions in order to improve the behavior of the structure. Thus, additional three transformations are included in the FBS model: $S \rightarrow S'$ (#6), $S \rightarrow Be'$ (#7), and $S \rightarrow F$ (#8). With all of these transformations, a total of eight design processes are obtained, as summarized in Table 2, along with the interpretations. This FBS design ontology provides us with the rationale for the transformation of design action data to design process data in support of the study of design thinking.

4 Predicting Sequential Design Processes in Solar Energy Systems Design With Two Case Studies

In this section, we present two case studies on solar energy systems design and implement the proposed approach to predicting designers' sequential design decisions. First, we introduce the design experiments conducted for data collection. Next, we present the collected data and introduce the methods for processing it.

4.1 The Design Context. In order to collect sequential design behavioral data, we conducted a series of design challenges on real-world engineering design problems. The challenges were held at the University of Arkansas. Both undergraduate and graduate students from engineering disciplines participated in these challenges. In this study, we mainly adopt the data collected from two design challenges. In the first challenge, the students were asked to design a solarized home in Texas with a budget of \$200,000 (see Fig. 3).

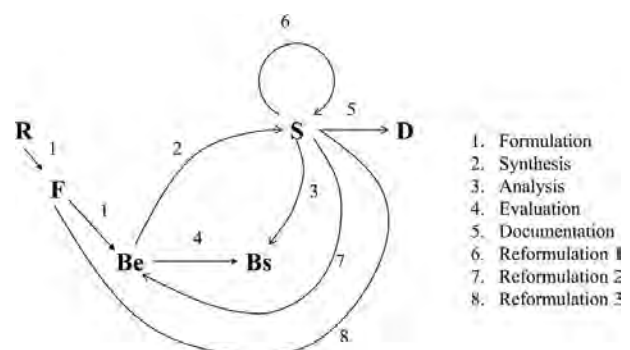


Fig. 2 The FBS design ontology (adapted from Ref. [33])

Table 2 Transformation of ontological design variables and the rationale of the FBS-based design processes

Transformation	Design process	Definition and interpretation
F → Be & R → F	Formulation	Generate when requirement is transformed into function and function is transformed into behavior
Be → S	Analysis	Obtain behavior from generated structure
S → Bs	Synthesis	Generate and tune structure based on the expected behavior
Be → Bs	Evaluation	Comparison of the expected behavior and actual behavior
S → D	Documentation	Generate design description based on structure
S → S'	Reformulation 1	Regenerate and modify one structure to another structure
S → Be'	Reformulation 2	Regenerate or modifies structure based on the expected behavior
S → Be'	Reformulation 3	Regenerate and modifies structure based on the formulation

The second challenge was to build a solarized parking lot at the University of Arkansas. The budget for this challenge was \$1.5M.

The design objective of both challenges is to maximize the annual net energy (ANE) with the given budget. The design requirements and constraints are provided to the participants so they can start the design with a focus on configuration and parametric design. In this study, the design variables are mainly related to the components that have a direct impact on the design objective. We limit the number of design variables and design constraints so that the experiment remains in a controlled manner. By doing so, we are able to compare the models in different settings of design complexity. Table 3 shows the requirements and constraints of both design challenges. These two design problems reflect different design complexity in terms of design variables and the couplings between these variables. For example, the energy plus home design problem has more design variables than the parking lot design problem. Therefore, the two design problems bring generality for a comparison of the proposed approach.

Students' designs were conducted within a CAD environment, called Energy3D. Energy3D is a full-fledged CAD software specially built for solar systems design [34]. It has several unique features, such as interactive visualization, high-fidelity simulation, and built-in financial evaluation [35]. These features can help designers effectively explore and exploit the design space. Moreover, Energy3D has a nonintrusive data action logger. That means designers are not aware of the data collection process, and this helps reduce participants' cognitive burden that could be introduced in experimental settings. As a result, the data that reflect designers thinking and decision-making can be less biased. Energy3D sorts and logs every performed action in JSON format.

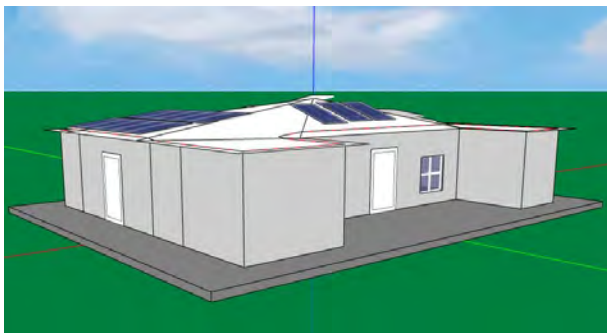


Fig. 3 An example of solarized home designed by one of the participants

Table 3 The design requirements and constraints of the two design problems

Components		Requirements
Solarized home	Story	1
	Number of windows	>4
	Size of windows	>1.44 m ²
	Number of doors	≥1
	Size of doors (Width × height)	>1.2 m × 2m
	Height of wall	>2.5 m
Solarized parking lot	Distance between ridge to panel	>0
	Base height	≥3.5
	Tilt angle	≤20
	Solar panel rack	Shall not produce any hindrance to the pedestrian zone and drive ways
	The pole of the rack	Shall be placed along with the parking lot line marker

This high-resolution data provide us with a large amount of data that is essential to implementing deep learning models.

4.2 Data Collection and Preprocessing. Energy3D collects the continuous flow of design action data, which includes time-steps, design actions, design parameter values, and simulation results. In the solarized home design problem, a total of 52 engineering students participated in this design challenge. Among them, 29 students are undergraduate students, and 23 are graduate students. 40 students are from Mechanical Engineering. On average, the design action log records 1500 lines and 220 intermediate files per student. In the parking lot design problem, a total of 41 students participated in which 35 students are from undergraduate, and five students are from graduate students, all in the major of Mechanical Engineering. The design action log records, on average, 1300 lines of data per student. An example of one logged design action is presented below:

```
{
  "Timestamp": "2017-12-01 12:56:30",
  "File": "EnergyPlusHome.ng3",
  "Add Wall": {
    "Type": "Wall",
    "Building": 2,
    "ID": 6,
    "Coordinates": [
      [{"x": -57.5, "y": -32.5, "z": 1}, {"x": -57.5, "y": -32.5, "z": 31}, {"x": -57.5, "y": 27.5, "z": 1}, {"x": -57.5, "y": 27.5, "z": 31}]
    ]
  }
}
```

We ignored the actions, such as “camera,”² that do not have direct effects on the design outcomes (i.e., ANE). After removing those irrelevant actions, there are about 300 actions per participant on average, and 115 are unique actions in the solarized home design problem. In the solarized parking lot design problem, the average number of design actions is about 350 after removing those trivial actions. Among these, 72 design actions are unique.

