Modeling Climate Management in a Smart Home using a Scaled Testbed with Accelerated Time

Estefanía Sánchez[1], Caitlin Petro[2], Salih Safa Bacanli[3], Furkan Çimen[3], Ladislau Bölöni[4] and Damla Turgut[4]
[134]Department of Computer Science, University of Central Florida
[2]Department of Computer Science, University of North Carolina at Charlotte
[1]estefania.sanchez@knights.ucf.edu, [2]cpetro@uncc.edu, [3]{bacanli, furkan}@knights.ucf.edu
[4]{Ladislau.Boloni, Damla.Turgut}@ucf.edu

***Abstract*— A smart home with a controller that can understand and predict the interaction between the external environment and the user's behavior and preferences can provide significant energy efficiency and savings. Unfortunately, experimentation of real world homes for the development of such a controller is prohibitively expensive. In this paper we describe techniques through which such experiments can be performed on scaled testbed with an accelerated time. We illustrate how the modeling of different geographical areas can be performed by the mapping of the model's temperature and time to their real-world equivalents. We train three different machine learning models for predicting different sensor readings in the testbed, and find that the achieved predictive accuracy supports the feasibility of the development of future smart climate controllers.**

***Index Terms* Internet of things, machine learning, smart home modeling, temperature prediction.**

I. Introduction

Many smart homes currently in development may come with a variety of features: a remote security system, controlling appliances and lights, and even playing music. However, one less advertised component of smart homes is their potential financial and environmental related benefits [1]. The cost of heating and cooling one's home can add up fast: in 2012 alone, the average American household spent an average of nearly \$1000 on temperature regulation. Additionally, it has been found that 39% of global energy-related carbon emissions are from buildings with most of these emissions coming from heating, cooling, and lighting costs [2].

Modern smart thermostats have shown potential for energy and cost savings, with studies showing an energy savings rate of up to 13% for heating and up to 25% for cooling [3]. However, modern smart homes and smart thermostats do not take into account temperature fluctuations between rooms within the home nor the state of the home when regulating internal temperature. One of the biggest obstacles to improving the way we cool and heat buildings and homes is the availability of testbeds. The development of a real-size testbed is costly and time consuming. It is also difficult to collect data which is applicable to various climatic conditions without the deployment of homes across a wide area, increasing the cost of research significantly. The work described in this paper was performed on ScaledHome, a testbed that allows us to perform accelerated experiments with a scaled IoT enabled smart home. By utilizing temperature and humidity scaling, we are able to replicate a variety of environments. We are also able to control the state of the house (windows and doors) in order to manipulate temperature and air flow. Our testbed performs simulations at a rate approximately 15 times faster than real time.

Our unique testbed gives us the ability to experiment with a variety of smart thermostat implementations. We developed a predictive learning model that is able to predict internal temperatures before undesirable conditions are ever reached. We envision the realization of an intelligent agent which is able to control the state of the house in order to regulate temperature. For example, a warm room can be cooled by opening its door and receiving cooler air from other areas of the house without the use of air conditioning. With predictive modeling, all of this can be done before the room becomes warm in the first place.

The rest of this paper is organized as follows. We summarize the related work in Section II. We explain the design of our testbed in Section III and describe our simulation development strategies in Section IV. We evaluate the performance of our models in Section V and conclude in Section VI.

II. Related Work

Several research projects used machine learning methods, neural networks, and linear models to forecast a home's internal conditions.

Al-Obeidat et al. [4] used publicly available data from two sensor-rigged homes to build internal temperature prediction models that utilize ridge and lasso regression. Potočnik et al. [5] developed machine learning models for the short-term prediction of indoor temperatures. They found that nonlinear models outperform linear models in fitting and generalization. Adding features such as future outdoor temperature and expected solar radiation improved the model's accuracy.

