Adversarial Creation of a Smart Home Testbed for Novelty Detection

Jarren Briscoe, Assefaw Gebremedhin, Lawrence B. Holder, and Diane J. Cook
School of Electrical Engineering and Computer Science
Washington State University

Abstract

This paper introduces SHGAN, an adversarially-trained system to create smart home testbeds for novelty detection. We design a unique structure to model the complexities of smart home data and generate an arbitrary amount of realistic sensor readings. We validate the approach based on data collected from real-world smart homes and discuss methods for utilizing this testbed to detect and handle novel smart home automated tasks.

Introduction

Individuals spend much of their time in their homes, turning these places into sanctuaries. Embedding an AI system into the environment also turns these settings into smart homes. An intelligent agent can perceive the state of the physical environment and residents using sensors, reason about the state using AI techniques, and take actions to achieve goals such as maximizing security, minimizing resource consumption, and maintaining resident health (Dahmen et al. 2017; Schmitter-Edgecombe and Cook 2021; Zhang, Srivastava, and Cook 2020).

Achieving such goals in a smart home is challenging because of the extreme variability in human behavior and environmental conditions (Arocha 2021; Olthof, Hasselman, and Lichtwarck-Aschoff 2020). As a result, smart homes provide a valuable testbed to evaluate the ability of an AI system to react to unexpected events. Frameworks exist to evaluate AI systems, but these are frequently based on a set of similar tests (e.g., OpenAI Gym (Brockman et al. 2016)) or a single complex environment (e.g., Urban Combat Testbed (Youngblood et al. 2006)). In contrast, a smart home testbed is both realistic, practical, and dynamic.

Two tasks that are inherent to smart homes and other smart environments are activity recognition and activity prediction (D. Cook and Schmitter-Edgecombe 2021; Minor, Doppa, and Cook 2017). In either case, the AI system is fed sensor data from a fixed temporal window (e.g., 30 seconds) and returns a label for the current activity or the time delay until an activity of interest will next occur. These capabilities allow an agent to describe a resident's behavior routine (useful for health and security monitoring) and automate home functions in preparation for a future anticipated behavior.

One limitation of real-world testbeds, like smart homes, is the limited amount of data that are available for training and testing AI systems. In this paper, we introduce a new method for generating synthetic smart home sensor data that are reflective of human behavior found in real smart homes.

2021-08-31 07:44:55.684393 Bedroom ON
2021-08-31 07:44:54.695151 Hall ON
2021-08-31 07:44:55.704645 Bedroom OFF
2021-08-31 07:44:55.707209 Bathroom ON
2021-08-31 07:45:00.292736 Hall OFF
2021-08-31 07:45:01.124967 Bathroom OFF
2021-08-31 07:45:02.248624 Bathroom ON
2021-08-31 07:45:02.805335 Bathroom OFF
2021-08-31 07:45:03.196815 Bathroom ON

Fig. 1. (top) Floor plan of a smart home with corresponding locations for motion ("M"), door ("D"), and item ("I") sensors. (bottom) Sample of sensor data generated by the smart home.

Fig. 1 illustrates the types of data that are collected in smart environments such as smart homes using the CASAS Smart Home in a Box kit (D. J. Cook et al. 2012). As the figure highlights, smart home testbeds are unique because 1) data are sequentially ordered and non-i.i.d., 2) data arrive at non-uniform time intervals, and 3) the underlying processes (e.g., human behavior) are not stationary but vary based on time of day. We describe a generative adversarial method that is designed to handle the unique nature of smart home data. We evaluate the method using real data collected from multiple smart homes and discuss how the testbed can be utilized to evaluate the robustness of AI systems in such "open worlds."

Related Work

Collecting human behavior data can be challenging, particularly if the data are collected longitudinally "in the wild," without constraints on the person's activities. Researchers initially created mathematical models, including Markov chains and Petri networks, to model behavior patterns (Virone et al. 2003). These models were combined with a Poisson distribution to add the corresponding sensor reading timestamps (Helal et al. 2011).