Analysis of such a high dimension action space would yield results hard to interpret. To better understand the design process and designers' sequential decision-making strategies, the FBS-based design process model introduced in Sec. 3.3 is applied. In this study, an encoding scheme (see Table 4) is established to transcribe different types of design actions to the seven design process stages, including *Formulation* (F), *Analysis* (A), *Evaluation* (E), *Synthesis* (S), *Reformulation 1* (R1), *Reformulation 2* (R2), and *Reformulation 3* (R3). In our design problem, adding any component such as add wall, add a solar panel, etc. refers to *Formulation*. Designers add components in order to construct the artifact to achieve the desired objective. According to Table 2, *Synthesis* occurs when parameters of a component are tuned to

²We observe that there is a large number of camera action in the collected design sequence data (more than 30%). Majority of the camera view is more likely to be an unconscious mouse operation or a habit while designers are pondering on their designs.

Table 4 Mapping of design actions to design process stage

Design process	Type of design actions
Formulation	Add any design component
Analysis	Analysis of annual net energy
Synthesis	Edit any component
Evaluation	Cost analysis
Reformulation 1	Remove structure
Reformulation 2	Remove solar device
Reformulation 3	Remove other component

Design process stages	F	A	E	S	R1	R2	R3
Formulation	1	0	0	0	0	0	0
Analysis	0	1	0	0	0	0	0
Reformulation 1	0	0	0	0	1	0	0
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
Analysis	0	1	0	0	0	0	0

Fig. 4 One-hot vector representation of a sequence

achieve the expected behaviors. So, the action of editing any components refers to *Synthesis*. When designers analyze the ANE of their solar system designs, this action refers to *Analysis* because designers aim to obtain the behavior from the generated structure. To compare the expected behavior and the actual structural behavior, designers check whether the design cost exceeds the given budget or not. According to Table 2, this can be defined as *Evaluation*. Finally, designers remove structures and regenerate new structures to meet the design requirements and their own intrinsic criteria. When designers remove structure related components such as walls, windows, or doors, these actions are referred to as *Reformulation 1*. However, when designers remove the roof, this action is primarily driven by the obtained expected behavior of design, e.g., the ANE does not meet the objective. So, they modify the roof style in order to put more solar panels for the potential increase of ANE. According to Table 2, these actions can be defined as *Reformulation 2*. Finally, if designers remove other structures, such as trees, these design actions are defined as *Reformulation 3* because these modifications are merely based on the formulation process. We did not consider *Documentation* because Energy3D automatically documents all the design process. Therefore, designers are not required to report their designs separately. The application of the FBS model helps map the design action space to the design thinking space to better understand the design rationale and the discovery of sequential design patterns. This treatment also helps dimension reduction that is useful to reduce the effect of the “curse of dimensionality” [36].

Given a set of sequential text data, we must encode the sequences so that it can be implemented by neural networks. The most popular encoding technique is known as “one-hot encoding” [37]. One-hot encoding transforms a single variable of n observations with m distinct variables into m binary variable with n observations. Each observation indicates the presence (1) for the corresponding position of that variable and absence (0) in all other dimensions. Figure 4 shows an example of the one-hot vector presentation of a design sequence.

5 Result and Discussion

In this section, we present the results of the LSTM and FNN models and compare them with the models that are commonly used in existing literature, such as the MM and the HMM. Additionally, we develop a repetitive model (REP model) for comparison because we found from our previous study [1] that designers quite frequently repeat the previous design action in the CAD

environment. For example, we found that in the solarized home design problem, on average, 51.2% of design actions were simply repeating the action in the previous step. In the solarized parking lot design, we found the repetition rate is at about 59.3%. So, in the REP model, we simply use the average percentage of occurrence of each design stage as the model to predict the next design stage with the highest percentage value. Finally, a random model is presented as the benchmark for all the models in comparison. The purpose is to examine whether the design sequences indeed follow certain patterns or just randomness. In this study, since there are seven design process stages, the prediction of the next stage will be a random selection of one process stage from seven following a uniform distribution. Thus, every process stage has a probability of 1/7 to be selected.

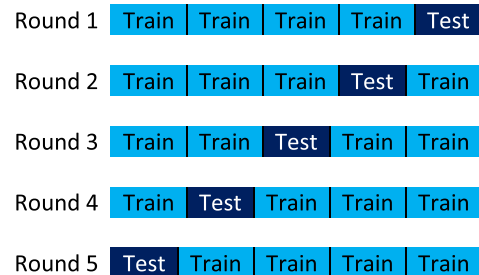
In the following two sections, we first evaluate the performance of different models in terms of prediction accuracy, precision, recall, and F1 score regardless of the category of design actions (i.e., the design process stages defined by the FBS model). Next, we perform an in-depth analysis of how accurately each design process stage in the next step can be predicted and compare the performance of different models using the metrics of the area under the receiver operating curve (AUROC).

5.1 Evaluation of Model Performance at the Level of the Entire Sequence.

To validate the models, we adopt the k-fold cross-validation [38] technique, where we divide our data into five folds. First, we use any four folds to train the models and leave the remaining fold for validation purposes. Next, we train the models on a new combination of 4 folds, including the previously withheld fold, and validate the model again with the remaining one. In this way, we iterate through all over the five rounds. An illustration of the fivefold cross-validation method is shown in Fig. 5.

Keras deep learning library [39] is used to run the HMM, FNN, and LSTM model, and we programmed for the Markov chain model. While going through each of the rounds, the training data set performs forward pass and backward pass (a.k.a. backpropagation) [40] in order to update the models’ parameters (including both weight values and bias values). When the entire dataset is passed forward and backward through the neural network, it’s called one *epoch*. During testing, we predict the next action (a_{t+1}) by passing the previous actions from time 0 to t as the input into the trained model. So, if a design sequence has n actions, then $n - 1$ predictions will be made. Then, by comparing with the real observation of a design sequence, we count the total number of correctly predicted actions (n_{cp}) and divide it by the total number of predictions, i.e., $n - 1$. In this way, we get the prediction accuracy of that model in every epoch. In this study, only the prediction accuracy of the last epoch (when the model is fully trained) in each round is taken, and the average from five rounds is used as the metric for evaluating a model’s predictive power. The mathematical expression of this metric is as follows:

$$\text{Prediction accuracy} = \frac{1}{R} \sum_{i=1}^R \left(\frac{n_{cp}}{n_i^{\max} - 1} \right) \quad (1)$$