Spencer et al. [6] used segmented linear regression to forecast future temperatures using publicly available data consisting of sensor readings indicating appliance usage, location of people, and atmospheric and weather conditions. In another publication, Spencer et al. [7] forecasted the internal temperature of a home using a linear model and found that one to two hours of sensor data was able to provide a stable accuracy of forecast horizons.

Barker et al. [8] explored the optimization of home energy consumption by collecting and analyzing data in heavily instrumented real-size test-beds. While the deployment of IoT devices was extensive, sensor data for doors and windows were not recorded and used during the project.

Zamora-Martínez et al. [9] studied the performance of difference covariate combinations in the prediction of indoor temperatures. In another publication by Zamora-Martínez et al. [10], on-line learning techniques was applied to explore the viability of predictive systems deployed in an unknown environment.

Lee et al. [11] presented a virtual system that allows for the simulation of real-world activity in a home by utilization of an autonomous agent generator. Machine learning is applied to the data generated from virtual sensors collecting user location and internal temperature. Their work, which demonstrates a strong relationship between in-home human behavior and air quality, reveals that the most impacted feature by human activity is temperature.

Marufuzzaman et al. [12] used decision tree-based machine learning algorithms to predict the activities of an inhabitant in a smart home with high accuracy. Magalhães et al. [13] developed a model that is able to capture the relationship between heating energy use and indoor temperatures at different levels of occupant behavior using artificial neural networks. Lin et al. [14] used machine learning to quantify the correlation between smart home features and chemical measurements of air quality.

Cvitić et al. [15] used logic regression and supervised machine learning to develop a model able to classify IoT devices based on traffic flow features.

Efficient temperature regulation has been explored in other scale modeled systems as well. Nada et al. [16] studied the effects of three air distribution systems on a scaled data center to determine the efficiency of the model to simulate actual data centers. Different isolation techniques led to the reduction of temperatures within different areas of the data center. Okulska et al. [17] explored the use of graph theory to find the optimal path of airflow in a home to regulate temperature in an energy efficient manner.

Our work extends previous research using earlier implementations of the ScaledHome testbed. The earliest publication on this project by Ling et al. [18] explores the implementation of fully connected and LSTM neural networks on publicly available real-world data sets and on data collected in the ScaledHome-1 prototype. Burns et al. [19] describes the ScaledHome testbed used in our experiments and further investigated the effectiveness of LSTM models on temperature and humidity prediction within the home. This study utilized changing the state of climate-control appliances at regular intervals and the state of windows and doors at random. Mendula et al. [20] implemented a management system that allowed for the remote performance of experiments on the Scaled-Home. Overall, we are expanding on these previous iterations by using the testbed as a basis to run our daily temperature simulations on. Furthermore, we are testing the same model's effectiveness in predicting temperature in comparison to other machine learning algorithms.