More recently, generative adversarial networks (GANs) have become the de facto standard for creating realistic artificial data. Although these have succeeded in synthesizing images that are indistinguishable from real (Schonfeld, Schiele, and Khoreva 2020), researchers have only recently considered adapting these methods to other types of data. The closest approaches to this work generate synthetic time series. WaveNet, for example, is a deep network that generates raw audio waveforms used to create music fragments or perform text-to-speech conversion (Jiao et al. 2021). TimeGAN trains an autoencoder to learn a latent representation while jointly training adversarial components to capture temporal relationships within the time series (Yoon et al. 2019). Methods that do not use GANs have also been attempted. One such method creates data by averaging a set of time series (Forestier et al. 2017). Many of these existing methods have been employed for data augmentation, to improve the accuracy of machine learning models (Dahmen and Cook 2019; Forestier et al. 2017).

Smart home data exhibit unique characteristics that are distinct from most sequence and time-series datasets. Data must be generated that are consistent with these characteristics. First, data do not arrive at a constant rate. This means that a data generator must create realistic time stamps for each sensor reading. Second, the generated series does not contain just continuous values. Instead, a sensor name must be generated with a corresponding value. In the case of ambient temperature and ambient light sensors, the corresponding value is numeric. In the case of motion and door sensors, the corresponding value is binary (e.g., motion sensors generate an ON or OFF value, door sensors generate an OPEN or CLOSED value).

Third, sensor readings are accompanied by corresponding activity labels. In these experiments, activity labels are provided by external annotators. Annotators offer ground truth labels based on information from smart home residents, a home floorplan with sensor locations as in Fig. 1, and a visualization of the sensor readings. The activity labels provide valuable context for the sensor readings. They also represent information needed for open world AI tasks such as activity recognition, activity forecasting, health assessment, and home automation. We label data in new homes based on models that were trained from these ground truth instances (D. Cook and Schmitter-Edgecombe 2021).

SHGAN

We introduce a system called Smart Home GAN (SHGAN) that generates an arbitrary amount of smart home data reflective of the data and behavior that are observed in a real smart home.[1]

Feature Space

To address the issue of timestamps, we add these to the feature space. Because timestamps must monotonically increase, we represent them as positive differentials from the previous sensor reading. We normalize time differences $\mathcal{T}$ to fall in the range [-1, 1], which can later be cast onto the desired final value range. Because we want to support finer precision of small time increments, we adjust the normalization using a nonlinear mapping, as shown in Fig. 2.

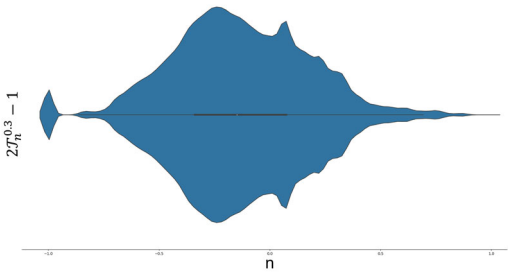

Fig. 2. $\mathcal{T}$ mapped by $2\mathcal{T}_n^k - 1$. In this case, *k=0.3*.

When addressing the second issue, generating data for continuous and binary sensors, we observe that the large number of binary outputs (one for each binary sensor, see Fig. 1 as an example) biases the output toward binary values when combining continuous and binary features into one vector. These cases are more effectively handled separately. As a result, the architecture uses two signal outputs, one for binary readings and another for continuous values. Numeric values are generated for all readings in the range [-1,1]. SHGAN computes the Wasserstein distance between the network output and binary options of -1 and 1 to select the final values for binary sensors. The generator's binary sensor name and activity label are selected through a softmax activation applied to the corresponding channels.

GAN Architecture

To generate smart home data, we start with a conditional Wasserstein GAN with gradient penalty (CWGAN+GP). One of the agents in the adversarial pair, the generator *G,*

[1] SHGAN is available at https://github.com/jbroot/SHGAN.

processes an input vector Z from a fixed distribution P_r and outputs data X following Equation 1:

$$\boldsymbol{G}: \boldsymbol{Z} \sim \boldsymbol{P_r} \rightarrow \boldsymbol{X} \sim \boldsymbol{P_\theta} \quad (1)$$

In this equation, P_r represents a Gaussian distribution with a mean of 0.0 and a standard deviation of 1.0 ($\boldsymbol{P_r} \in \mathbb{R}^{\boldsymbol{1\times128}}$) and P_θ is the output from a softmax function adjusted to fall in the range [-1,1]. Thus $\boldsymbol{P_\theta} \in \mathbb{R}_{\boldsymbol{\pm1}}^{\boldsymbol{64\times112}}$, where $\mathbb{R}_{\boldsymbol{\pm1}} \in [\boldsymbol{-1, 1}]$. That is, we map the noise vector z of size 128 to a time-step series with 64 time steps and 48 features (1 for time, 1 for signal, 32 for sensors, and 14 for activities).