**Fig. 5 Training and testing data split according to fivefold cross-validation technique**

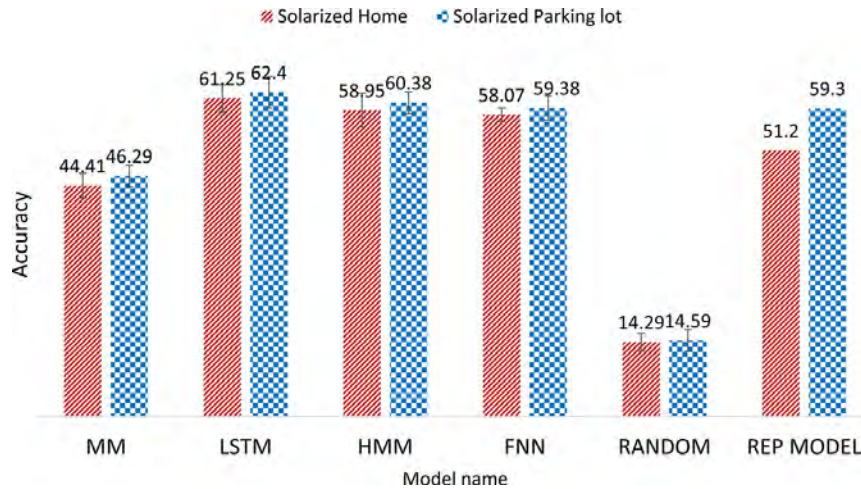


Fig. 6 Testing accuracy of different predictive models

where R denotes the number of rounds for cross-validation. $R = 5$ in this study. n_i^{max} is the maximal number of actions of the design sequence (i.e., the length of the longest design sequence) in round i . The models were trained by the stochastic gradient descent algorithm [41] with a learning rate of 0.001. This learning rate is determined by trial and error for producing the best accuracy, and we found all the models converge after 40 epochs. Section 5.3 presents a sensitivity analysis of the hyperparameters.

MM provides the prediction of the design process state in the next step based on the state in the current time-step. With the design sequence as input, training of an MM will produce a 7×7 transition probability matrix because the seven design process stages in the FBS model represent seven states in MM. Each of the entries of the matrix defines the probability of one design stage transitioning to the next design stage. We follow the same structure in Fig. 5 to train and test MM. The trained model (i.e., the transition probability matrix) is an aggregation by averaging the matrices obtained from every designer in the training dataset. When testing a MM model, for every given FBS design process stage in a design sequence, the MM will predict seven probabilities of design stages following that given stage, and the one with the highest probability is picked for comparing with the real data, and then the prediction accuracy is reported. MM is not associated with any external parameters. So, we calculate the prediction accuracy just based on the transition probability matrix.

Figure 6 shows a comparison of the testing accuracy of all the models for both the solarized home design and the solarized parking lot design. The baseline model, i.e., the random model, shows the least accuracy of 14.29% (standard deviation 1.66%) and 14.59% (standard deviation 2.02%) for solarized home and solarized parking lot, respectively. The accuracy of the REP model is much higher than the random model, which is about 51.2% for the solarized home dataset and 59.3% for the solarized parking lot dataset. This is because the REP model is developed based on the simple repetition process in design, and it is observed that when designers were working on this design problem using CAD software, many of them repeated their previous action very frequently, and the REP model captures such a pattern. We present only the prediction accuracy for the random model and the REP model because the training process is not needed in these models. The accuracy of the REP model for the solarized parking lot design is higher than the solarized home design. This is probably due to the reason that the number of unique design actions in the solarized parking lot design is lower than those in the other design problem. Therefore, the design actions can be more frequently repeated in the former design context, and this is well captured by the REP model.

It is shown from Fig. 6 that both MM and HMM yields better performance than the random model for both datasets. In the solarized

home design problem, the prediction accuracy of MM and HMM are 44.41% and 58.95%, respectively. Similarly, in the solarized parking lot, the accuracies for MM and HMM are 46.29% and 60.38%, respectively. This indicates that designers' actions indeed follow certain patterns and are not random. In MM, each action is dependent only on the action of the previous step. As a consequence, MM does not encode long-term "memory" of past events in the prediction. On the other hand, the inclusion of hidden-state architecture in HMM allows it to "remember" the past state. As in design, designers do have to refer to past information in guiding their future design decisions, the successful modeling of past information into the hidden state may be the reason why HMM has significantly higher prediction accuracy (the average is 58.1%) than those of MM and REP model. This observation echoes many of the existing studies on the comparison between MM and HMM, and the conclusion that HMM outperforms MM [42] is very likely due to the reason that HMM can better model the interdependencies between past states and current state.

We also observe that HMM slightly outperforms the FNN model on average. FNN gives an average prediction accuracy of 58.07% (with a standard deviation of 2.14%), which is 0.88% lower than that of the HMM in the solarized home design. For the parking lot dataset, FNN achieves a prediction accuracy of 59.38% (with a standard deviation of 2.3%), which is 1% lower than HMM. The ability to pass information from the previous states to the current hidden state makes HMM a better predictor than FNN in this study. FNN does not essentially have a hidden state as it does not consider feedback loop in hidden layers and hidden units are not connected (see Sec. 3.2.1 for the architecture of FNN).

Among all of the models, LSTM produces the highest prediction accuracy in both datasets. The prediction accuracy for the solarized home design and the parking lot design is 61.25% and 62.4%, respectively. The significance of the difference between the deep learning models and the existing models are assessed using the paired t -test. Among the existing models, the prediction accuracy of the HMM is close to the deep learning models (i.e., LSTM and FNN). Therefore, using this instance, the null hypothesis (H_0) of the test is that the mean of the prediction accuracy of the deep learning models is equal to that of the HMM model. The alternative hypothesis (H_a) is that the mean of the accuracy of the deep learning models is higher than that of the HMM. With the level of significance 0.05, the p -value indicates (0.04756 for the solarized home design and 0.03234 for the solarized parking lot design) that, for both datasets, the prediction accuracy of the LSTM model is higher than that of the HMM, and the difference is statistically significant; however, HMM is not significantly better than FNN as indicated by the p -values (0.2339 and 0.2088 for solarized home