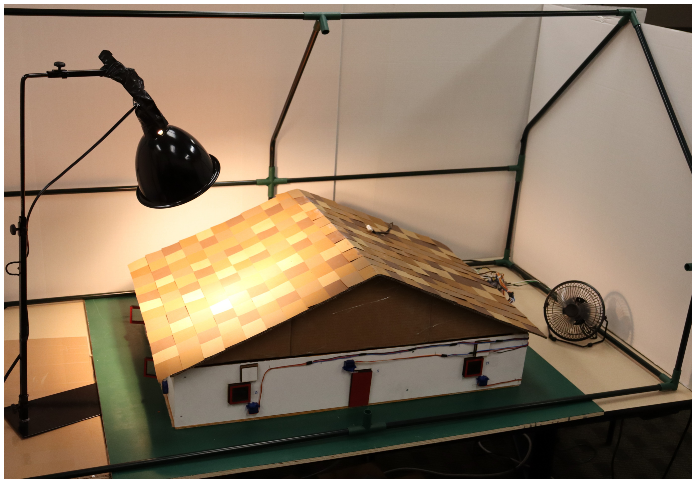

Fig. 1. External view of the ScaledHome testbed

III. The ScaledHome testbed

The ScaledHome testbed models the architecture of a small American suburban home. It contains six rooms: two bedrooms, one bathroom, a living room, a dining room, and a kitchen. The model was built using plywood for the walls and floor, wooden posts for supporting beams, and a cardboard and paper-based sloped roof. We have found that typical homes in the US with layouts comparable to our ScaledHome average between an area of 800 - 1200 sq. ft. Our ScaledHome itself is 2ft x 3ft and corresponds to a 26ft x 39ft house with an area of 1014 sq ft. This means that our ScaledHome has a length and width approximately 13 times shorter and an area 169 times smaller than a real life home.

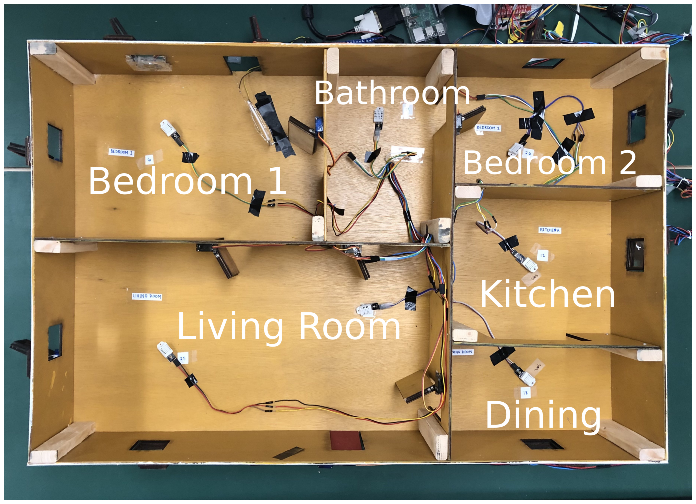

Fig. 2. Internal view of the ScaledHome testbed

The testbed operates inside an environmental enclosure, with infrared lamps and fans used to model weather patterns, as well as internal heating and air conditioning. There are a total of eight windows and seven doors in the ScaledHome, with two doors leading to the front and back of the house. Each room is separated by a door, except the dining room and the kitchen, which are divided by a partially opened wall. Each room, excluding the bathroom, contains two windows on perpendicular walls. The dining and kitchen area also has a total of two windows, with each room having a single window on perpendicular walls.

Temperature and humidity in the rooms of the testbed were measured using DHT-22 sensors. These sensors provide an accuracy of ±2% with a humidity range of 0 to 100%, and an accuracy of ±0.5°C with a temperature range of -40 to 80°C [21].

In the house, there are a total of eight sensors which collect temperature and humidity data: one in each room, and two in the living room due to its large size. Each sensor was placed in the center of the room, excluding the living room in which each sensor was placed on opposite sides of the room. Additionally, we attached fifteen Raspberry Pi micro servo motors, eight to the windows and seven to the doors, as actuators for opening and closing them.

Two different Raspberry Pi 3's were employed in our ScaledHome environment. The first Raspberry Pi collected all temperature and humidity data from the DHT-22 sensors scattered throughout the home. The second Raspberry Pi controls and records the actuator states of all doors, windows, and appliances in the ScaledHome Environment. Because a single Raspberry Pi is not able to power all the motors and appliances, we used a Pi HAT module to add an external power source that allowed the control of all fifteen motors with a single Raspberry Pi.

IV. Experimental setup

A. Choosing Locations

One of the benefits to using a testbed with an artificial weather enclosure is that it allows us to model the home as if situated in a variety of geographical areas. For modeling, we chose five locations from the United States with different climates: (1) Denver, Colorado (2) Detroid, Michigan (3) Las Vegas, Nevada, (4) Jacksonville, Florida and (5) Charlotte, North Carolina (see Figure 3). For each location, modeled a representative day for the specific climate.