SHGAN creates a window of data whose number of readings is bounded by a single generation cycle. To create arbitrarily-long sequences that can be fused, the model must maintain some memory. A bidirectional LSTM (Bi-LSTM) accomplishes this using two LSTM layers, one that passes data forward and the other backward. The layers learn context-sensitive relationships and maintain memory. In our experiments, SHGAN employs the Adam optimizer with a learning rate of 0.0002, exponential decay of 0.5 for first-moment estimates, and exponential decay of 0.9 for second-moment estimates.

In the first layer of the architecture, generator G upsamples Z via a dense layer of leaky ReLU activations (x = max(0.20, θ)). Hidden layers are upsampled by repeating the temporal step twice along the time axis. The output is then passed through a single-dimensional convolution layer. We use zero padding and a stride of one to preserve the dimensions. The feature size is scaled approximately linearly to distribute the cost of upsampling amongst the layers.

The target output is in $\mathbb{R}^{\boldsymbol{64\times48}}$. SHGAN splits the last hidden layer to predict each of the original channels with their own convolutional layer. These channels correspond to time, signal, sensor, and activity. The time and signals final layers are straightforward convolutional layers with one layer employing zero padding, a stride of one, and the tanh activation function. The sensor and activity outputs are similar except they use an adjusted softmax function. This generated data is given to the discriminator without additional processing or masking.

The GAN's discriminator D scores the generated data using the Wasserstein distance with the first moment W_1, as formalized in Equation 2.

$$\boldsymbol{W_1(S, R)} = \boldsymbol{inf_{\gamma\in\Gamma(S,R)}} \int_{\boldsymbol{MxM}} \boldsymbol{d(x, y)d\gamma(x, y)} \quad (2)$$

Intuitively, W_1 represents the minimum earth mover's distance between the synthetic (S) and real (R) data distributions. The discriminator halves the timesteps in each layer until there is only one step. The data are scaled approximately linearly using the same process as for the generator. Finally, a fully dense layer yields a single linear output. As recommended in the literature, the discriminator is trained five iterations for every one iteration of generator training (Arjovsky, Chintala, and Bottou 2017).

Experimental Results

We validate our approach to creating a smart home novelty testbed by analyzing synthetic data based on three real-world smart homes. We start by visually inspecting the data then quantify similarity between real and synthetic data.

Qualitative Analysis

Fig. 3 plots the distribution of labels that are generated for sensor and activity categories. The two graphs reflect the similarity in distribution between real and synthetic data. This figure also highlights another challenge that is faced by SHGAN. As the histograms indicate, smart home data tend to be imbalanced both among sensor types and activity categories. First, there is an imbalance among the sensor types: the door sensors (labels start with "D") and temperature sensors (labels start with "T") do not generate as many readings as light sensors (labels start with "L") and motion sensors (labels start with "M"). Additional sensors are attached throughout the smart home to reflect the battery level of smart home components. Because none of the batteries ran low, these sensors (labels start with "B") did not generate a single reading.

Second, this imbalance also exists among activity categories: activities "Work" and "Other" appear much more often than the other activities. This situation creates a challenge for SHGAN, because the generator often ignores underrepresented categories. To address this issue, SHGAN replays windows to the discriminator that contain instances of minority categories.

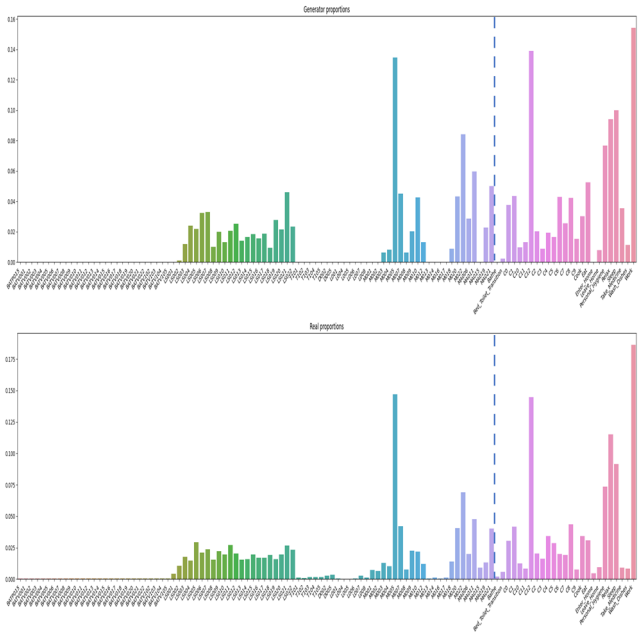

Fig. 3. Histogram of data distributions for (top) synthetic data and (bottom) real data. Entries to the left of the vertical dashed line represent sensors, entries to the right represent activities.