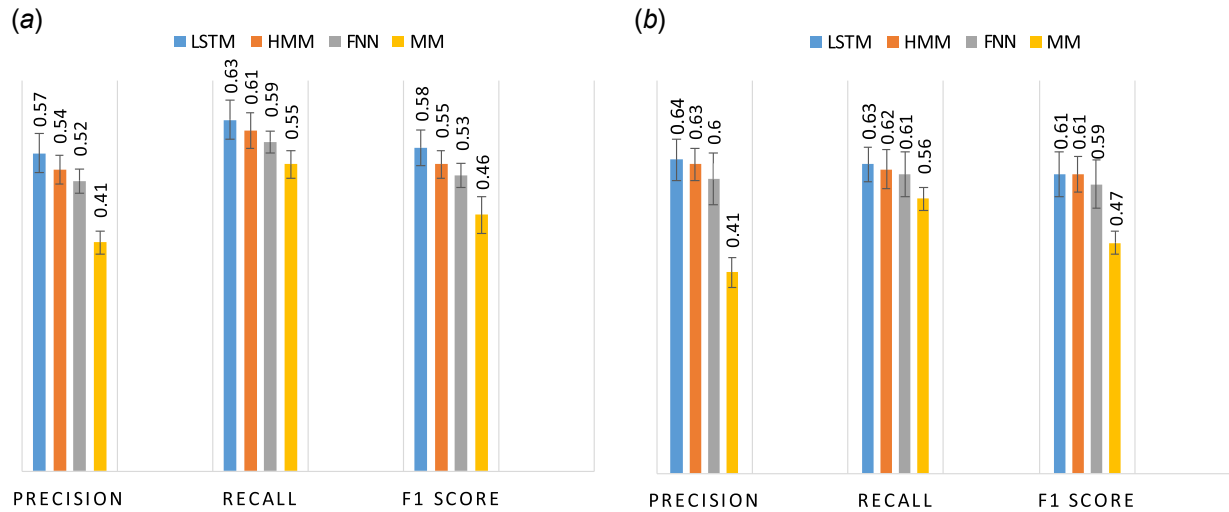


Fig. 7 F1, Precision, and Recall score for different models: (a) solarized home design and (b) solarized parking lot design

and solarized parking lot, respectively). The results imply that during the systems design process, designers' future actions do have strong dependencies with past design information. For example, in the solarized home design problem, we observe that one designer, most of the time, analyzes "Building cost" after adding several new components, such as "Add window" → "Edit window" → "Building cost", and again, "Add Rack" → "Edit Rack" → "Add SolarPanel" → "Building cost". This infers that after adding new components, this designer started configuring the related components to try improving the design performance and check if the total cost is still within the budget. Since LSTM leverages longer "memory" of past events and their interconnections in predicting future states, its architecture best resembles designers' decision-making process, and this is probably the reason why it yields the best performance in this study. LSTM's highest prediction accuracy also implies that designer does not only recall short-term memory (like what MM and HMM do) but also use long-term memory information in their design process.

Generally, the models that encode longer "memory" in their architecture performs better in predicting designers' future actions. The possible reason is that in systems design, there are many design variables that are interdependent, and designers may not be able to immediately understand such a complex relationship. One method that can help understand the interdependencies among design variables and their effects on the design objective is to constantly change the variables and then run simulations to see how it would affect the objective value. Since there are multiple variables in these design case studies (as presented in Table 1), designers often perform a series of configurations (such as "Add wall" → "Edit wall" → "Edit Foundation" → "Add Roof" → "Add Rack" for home design problem and "Add solar panel" → "Edit panel" → "Change pole height" for solarized parking lot design problem), and then perform ANE analysis cost evaluation. These processes may have to be repeated several times in order to find the best configuration/combination of design variables for the desired objective. Such a pattern can reflect designers' exploration-exploitation strategies for design tradeoff (i.e., the sequential decision-making strategy) and their design heuristics. The results indicate that the hidden states of LSTM and HMM work as a memory unit seem to well capture those design patterns.

In addition to the prediction accuracy defined in Eq. (1), we also report the metrics, including Precision, Recall, and F1 score (see Eqs. (2)–(4)), for comparison

$$\text{Precision} = \frac{\text{True positive (TP)}}{\text{True positive (TP)} + \text{False positive (FP)}} \quad (2)$$

$$\text{Recall} = \frac{\text{True positive (TP)}}{\text{True positive (TP)} + \text{False negative (FN)}} \quad (3)$$

$$\text{F1 score} = 2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \quad (4)$$

These three metrics are often used simultaneously, as each of these metrics reveals different aspects of a model's predictive power [43]. For example, there might be a case where precision is higher, but the recall is lower than the other models. In that case, for a proper evaluation, the F1 score, which takes the harmonic means of the precision and recall, can be used. Figure 7 shows the result of the scores of the metrics of different models for both case studies. Among all the models, LSTM outperforms the other models, especially in the solarized home dataset. LSTM archives about 0.57, 0.63, and 0.58 for precision, recall, and F1 score, respectively, while the nearest scores for HMM are 0.54, 0.61, and 0.55, respectively. The same conclusion holds in the parking lot dataset.

5.2 Evaluation of Model Performance at Each Category of Design Actions. In order to understand the models' performance at a finer resolution, we check how well each model can predict each category of design actions, i.e., the design process stage defined by the FBS design process model. To achieve this, we adopt receiver operating characteristics (ROC) curve [44] as the method evaluates the model's two operating characteristics (the true positive rate and the false positive rate) on each design process stage under different binary threshold values from 0 to 1.

After obtaining the ROC curves for each design process stage, the area under the ROC curve (AUROC) is used to provide one single metric which aggregates the predictive performance cross the thresholds so that we can compare in which design process stage does the model perform better. A larger AUROC indicates a better predictive performance. Figure 8 shows an example of LSTM's ROC curve of each design process stage in one fold of prediction for the solarized home design problem. If the AUROC values from all the five folds are averaged, we obtain Fig. 9. For example, the AUROC of *Formulation* and *Analysis* for the LSTM model in the solarized home design problem reaches a maximum of 0.82 and 0.80, respectively, among the seven design process categories. LSTM model also produces a decent AUROC score (0.77) for *Evaluation*. These results imply that designers tend to enter into these design stages after completing a certain series of design tasks. For example, the designers must first construct the house, which involves many design actions related to *Formulation* (i.e., Add Wall, Add Window, etc.) and then evaluate the performance by