In order to simulate different climates in our ScaledHome, we first had to determine what our limitations were in regards to temperature. To do this we ran each of our appliances individually to find its maximum and minimum stabilization points as well as how long it takes to reach those points. We mapped the range of temperatures feasible in the ScaledHome to the range of temperatures possible on the specific geographic location with a location specific scaling formula:

$$T^{SH} = \frac{T_{target} - T_{min}}{T_{max} - T_{min}} \left(T^{SM}_{max} - T^{SH}_{min}\right) + T^{SH}_{min} \tag{1}$$

To maintain the realism of the house, we devised three different kinds of simulations, each modeling the movements of one, two, and three people, respectively, throughout the course of a 15 hour day. There are three kinds of commands in the simulation files: toggling appliances, toggling motors, and setting a wait time. The wait time tells the actuator controller how long the house should remain in its current state after the commands are run. These simulations have slight variations based on the city's climate. For example, people who live in a moderate climate are more likely to have their windows open during the day; this is reflected in our simulations.

Each simulation is broken up into three phases: a morning phase of 1 hour and 30 minutes of real time, a midday phase of 9 hours and 30 minutes of real time, and an evening phase of 4 hours of real time. All the simulations have four or five key times depending on when the temperature peaks in a given city. The first two key times are at 7:00 AM and 8:30 AM which mark the start and end of the morning phase. The next key time is the hottest time of the day. Most of the cities have their peak during the midday phase however, Charlotte's peak time coincides with the start of the evening phase. So, Charlotte only has four key times, while the other cities have five. The final two key times are the start and end of the evening phase at 6:00 PM and 10:00 PM. The temperature scaling described above was used to scale the temperatures at these key times for each city. Table I shows the temperature mapping for Charlotte at key times of the day.

Using the mapped temperatures and the appliance stabilization data we collected, we were able to determine how long each phase took for each city. We did this by looking at how long it took the lamp to heat or cool from the temperature at the start of a phase to the end of a phase. Once we had the times each phase took in each city, we were able to change the times of the actions in the simulation files to fit accordingly.

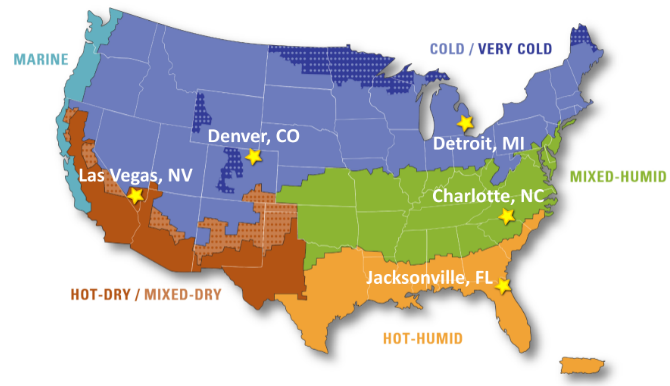

Fig. 3. Climate map with chosen locations marked [22] .

TABLE I
Charlotte, NC scaled temperatures and humidities

Time of Day	Target Temp (in C)	Scaled Temp (in C)	Target Humidity	Scaled Humidity
7:00 AM	11.7°	22.7°	48%	47.1%
8:30 AM	14.4°	23.6°	44%	46.3%
6:00 PM	23.9°	26.7°	37%	44.7%
10:00 PM	18.3°	24.9°	52%	48.0%