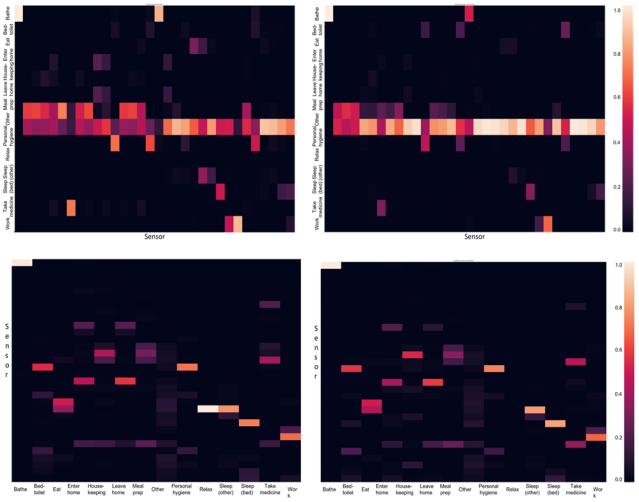

Fig. 4. Heat maps of the probabilistic relationship between the sensor identifier and the activity label for each reading in the real and synthetic smart home data. (top left) P(activity|sensor) for the real data and (top right) the corresponding synthetic data. (bottom left) P(sensor|activity) for the real data and (bottom right) the coresponding synthetic data. Colors range from black (P=0.0) to white (P=1.0). Data are generated for 14 activities and 32 sensors.

Next, we examine whether SHGAN captured the dependencies between sensors and activity labels. The heat maps in Fig. 4 show the probabilistic relationship between activities and the corresponding sensors for each reading in real and synthetic data. Each row of the figure compares real data on the left with the corresponding synthetic data on the right. Despite the issues with imbalanced data distributions, the graphs indicate the large degree of similarity that exists between the sensor and activity relationships in the real and synthetic datasets.

Quantitative Analysis

To quantitatively analyze the ability of SHGAN to generate realistic synthetic smart home data, we use three metrics. First, we estimate sequential conditional probabilities for real and synthetic data. Based on these estimates, we calculate the average difference between the bigram probabilities for sensor labels and activity labels. We report the normalized difference over all datasets. Using this metric, the difference for sensor bigrams is 0.0002 ± 0.0005, and for activity bigrams is 0.0217 ± 0.0019. While the deviations are small, we recognize that small perturbations have cascading effects in time-series data.

Second, we employ the two-sample Kolmogorov-Smirnov nonparametric test (KS) to determine the equality of the real and synthetic data distributions. This metric could be replaced or supplemented in future work with other distribution comparisons such as the Jensen-Shannon distance or Kullback-Leibler distance.

Using this metric, the mean KS distance between synthetic data and hold-out real data is 0.080 with a standard deviation of 0.021. While the distance is small, the values are larger than the average KS distance between multiple days from the same home which is 0.070 with a standard deviation of 0.022. The result indicates that SHGAN would benefit from continued improvement, particularly to address the imbalanced distribution challenge. On the other hand, the average p value for the KS test comparing real and synthetic data is 0.876 with a standard deviation of 0.029. Because $p > .05$, we cannot reject the null hypothesis that real and synthetic data originate from the same distribution.

Third, we utilize a "train on synthetic, test on real" (TSTR) metric in which sensor labels and values are used to predict the activity label for a window of data.

We note that all these quantitative metrics can be used not only to validate SHGAN but also to quantify the difference between training and test data for an AI system and thus the novelty of a particular task. Training and testing on synthetic data (TSTS) yields a predictive accuracy of 0.83 for 14 activities. In comparison, training on synthetic data and testing on real yields a predictive accuracy of 0.80 for the same activities. The p value using a paired t-test is $> .05$.