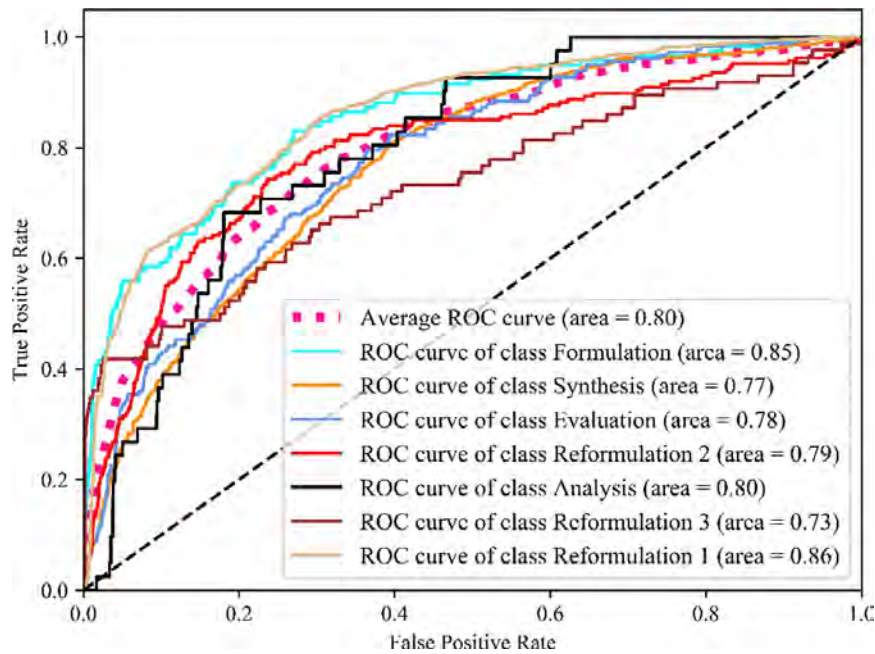


Fig. 8 Receiver operating characteristics (ROC) curves of LSTM in fold 5 for solarized home design dataset. Each curve represents the model performance for each design process stage.

simulating annual net energy (*Analysis*) and analyzing the building cost (*Evaluation*). However, in the solarized parking lot problem, *Evaluation* achieves the highest AUROC score (0.91), which indicates that designers follow certain strong behavioral patterns while checking the design cost. Also, the AUROC scores for *Formulation* (0.83) and *Reformulation 2* (0.85) are higher than the other design process stages. The potential explanation is that as solarized parking lot design is less complex, the only design component that designers add and remove is the solar panel. While designers are adding solar panels, they frequently check if the overall design cost (i.e., *Evaluation*) exceeds the budget limit or not. Also, in order to increase the ANE, designers may need to remove existing solar panels (i.e., *Reformulation 2*) and try a new layout. These patterns of design behaviors make *Formulation* and *Reformulation 2* more easily to be captured by the model.

LSTM also produces some lower AUROC values, particularly for *Reformulation 1* and *3*, in both datasets. This is because *Reformulation* involves the design actions of removing components, such as remove a tree or remove a window. These removal actions are often paired with another *Reformulation* and/or *Formulation* actions, such as add a wall or add a window. These action pairs reflect designers' fine-tuning behaviors (exploitation) on particular design components immediately based on the observations from the CAD interface and no necessary to run a simulation for feedback to support their design decisions. Therefore, referring to the action in the last step should be sufficient for prediction, and it does not require to use long-term memory in predicting these design stages. This may also be the reason why MM can produce higher AUROC scores for *Reformulation 2* (0.60 in the solarized home design and 0.57 in the parking lot design, respectively). For example, *Reformulation 2* contains the design actions related to the removal of solar panels. As solar panels directly affect the system performance, most designers spent a significant amount of time fine-tuning (e.g., add, remove and then add back again) this component, and therefore, there exist a large number of action pairs of "Add Solar Panel → Remove Solar Panel" in the design sequence. Since MM predicts the state only one time-step ahead based on the current state, it captures this design pattern very well. But on the other hand, it does not effectively capture the

patterns that involve longer historical information, such as *Evaluation* and *Analysis* stages in the solarized home design problem, as compared with the other models. However, in the parking lot dataset, the more frequent short-term action pairs are observed, such as *Evaluation*→*Evaluation* and *Analysis*→*Evaluation* (see Table 5 for the transition probabilities from other design processes to *Evaluation* as an example), this pattern can be better captured by MM; therefore, their AUROC scores are higher.

Please note that in the solarized house design problem, MM has the least AUROC for *Formulation*. This is because MM is derived from the frequency of the event. In the design, the repetition of *Formulation* (corresponds to adding components) does not occur frequently. For example, once a designer finishes adding all the necessary components, e.g., "Add Wall" and "Add Window," she/he would never take those actions again because the house has already been established. Instead, she/he tends to start fine-tuning the associated parameters through the actions of "Edit Wall" and "Edit Window" (i.e., the actions related to *Synthesis*).

If we take an average for the AUROC scores from every design process stage, that average value can be used to compare the performance between different models, as shown in the last column of Fig. 9. For the solarized home design problem, we observe that, on average, the LSTM model outperforms the other models with the AUROC of 0.78. The HMM model (0.75) and FNN (0.63) achieve a lower AUROC score than the LSTM model. This indicates that even if HMM and FNN take historical design information into their prediction, they do not effectively process that information during the model training as what LSTM does. Both FNN and HMM perform relatively better on average across all the design process stages than the MM (0.45).

In the parking lot design problem, the LSTM outperforms the other models as well (0.82). However, other predictive models such as HMM and FNN also perform better with the AUROC of 0.81 and 0.79, respectively. The AUROC score of each design process stage predicted by the models is also close to each other. This phenomenon indicates that for the less complex design problem where designers use less complex design patterns and fewer types of design actions (i.e., design variables), there are no significant differences between LSTM and other models.