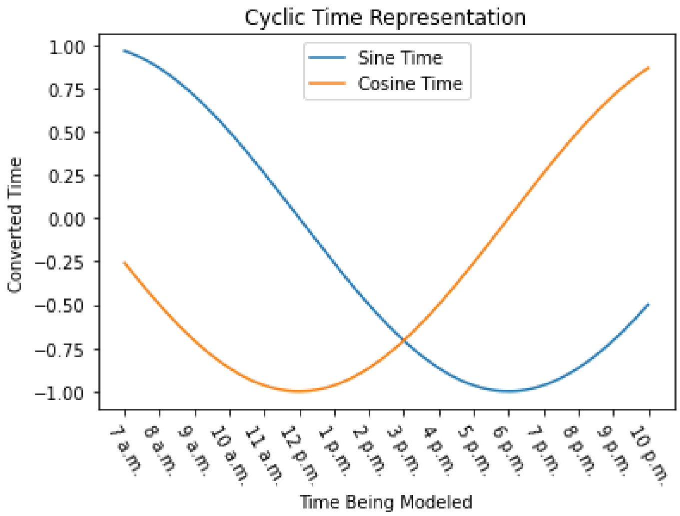

Fig. 4. Cyclic representation of time used as an input variable.

Altogether, we have 15 baseline simulations ranging between 30 to 40 minutes long. Some simulations were repeated to ensure that target temperatures were reached, so we have data from 20 simulation runs in total. This gives us roughly 5,500 data points to train and test our machine learning algorithms on.

B. Time as an Input

Because sensor data and actuator state data are collected separately, we must first combine the two according to their time stamps before we can analyze them. This presented an issue for us when it came time to put our data in our chosen machine learning algorithms. In addition, we ran simulations over the course of many days and hours, so the timestamps on our data is not consistent from simulation to simulation. However, all the simulations represent the same 15 hour day. In order to make time a consistent and valuable input for the machine learning algorithms, we converted time to a cyclic variable represented by the sine and cosine functions. We did this by first converting the real time data we collected to seconds starting from zero and going to the end of the simulation. After scaling the seconds between zero and one, we converted each time into a sine and cosine variable. The combination of the sine and cosine value maps to a specific time in a 24-hour period. Because our simulations model a day that starts at 7:00 A.M., the starting sine value is 1.00 and the starting cosine value is -0.25. The time mapping can be seen in Figure 4. We theorized that including this cyclic time variable would drastically improve the algorithms' performances since daily temperature patterns are also cyclic in nature.

C. Prediction Algorithms

To generate predictions from our data we decided to compare the performance of three algorithms: long short-term memory (LSTM), k-nearest neighbors (KNN) and random forest. In previous iterations of this project, Mendula et al. [20], Burns et al. [19] and Ling et al. [18], LSTM was found to be an effective algorithm for generating predictions. Mendula et al. [20] also found KNN to work well particularly for large datasets like the one we have been able to collect. Random forest was chosen because it is a popular machine learning algorithm that is known to work well on larger datasets. For our initial analysis of our data we used a basic LSTM setup with the hyperparameters seen in Table II. For KNN and Random Forest we used grid search to find the best parameters for our data set before training our models. We found that using five neighbors and distance as the weight function worked best for KNN and using 150 estimators with a maximum depth of five and using mean squared error to determine the quality of the split worked best for Random Forest.

TABLE II
LSTM HYPERPARAMETERS

Hyperparameter	Value
number of features	36
batch size	64
time steps	10
shift steps	15
learning rate	0.01
time steps	10
epochs	50
size of dense layer	16
optimizer	Adam

All three algorithms are predicting 15 data points in the future. This is one to two minutes in the future in our simulations which corresponds to approximately an hour in real time. This range would give an intelligent agent operating a house based on these predictions ample time to take temperature regulation measures.

We set aside 15 of our 20 simulations to be used as training data. This contained approximately 4,000 data points. The remaining five simulations were reserved for testing our algorithms. This set contained approximately 1,500 data points, giving us roughly a 70% - 30% split in our training and testing sets. In addition, we trained and tested each algorithm twice, once including a cyclic sinusoidal time variable as an input and once without. The code for our experimentation and data analysis, along with the simulation files, can be found at `https://github.com/nia-00/UCF_REU_SmartHome_2021`.