Conclusions

This paper demonstrates that an adversarial approach can be designed to generate realistic smart home sensor data. Our proposed algorithm, SHGAN, is adapted from a conditional Wasserstein GAN to handle the unique challenges of non-homogeneous feature types and readings that arrive at irregular time increments. Metrics including bigram probability comparison, KS distance, and TSTR validate the realism of the generated data and provide a mechanism to evaluate the novelty of a smart home situation. Future work can improve the quality of the data by more completely addressing the challenges of imbalanced data distributions and incorporating a greater variety of sensor types.

Acknowledgements

Research was sponsored by the Defense Advanced Research Projects Agency (DARPA) and the Army Research Office (ARO) and was accomplished under Cooperative Agreement Number W911NF-20-2-0004. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the DARPA or ARO, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

Arjovsky, M, S Chintala, and L Bottou. 2017. “Wasserstein Generative Adversarial Networks.” In *International Conference on Machine Learning*, Sydney, Australia, 214–23.

Arocha, J F. 2021. “Scientific Realism and the Issue of Variability in Behavior.” *Theory and Psychology* 31(3): 375–98.

Brockman, G et al. 2016. “OpenAI Gym.” *arXiv*. https://arxiv.org/pdf/1606.01540.pdf.

Cook, Diane J, Aaron Crandall, Brian Thomas, and Narayanan Krishnan. 2012. “CASAS: A Smart Home in a Box.” *IEEE Computer* 46(7): 62–69.

Cook, Diane J and Maureen Schmitter-Edgecombe. 2021. “Fusing Ambient and Mobile Sensor Features into a Behaviorome for Predicting Clinical Health Scores.” *IEEE Access* 2: 65033–43.

Dahmen, Jessamyn, and Diane J Cook. 2019. “SynSys: A Synthetic Data Generation System for Healthcare Applications.” *Sensors* 19(1181).

Dahmen, Jessamyn, Brian Thomas, Diane Cook, and Xiaobo Wang. 2017. “Activity Learning as a Foundation for Security Monitoring in Smart Homes.” *Sensors* 17(4).

Forestier, G et al. 2017. “Generating Synthetic Time Series to Augment Sparse Datasets.” In *IEEE International Conference on Data Mining*, New Orleans, LA, 865–70.

Helal, S et al. 2011. “Persim - Simulator for Human Activities in Pervasive Spaces.” In *International Conference on Intelligent Environments*, Nottingham, UK.

Jiao, Yunlong et al. 2021. “Universal Neural Vocoding with Parallel WaveNet.” In *IEEE International Conference on Acoustics, Speech and Signal Processing*, Virtual.

Minor, Bryan, Janardhan Rao Doppa, and Diane J Cook. 2017. “Learning Activity Predictors from Sensor Data: Algorithms, Evaluation, and Applications.” *IEEE Transactions on Knowledge and Data Engineering*.

Olthof, M, F Hasselman, and A Lichtwarck-Aschoff. 2020. “Complexity in Psychological Self-Ratings: Implications for Research and Practice.” *BMC Medicine* 18: 317.

Schmitter-Edgecombe, Maureen, and Diane J Cook. 2021. “Partnering a Compensatory Application with Activity-Aware Prompting to Improve Use in Individuals with Amnestic Mild Cognitive Impairment: A Randomized Controlled Pilot Clinical Trial.” *Journal of Alzheimer’s Disease*.

Schonfeld, Edgar, Bernt Schiele, and Anna Khoreva. 2020. “A U-Net Based Discriminator for Generative Adversarial Networks.” In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Virtual, 8207–16.

Virone, G, B Lefebvre, N Noury, and J Demongeot. 2003. “Modeling and Computer Simulation of Physiological Rhythms and Behaviors at Home for Data Fusion Programs in a Telecare System.” In *Workshop on Enterprise Networking and Computing in Healthcare Industry*, Santa Monica, CA, 111–17.

Yoon, Jinsung, Jarrett, Daniel, and Mihaela van der Schaar. 2019. “Time-Series Generative Adversarial Networks.” In *Conference on Neural Information Processing Systems*, Vancouver, BC.

Youngblood, G M, B Nolen, M Ross, and L B Holder. 2006. “Building Test Beds for AI with the Q3 Mode Base.” In *Artificial Intelligence and Interactive Digital Entertainment*, Marina del Rey, CA.

Zhang, Y, A Srivastava, and Diane J Cook. 2020. “Machine Learning Algorithm for Activity-Aware Demand Response Considering Energy Savings and Comfort Requirements.” *IET Smart Grid* 3(5): 730–37.