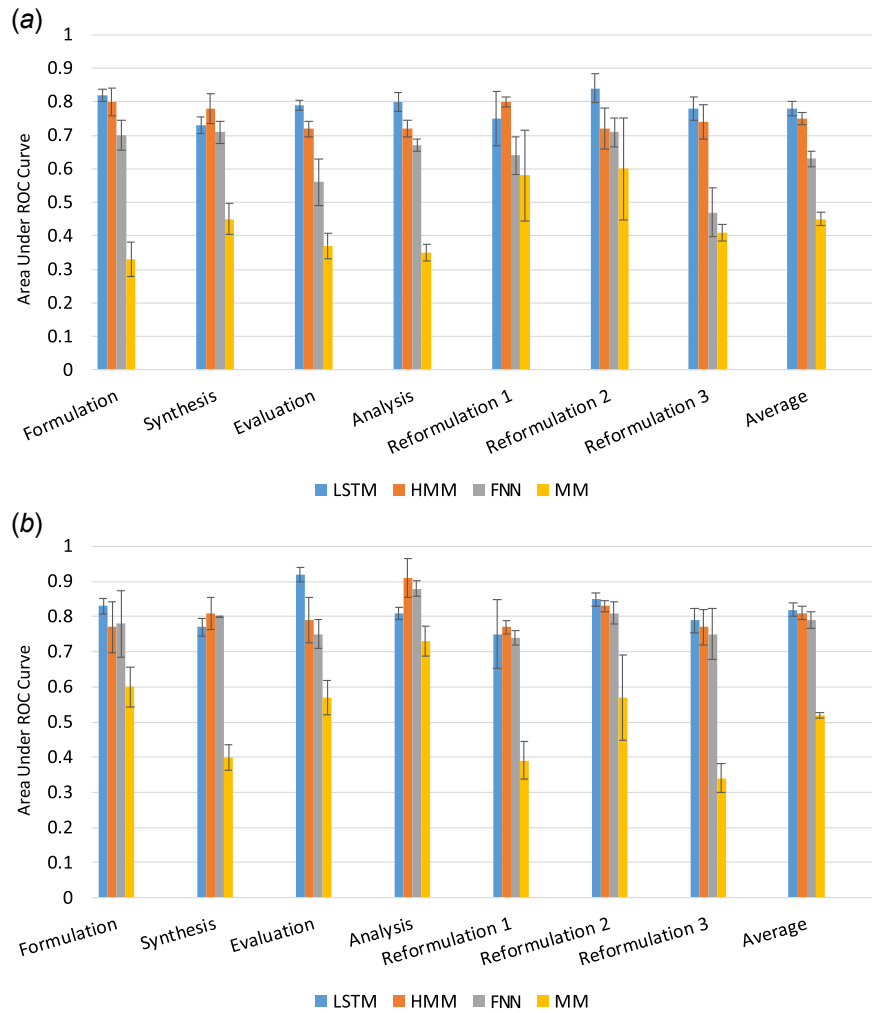


Fig. 9 Area under the receiver operating characteristics curve (AUROC) scores for different models. The average in the last column is the average AUROC value of all design process stages per model: (a) solarized home design and (b) solarized parking lot design

However, for complex design problems where various types of design actions exist, there could exist different design approaches to reaching the objective. As a result, designers' behavior may follow different patterns that are hard to be captured by models with simple structures, such as HMM and FNN. But LSTM's *gate mechanism* (e.g., input gate, forget gate, and output gates) seems well to capture and process the dependent relations between different design stages during a design process, therefore, yields the best performance regardless of the complexity of the dataset. These results cross-validate the conclusion we reached from the prediction accuracy results shown in Fig. 6.

Table 5 Transition probability from other design process stages to evaluation in both datasets

Transit to Transit from	Evaluation	
	Solarized home design	Solarized parking lot design
Analysis	0.197	0.23
Evaluation	0.061	0.2
Formulation	0.036	0.012
Reformulation 1	0.034	0.002
Reformulation 2	0.036	0.07
Reformulation 3	0.051	0.021
Synthesis	0.08	0.31

5.3 Sensitivity Analysis. When training an LSTM model, there are several pre-determined hyperparameters, such as the number of LSTM layers, LSTM size, the number of dense layers, the size of a dense layer, learning rate and dropout value. LSTM size refers to LSTM nodes in each LSTM layer. The fully connected layer indicates the number of layers of the feed-forward network. The size of a fully connected layer indicates the number of nodes in each dense layer. Dropout is the value of dropout regularization. In order to prevent the model from overfitting, we use dropout regularization [45] with two different values. The learning rate is the converge rate used in the stochastic gradient descent algorithm in backpropagation.

To investigate how the prediction accuracy would be affected by these hyperparameters, we perform a sensitivity analysis by changing the values of these parameters and study the corresponding prediction accuracies. In the experiment, we use one layer of LSTM for all the settings with a various number of LSTM nodes. Table 6 shows the test accuracy of the LSTM models with different hyperparameter settings. From all the settings, it is observed that the model with one fully connected layer performs better (i.e., above 58%) than the models with two fully connected layers (i.e., 56.17% for the solarized parking lot dataset and 54.95% for the solarized home design dataset). Given the same number of fully connected layers and the same fully connected size, a learning rate of 0.1 produces relatively lower performance (57.50%) than those of other settings. But the dropout rate (changing from 0.3 to 0.2) and the LSTM

Table 6 Different hyperparameter settings for LSTM model

No.	LSTM size	Fully connected layer	Fully connected layer size	Dropout	Learning rate	Testing accuracy	
						Solarized home	Solarized parking lot
1	256	1	7	0.3	0.1	57.50%	56.27%
2	256	1	7	0.3	0.001	61.25%	62.4%
3	256	1	7	0.2	0.01	58.97%	60.81%
4	128	1	7	0.3	0.01	59.16%	59.38%
5	128	1	7	0.3	0.1	58.08%	57.49%
6	256	2	128 and 7	0.2	0.1	54.95%	56.17%

size (changing from 256 to 128 nodes) do not influence the model significantly. Among all the settings, it is found that the model with LSTM unit 256, dropout value with 0.3, and learning rate with 0.001 provides the best accuracy in both datasets.

6 Conclusion and Closing Thoughts

In this study, a deep learning approach is developed to analyze and predict the sequential design decisions in the systems design context. We use Energy3D as the research platform to conduct design challenges and collect designers' sequential design behavioral data. Then, the FBS-based design process model is adopted to transform the sequential design action data into the sequential design process data. Based on the design process data, we adopted two deep learning models, i.e., the FNN and the LSTM, to predict designers' next immediate design process stage. These deep learning models are evaluated with different performance metrics, including testing accuracy, Precision, Recall, F1 score, and area under the ROC curve. Their predictive performances are compared with the other four models, including a MM, an HMM, a repetitive model, and a random model. The predictive power is assessed at the level of the entire design sequence as well as at the level of each design process stage.

We found that, on average, the LSTM model outperforms all the other models, while FNN shows lower performance than traditionally used HMM. From the ROC curve analysis, we found that in both datasets, LSTM yields overall better performance for all of the design process. In contrast, the predictive performance of the other models is not consistent. Moreover, from this study, we also observe that for design problems that are less complex and involve a fewer number of design variables, predictive models perform similarly. For complex design problems, the performance of the predictive models differs. However, regardless of the design complexity, LSTM performs better than the other models. With these findings, we conclude that both short-term and long-term memories have together influenced human sequential design decision-making. The neglect of either aspect in the modeling would lead to inadequate prediction accuracy. However, *such an effect is not always significant for all design actions in every stage* because, indeed, it is found that in predicting certain design actions (e.g., Remove Wall, Remove Window), LSTM was not the best model.

This work shows that deep learning, particularly the LSTM, can be a stepping stone for modeling and predicting sequential decision-making in engineering design and facilitating design automation. By predicting the design process stage at both the aggregated level and individual level, the models exhibit designers' thinking and strategies. The approach introduced in this paper is general and can be implemented in other design contexts to understand the design thinking and decision-making strategies as long as the data of the design action sequences are available.