V. PERFORMANCE EVALUATION

In this section, we present the results of our experiments on the prediction of the internal and external temperatures of the ScaledHome. We compared the performance of three machine learning models using the metrics of root mean squared error (RMSE), mean absolute error (MAE) and the R^2 score. The root mean squared error denotes the standard deviation of the predictions. The mean absolute error shows, on average, how big of an error we can expect in our predictions. Finally, the R^2 score indicates how closely fit the predictions are to the regression line. The results are shown in Table III.

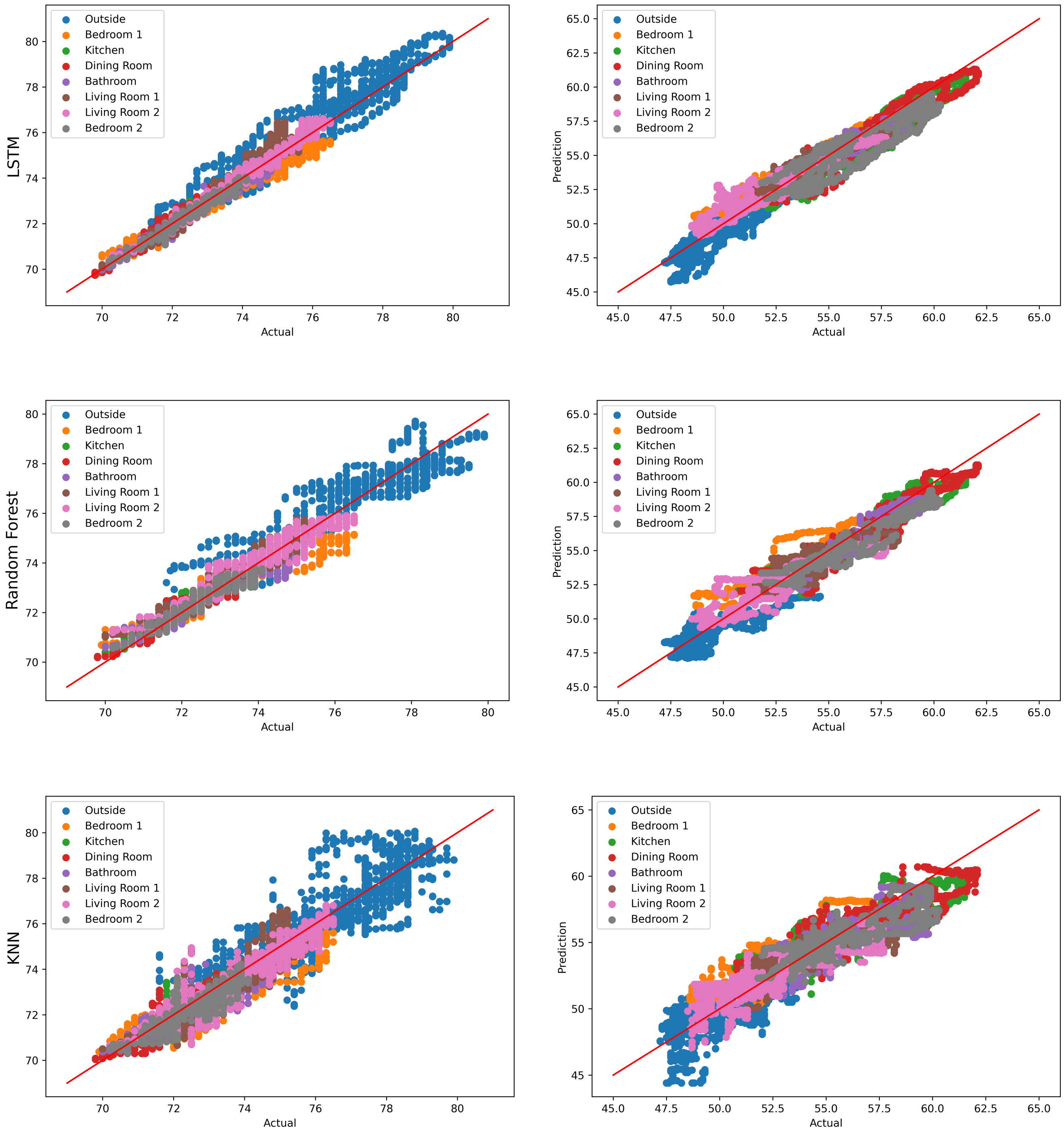

Fig. 5. The prediction accuracy for temperature (left) and humidity (right) using (top) LSTM, (middle) Random Forest and (bottom) K-nearest neighbor based prediction

A. *Algorithm Accuracy*

In Figure 5 we present the prediction accuracy for all three algorithms. The red line on the graphs represent completely accurate predictions. The closer the points on the graph are to the red line, the more accurate the prediction is. Each color represents a different sensor of the ScaledHome. We are predicting 15 time steps in the future, which is approximately 3 minutes in simulated time or one hour in real time. Though we only focused on matching the simulation temperatures to their scaled real-world equivalent in our experimentation, the

TABLE III
ALGORITHM PERFORMANCE EVALUATION

Performance Metric	LSTM	KNN	Random Forest
RMSE	0.64	1.01	0.75
MAE	0.44	0.73	0.54
R^2 Score	0.89	0.69	0.84

sensors in the ScaledHome environment collect temperature and humidity data simultaneously. Therefore we used both temperature and humidity readings in our datasets.

Table III and Figure 5 show that all models were able to generate accurate predictions. This shows that, with some tweaking, all three algorithms are suitable for further development to an intelligent, temperature regulating agent. That being said, LSTM was the best performing algorithm of the three.

