Cooking Event Detection from Temporal Thermal Condition of Residential Home

Naima Khan*, Nirmalya Roy*
*Department of Information Systems, University of Maryland, Baltimore County
nkhan4@umbc.edu, nroy@umbc.edu

***Abstract*—Contact-free activity detection is being used in several domains i.e., healthcare, cyber physical systems for its non-intrusive and flexible characteristics for end users. Thermal condition of residential homes are affected by both the outdoor weather conditions and the inside human activities. The activity of cooking affects the thermal comfort of residents inside the home and incurs a significant amount of electricity consumption in commercial kitchens. Though camera and body sensor based frameworks are proposed in the existing literature to detect cooking activities, contact free activity inference is necessary to non-intrusively assess the daily thermal comfort, monitor building envelope, electricity usage. In this work, we collected thermal condition of an apartment from surface temperature sensors and cooking activity of residents were recorded by visual observations. We avoided other activities i.e., opening doors or windows which can affect the thermal condition of residential home during cooking. We proposed a framework for cooking activity detection using recurrent neural network from surface temperature and humidity signals of residential homes. Our proposed algorithm achieves approximately 93% accuracy in cooking event detection from ambient building thermal condition.**

***Index Terms*—Contact-free activity, event, cooking, building data, thermal condition**

I. Introduction

With the emergence of lot of IoT devices, smart environments focus on detecting, monitoring and identifying human activities to improve their living condition. Different sensors, actuators, and analytic techniques of smart environments process available surrounding data to infer contextual information. Along with the development of IoT devices with smart services, we have been able to infer different information of inhabitants in building i.e., occupancy [1]–[3], thermal comfort preference [4], energy consumption [5] etc. In this work, we introduced cooking activity inference from ambient building envelope monitoring factors i.e., temperature, humidity.

Analyzing the characteristics of cooking activity inside residential homes can benefit us in several ways. Cooking area is one of the major sources of moisture in buildings and it is recommended to provide adequate mechanical ventilation in this area tightly built envelopes. This activity affects the indoor air quality by burning gas or discharging ultrafine particles in the air. It impacts indoor thermal condition by changing temperature of the entire home which requires more energy for cooling purpose to control the temperature within thermal comfort. Beside of thermal condition, cooking also accelerates the decay in indoor surfaces of building envelope over time. It causes temperature fluctuation on the building surface which gradually generates concrete deformation due to expansion or contraction of surface material. Moreover, the amount of energy consumption surpluses in commercial kitchens along with more frequent occurrence of temperature fluctuation on the building surfaces [6]. However, quantitative analysis of the impact of cooking on energy consumption and building envelope are scarce in existing literature. In this work, we attempted to monitor how much temperature fluctuation occurs due to cooking which can help us in energy saving plan and durability study of building envelope.

A 2016 survey conducted with 502 online respondents in USA reported 50% of them cook between three and six days a week [7]. The report mentions retirees and stay-home people are more likely to cook each day than full-time employees. In residential apartment homes people usually cook for $20-30$ minutes which is on average 1% for each day. Therefore, we considered cooking activity as rare events for residential homes in this work. We used temperature and humidity of different surfaces in the apartment home to detect cooking activity. Usually, a rare-event problem deals with an unbalanced dataset which contains fewer positively labeled samples than negative. We collected cooking activity information for around 60-days in two working people home.

Detecting cooking events at apartment homes is challenging. Simple threshold based algorithms cannot differentiate the temperature rise due to outside weather. Indoor thermal condition changes due to cooking activity should be separated from weather and other human activities i.e., opening windows, doors. Besides, temperature change at residential homes due to cooking are almost negligible for short cooking duration. Our cooking dataset contains only 3% data with positive or 'cooking' labels. We avoided other shallow learning algorithm (i.e., one-class SVM, decision tree) as they were not able to capture the temporal features for cooking inference from building thermal condition data. However, the small number of positively labeled samples prohibits deep learning application as well. Therefore, we approached to calculate reconstruction error of temporal sequences with recurrent neural network based long short term memory (LSTM) autoencoder. We used reconstruction error to set adjustable threshold for cooking data points.

In this paper, we proposed supervised cooking event detection method from multi-variate temporal thermal condition of residential homes. We demonstrated temperature changes in examined places for cooking activity. We determined the effective thermal variable combination to infer cooking event at residential homes and also observed the temperature variation during cooking with varying number of stoves and oven.

The rest of the paper is organized as follows: section II presents some previous studies on cooking activity recognition and event detection, section III presents the idea of overall framework, section IV describes our framework in detail, section V shows the results for cooking event detection, section VI concludes the paper.