However, there are some limitations to our approach. For example, the accuracy we obtained in this study is below the state-of-the-art accuracy of the deep learning methods in other fields. This is because design activities could be diverse and complicated in complex systems design, and it is challenging to learn prominent patterns due to the heterogeneities within the training

dataset. This is different from other types of human behaviors, such as consumer behaviors, where individual shopping mode shows more tractable patterns and would be more easily to be learned by deep neural networks. Additionally, indifferent from other fields where a large amount of human behavioral data can be obtained for model training, such as the customers' shopping records and purchase history collected from Amazon, the amount of data collected from human-subject experiments based on students is not ideal. But we were trying to overcome this limitation, e.g., by increasing the amount of data in temporal dimensions in addition to the number of human subjects, and our approach showed better performance compared with those existing methods.

In addition, we used the FBS design process model to encode the design action data as well as to reduce its dimensionality. However, the prediction is made at the design process stage level. In order to predict design actions, various embedding techniques (e.g., word2vec [46], Glove [47], etc.) could help because both embeddings and the FBS model serve the purpose of mapping the high-dimensional sequential data to a low-dimensional latent space, yet embedding provides both encoding and decoding schemes that can be used to map low-dimensional data back to the high-dimensional design action data.

However, current embedding techniques do not take care of the nature of the design process and cannot help interpreting designers' sequential decisions. In the future work, we would like to establish a theoretical bond between the ontological model (such as the FBS used in this study) and embedding technique and identify embedding technique that could best capture the latent design thinking space of a particular design ontology model. Also, to improve the prediction accuracy as well as to further validate the proposed model, we will collect more data based on our designed experiment and research new models, such as the Markov decision process, which consider the "reward" information (i.e., the design objective values) along with the design action sequence to test the feasibility of reinforcement learning techniques in design thinking research.

Finally, the proposed approach can be used to study the difference between experts and novice designers. It would be expected that experts show different patterns compared with the novice designers, and the proposed approach is promising to learn such patterns and help discover who would better utilize memory in guiding their future design decisions. But to answer this research question requires a rigorous design of the human-subject experiment and relies on the validity of the developed approach. This is the reason why in this first study, we focused on investigating the performance of the model and did a comparative study benchmarking on commonly used models in current literature before we apply this approach to answer other interesting research questions.

Acknowledgment

The authors acknowledge the advice provided by Dr. Xintao Wu and Dr. Shuhan Yuan on the topic of recurrent neural network modeling. The authors gratefully acknowledge the financial support from NSF-CMMI-1842588 and NSF-DRL-2105695.

Conflict of Interest

There are no conflicts of interest.

Data Availability Statement

The datasets generated and supporting the findings of this article are obtainable from the corresponding author upon reasonable request. The authors attest that all data for this study are included in the paper.