Out of all the sensors, the outside sensor had the most variation and random spikes away from the regression line in each of the six models. This is likely because it was the only external sensor. Due to being directly under the lamp, it had different temperature patterns to the internal sensors that were protected by the roof of the house.

VI. CONCLUSION

In this paper we used the ScaledHome platform to develop predictive models of temperatures in smart homes. Using temperature mapping and time-accelerated experiments, we had shown that machine learning models can achieve good accuracy for such a prediction under a variety of climate conditions and user behavior. We found that using a deep recurrent neural network (LSTM) outperforms approaches such as K-nearest neighbor and Random Forest. Future work will broaden the range of modeled climate conditions and add humidity to the predicted values.

ACKNOWLEDGEMENT

The support for this work was provided by the National Science Foundation REU program under Award No. 1852002.

REFERENCES

[1] Y. Strengers, L. Nicholls, T. Owen, and S. Tirado-Herrero, “Smart home control devices. summary and assessment of energy and lifestyle marketing claims,” Centre for Urban Research (CUR) - RMIT University, Tech. Rep., April 2017.

[2] W. G. B. Council, “Air quality in the built environment,” https://www.worldgbc.org/clean-air-buildings, 2019.

[3] R. Ford, M. Pritoni, A. Sanguinetti, and B. Karlin, “Categories and functionality of smart home technology for energy management,” *Building and Environment*, vol. 123, 07 2017.

[4] F. Al-Obeidat, B. Spencer, and O. Alfandi, “Consistently accurate forecasts of temperature within buildings from sensor data using ridge and lasso regression,” *Future Generation Computer Systems*, vol. 110, pp. 382–392, September 2020.

[5] P. Potočnik, B. Vidrih, A. Kitanovski, and E. Govekar, “Neural network, arx, and extreme learning machine models for the short-term prediction of temperature in buildings,” *Building Simulation*, vol. 12, pp. 1077–1093, April 2019.

[6] B. Spencer, O. Alfandi, and F. Al-Obeidat, “Forecasting temperature in a smart home with segmented linear regression,” in *Procedia Computer Science*, vol. 155, August 2019, pp. 511–518.

[7] B. Spencer and F. Al-Obeidat, “Temperature forecasts with stable accuracy in a smart home,” *Procedia Computer Science*, vol. 83, pp. 726–733, December 2016.

[8] S. Barker, A. Mishra, D. Irwin, E. Cecchet, P. Shenoy, and J. Albrecht, “Smart*: An open data set and tools for enabling research in sustainable homes,” *Proc. SustKDD.*, 01 2012.

[9] F. Zamora-Martinínez, P. R. Guallart, P. Botella-Rocamora, and J. P. Albiach, “Towards energy efficiency: Forecasting indoor temperature via multivariate analysis,” *Energies*, vol. 6, pp. 4639–4659, October 2013.

[10] ——, “On-line learning of indoor temperature forecasting models towards energy efficiency,” *Energy Build*, vol. 83, pp. 162–172, 2014.

[11] W. Lee, S. Cho, P. Chu, H. Vu, S. Helal, W. Song, Y.-S. Jeong, and K. Cho, “Automatic agent generation for iot-based smart house simulator,” *Neurocomputing*, vol. 209, 06 2016.

[12] M. Marufuzzaman, T. Tumbraegel, L. F. Rahman, and L. M. Sidek, “A machine learning approach to predict the activity of smart home inhabitant,” *Journal of Ambient Intelligence and Smart Environments*, vol. 13, no. 4, p. 271–283, 2021.

[13] S. M. Magalhães, V. M. Leal, and I. M. Horta, “Modelling the relationship between heating energy use and indoor temperatures in residential buildings through artificial neural networks considering occupant behavior,” *Energy and Buildings*, vol. 151, pp. 332–343, September 2017.

[14] B. Lin, Y. Huangfu, N. Lima, B. Jobson, M. Kirk, P. O’Keeffe, S. N. Pressley, V. Walden, B. Lamb, and D. J. Cook, “Analyzing the relationship between human behavior and indoor air quality,” *Journal of Sensor and Actuator Networks*, vol. 6, August 2017.

[15] I. Cvitíc, D. Perakovíc, M. Periša, and B. Gupta, “Ensemble machine learning approach for classification of iot devices in smart home,” *International Journal of Machine Learning and Cybernetics*, vol. 12, pp. 3179–3202, 2021.

[16] S. Nada, K. E. Elfeky, and A. Attia, “Experimental investigations of air conditioning solutions in high power density data centers using a scaled physical model,” *International Journal of Refrigeration*, pp. 87–99, November 2015.

[17] I. Okulska and M. Ławryńczuk, “Make a difference, open the door: The energy-efficient multi-layer thermal comfort control system based on a graph airflow model with doors and windows,” *Information Sciences*, vol. 579, pp. 553–573, 2021.

[18] J. Ling, S. Zehtabian, S. S. Bacanlı, L. Bölöni, and D. Turgut, “Predicting the temperature dynamics of scaled model and real-world iot-enabled smart homes,” in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019.

[19] T. J. Burns, G. Fichthorn, J. Ling, S. Zehtabian, S. Bacanli, L. Bölöni, and D. Turgut, “Exploring the predictability of temperatures in a scaled model of a smarthome,” *Sensors*, vol. 21, no. 18, September 2021.

[20] M. Mendula, S. Khodadadeh, S. S. Bacanlı, S. Zehtabian, H. U. Sheikh, L. Bölöni, D. Turgut, and P. Bellavista, “Interaction and behaviour evaluation for smart homes: Data collection and analytics in the scaledhome project,” in *Proc. of the 23rd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, November 2020, p. 225=233.

[21] L. Aosong (Guangzhou) Electronics Co., “Digital-output relative humidity and temperature sensor/module DHT22.” [Online]. Available: http://www.aosong.com

[22] Department of Energy. Building America Climate-Specific Guidance Image Map. [Online]. Available: https://www.energy.gov/eere/buildings/maps/building-america-climate-specific-guidance-image-map