References

- [1] Rahman, M. H., Gashler, M., Xie, C., and Sha, Z., 2018, "Automatic Clustering of Sequential Design Behaviors," ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Quebec City, Quebec, Canada, Aug. 26–29.
- [2] Panchal, J. H., Sha, Z., and Kannan, K. N., 2017, "Understanding Design Decisions Under Competition Using Games With Information Acquisition and a Behavioral Experiment," *ASME J. Mech. Des.*, **139**(9), p. 091402.
- [3] Sexton, T., and Ren, M. Y., 2017, "Learning an Optimization Algorithm Through Human Design Iterations," *ASME J. Mech. Des.*, **139**(10), p. 101404.
- [4] Baxter, G., and Sommerville, I., 2011, "Socio-Technical Systems: From Design Methods to Systems Engineering," *Interact. Comput.*, **23**(1), pp. 4–17.
- [5] Sha, Z., Chaudhari, A. M., and Panchal, J. H., 2019, "Modeling Participation Behaviors in Design Crowdsourcing Using a Bipartite Network-Based Approach," *ASME J. Comput. Inf. Sci. Eng.*, **19**(3), p. 031010.
- [6] Fuge, M., Tee, K., Agogino, A., and Maton, N., 2014, "Analysis of Collaborative Design Networks: A Case Study of Openideo," *ASME J. Comput. Inf. Sci. Eng.*, **14**(2), p. 021009.
- [7] Brockmann, E. N., and Anthony, W. P., 1998, "The Influence of Tacit Knowledge and Collective Mind on Strategic Planning," *J. Manag. Issues*, **10**(2), pp. 204–222.
- [8] Collobert, R., and Weston, J., 2008, "A Unified Architecture for Natural Language Processing: Deep Neural Networks With Multitask Learning," Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, July, pp. 160–167.
- [9] Miotto, R., Wang, F., Wang, S., Jiang, X., and Dudley, J. T., 2017, "Deep Learning for Healthcare: Review, Opportunities and Challenges," *Brief. Bioinform.*, **19**(6), pp. 1236–1246.
- [10] He, K., Zhang, X., Ren, S., and Sun, J., 2016, "Deep Residual Learning for Image Recognition," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, June 27–30, pp. 770–778.
- [11] Amershi, S., Conati, C., and Maclaren, H., 2006, "Using Feature Selection and Unsupervised Clustering to Identify Affective Expressions in Educational Games," In Proceedings of The Intelligent Tutoring Systems Workshop on Motivational and Affective Issues in ITS (ITS 2006), Jhongli, Taiwan, June 26–30.
- [12] Kan, J. W. T., Gero, J. S., and Press, C. R. C., 2009, "Using the FBS Ontology to Capture Semantic Design Information in Design Protocol Studies," *About: Designing. Analysing Design Meetings*, pp. 213–229.
- [13] Yu, R. O. N. G., and Gero, J. S., 2015, "An Empirical Foundation for Design Patterns in Parametric Design," 20th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA), Daegu, South Korea, May, pp. 20–23.
- [14] McComb, C., Cagan, J., and Kotovsky, K., 2017, "Capturing Human Sequence-Learning Abilities in Configuration Design Tasks Through Markov Chains," *ASME J. Mech. Des.*, **139**(9), p. 091101.
- [15] McComb, C., Cagan, J., and Kotovsky, K., 2017, "Mining Process Heuristics From Designer Action Data via Hidden Markov Models," *ASME J. Mech. Des.*, **139**(11), p. 111412.
- [16] Sha, Z., Kannan, K. N., and Panchal, J. H., 2015, "Behavioral Experimentation and Game Theory in Engineering Systems Design," *ASME J. Mech. Des.*, **137**(5), p. 051405.
- [17] Chaudhari, A. M., and Panchal, J. H., 2019, "An Experimental Study of Human Decisions in Sequential Information Acquisition in Design: Impact of Cost and Task Complexity," *Research Into Design for a Connected World*, Springer, Singapore, pp. 321–332.
- [18] Chaudhari, A. M., Bilonis, I., and Panchal, J. H., 2020, "Descriptive Models of Sequential Decisions in Engineering Design: An Experimental Study," *ASME J. Mech. Des.*, **142**(8), p. 081704.
- [19] Shergadwala, M., Bilonis, I., Kannan, K. N., and Panchal, J. H., 2018, "Quantifying the Impact of Domain Knowledge and Problem Framing on Sequential Decisions in Engineering Design," *ASME J. Mech. Des.*, **140**(10), p. 101402.
- [20] Shergadwala, M. N., Panchal, J. H., and Bilonis, I., 2019, "Quantifying the Influence of Information Sharing About Competitor's Performance on a Participant's Sequential Design Behaviors in Design Contests," *ASME J. Mech. Des.*, under review.
- [21] Smith, R. P., and Eppinger, S. D., 1997, "A Predictive Model of Sequential Iteration in Engineering Design," *Manage. Sci.*, **43**(8), pp. 1104–1120.
- [22] Miller, S. W., Simpson, T. W., and Yukish, M. A., 2015, "Design as a Sequential Decision Process: A Method for Reducing Design set Space Using Models to Bound Objectives," ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Boston, MA, Aug. 2–5, p. V02AT03A020.
- [23] Griffin, S., Welton, N. J., and Claxton, K., 2010, "Exploring the Research Decision Space: The Expected Value of Information for Sequential Research Designs," *Med. Decis. Mak.*, **30**(2), pp. 155–162.
- [24] Meier, C., Yassine, A. A., and Browning, T. R., 2007, "Design Process Sequencing With Competent Genetic Algorithms," *ASME J. Mech. Des.*, **129**(6), pp. 566–585.
- [25] Duff, M. O., and Barto, A., 2002, *Optimal Learning: Computational Procedures for Bayes-Adaptive Markov Decision Processes*, University of Massachusetts at Amherst, Amherst, MA.
- [26] Raina, A., McComb, C., and Cagan, J., 2019, "Learning to Design From Humans: Imitating Human Designers Through Deep Learning," *ASME J. Mech. Des.*, **141**(11), p. 111102.
- [27] Oh, S., Jung, Y., Kim, S., Lee, I., and Kang, N., 2019, "Deep Generative Design: Integration of Topology Optimization and Generative Models," *ASME J. Mech. Des.*, **141**(11), p. 111405.
- [28] Stump, G. M., Miller, S. W., Yukish, M. A., Simpson, T. W., and Tucker, C., 2019, "Spatial Grammar-Based Recurrent Neural Network for Design Form and Behavior Optimization," *ASME J. Mech. Des.*, **141**(12), p. 124501.
- [29] Sundermeyer, M., Ney, H., and Schlüter, R., 2015, "From Feedforward to Recurrent LSTM Neural Networks for Language Modeling," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, **23**(3), pp. 517–529.
- [30] Hanin, B., 2019, "Universal Function Approximation by Deep Neural Nets with Bounded Width and Relu Activations," *Mathematics*, **7**(10), pp. 992.
- [31] Bengio, Y., Simard, P., and Frasconi, R., 1994, "Learning Long-Term Dependencies With Gradient Descent is Difficult," *IEEE Trans. Neural Networks*, **5**(2), pp. 157–166.
- [32] Hochreiter, S., and Schmidhuber, J., 1997, "Long Short-Term Memory," *Neural Comput.*, **9**(8), pp. 1735–1780.
- [33] Gero, J. S., 1990, "Design Prototypes: A Knowledge Representation Schema for Design," *AI Mag.*, **11**(4), p. 26.
- [34] Xie, C., Schimpf, C., Chao, J., Nourian, S., and Massicotte, J., 2018, "Learning and Teaching Engineering Design Through Modeling and Simulation on a CAD Platform," *Comput. Appl. Eng. Educ.*, **26**(4), pp. 824–840.
- [35] Rahman, M. H., Schimpf, C., Xie, C., and Sha, Z., 2019, "A Computer-Aided Design Based Research Platform for Design Thinking Studies," *ASME J. Mech. Des.*, **141**(12), p. 121102.
- [36] Verleysen, M., and François, D., 2005, "The Curse of Dimensionality in Data Mining and Time Series Prediction," International Work-Conference on Artificial Neural Networks, Barcelona, Spain, June, pp. 758–770.
- [37] Potdar, K., Pardawala, T. S., and Pai, C. D., 2017, "A Comparative Study of Categorical Variable Encoding Techniques for Neural Network Classifiers," *Int. J. Comput. Appl.*, **175**(4), pp. 7–9.
- [38] Kohavi, R., et al., 1995, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection," *J. Jcai*, **14**(2), pp. 1137–1145.
- [39] Chollet, F., 2017, *Deep Learning with Python*, 1st, Manning Publications Co., USA.
- [40] Friedman, J., Hastie, T., and Tibshirani, R., 2001, *The Elements of Statistical Learning*, Vol. 1, No. 10, Springer Series in Statistics, New York.
- [41] Ruder, S., 2016, "An Overview of Gradient Descent Optimization Algorithms," *arXiv Prepr. arXiv1609.04747*.
- [42] Lee, K. C., Phon-Amnuaisuk, S., and Ting, C. Y., 2010, "A Comparison of HMM, Naive Bayesian, and Markov Model in Exploiting Knowledge Content in Digital Ink: A Case Study on Handwritten Music Notation Recognition," 2010 IEEE International Conference on Multimedia and Expo, Singapore, July 19–23, pp. 292–297.
- [43] Sha, Z., Huang, Y., Fu, S., Wang, M., Fu, Y., Contractor, N., and Chen, W., 2018, "A Network-Based Approach to Modeling and Predicting Product Consideration Relations," *Complexity*, **2018**, pp. 1–14.
- [44] Fawcett, T., 2006, "An Introduction to ROC Analysis," *Pattern Recognit. Lett.*, **27**(8), pp. 861–874.
- [45] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R., 2014, "Dropout: A Simple Way to Prevent Neural Networks From Overfitting," *J. Mach. Learn. Res.*, **15**(1), pp. 1929–1958.
- [46] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J., 2013, "Distributed Representations of Words and Phrases and Their Compositionality," Advances in Neural Information Processing Systems, Lake Tahoe, NV, December, pp. 3111–3119.
- [47] Pennington, J., Socher, R., and Manning, C. D., 2014, "Glove: Global Vectors for Word Representation," Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, October, pp. 1532–1543.