II. Related Works

In most of the previous literature for cooking activity analysis, used temperature data from stove use monitors (SUMs) [8], [9]. Ruiz-Mercado et al. [8] deployed SUMs on chimney cookstove and traditional open-cookfire for counting the meals and daily use of the stoves using a peak selection algorithm based on instantaneous derivatives and statistical behavior of ambient temperature. Another study on the number of meals and duration of cooking was reported in [9] by comparing the survey data from cell phone and sensor data from cookstoves. They reported the average cooking duration is 1.2 hour approximately twice a day in Sudan. Beside of these studies, cooking activity has been recognized from video data using deep learning [10], [11]. Rohrbach, Marcus et al. [10] presented a dataset with 65 cooking activities which are recognized by distinguishing movements of human body from high resolution videos. The other study used 2D, 3D convolutional neural network to recognize eight cooking activities from egocentric video data [11]. A distributed framework based on state space model for detecting changes in indoor air quality, such as accidental or malicious airborne contaminant release in the building interior was proposed in [12]. They considered these incidents as events and modeled event sources i.e., doors, windows etc. as a fault in the process. This is an application of advanced fault diagnosis tools to the problem of contaminant event monitoring in intelligent buildings. Besides, in pulp and paper manufacturing industry, LSTM autoencoder was used to detect the rare event of paper breaking [13]. For anomaly detection in multivariate time series LSTM encoder-decoder model is also popular [14]. However, cooking event detection from ambient thermal condition introduces new challenges of detecting precisely distinguishable non-frequent events in residential homes.

III. Overall Framework

We design our framework for detecting cooking events occurred at residential apartment home. Our cooking event detecting framework consists of three modules, i) Data Processing, ii) Reconstruction iii) Adaptive Threshold Detection. We showed our overall process in figure 1. In data processing step, we excluded all instances with any missing value of the sensor data variables. Then we upsampled our data for 1-minute granularity and applied a median filter to raw temperature, humidity data from sensors which will eliminate extreme outlier values from the data. From temperature series, we computed

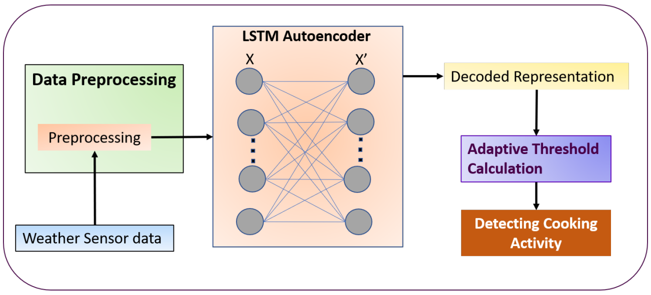


Fig. 1: Overall process

temperature change between consecutive time stamps which refers to the temperature increment or decrement rate. We prepared different combinations of data variables such as by separately considering temperature, humidity or temperature change, or by considering all these variables from all places. Then we divided our data with selected variables into three sets i.e., training, validation and testing set. In reconstruction step, we separated instances labeled with non-cooking from training set. We train our lstm-autoencoder model with only non-cooking labeled instances of the training set. Using this trained lstm-autoencoder model, we reconstructed validation set and computed reconstruction errors. In the adaptive threshold detection step, we used reconstruction errors to compute a deviation score for each of the instances in the validation set. We calculated threshold value by maximizing the F1-score with higher priority for recall. Finally, we reconstruct our testing set and computed reconstruction errors. We assign cooking labels to the instances which shows higher deviation score than the threshold values. We evaluated our framework with different combination of sensor data variables to detect cooking events occurred at apartment home. It appears that considering all data variables (i.e., temperature, humidity, temperature change rate) from all places provides better detection of cooking activities at the apartment home.

IV. Event Detection Methodology

In this section, we describe the detailed methodology to infer cooking activity from ambient temperature, humidity data. We use autoencoder based reconstruction technique to select threshold which helps to infer cooking activity. We consider cooking activities as events of various duration and assume it can happen multiple times a day. This refers to have a very little amount of positive labeled instances in our cooking activity dataset. Besides, we are required to differentiate temperature rise by cooking from the diurnal midday temperature rise at home as well as to extract subsequences of different lengths labeled with cooking activities.

In this work, we use ambient sensor data of temperature and humidity which is noisy and need to be processed before applying event detection algorithm. In data processing step, we discarded instances with missing values of all variables provided by the sensors. Collected raw temperature and humidity signals are upsampled with 1-minute granularity and filtered using a low-pass median filter to exclude any extreme outliers. The filtered data is then prepared in order to pass through lstm-autoencoder. We transformed two-dimentional data array into a 3D array of size: $samples \times lookback \times features$. Samples are just the number of instances, features are the number of data variables we want to consider and lookback is the number of instances we want to look back. We construct these frames with sliding window of having length of 90% overlap with previous frame. We choose deep learning over existing probabilistic and deterministic event detection algorithms for automatic temporal feature extraction and threshold selection. RNNs, in general, and LSTM (long short term memory), specifically, can extract temporal information for sequential or time series data. The effect of past events can automatically be extracted by these models. It can capture both long and short period effects of past events.

Mathematically we define our problem as follows. Let a home H has N number of interior and exterior surfaces. We select m surfaces to be examined where $m \leq N$. Consider $CH_{t_1}, CH_{t_2}, \ldots, CH_{t_m}$ and $CH_{h_1}, CH_{h_2}, \ldots, CH_{h_m}$ are the temporal sequences of temperature and humidity respectively, obtained from m surface locations of a home. Let each of these temporal sequences CH_{t_i} and CH_{h_j} $(i, j \leq m)$ contain n instances of temperature $temp_{i_1}, temp_{i_2}, \ldots, temp_{i_n}$ and humidity $humid_{j_1}, humid_{j_2}, \ldots, humid_{j_n}$ values. We derived temperature change or heating-cooling rate between two consecutive timestamps by this formula: $hcr_j = (temp_{j+1} - temp_j)/(t_{j+1} - t_j)$ Thus we obtained a temperature change series for each of the areas which are denoted by $CH_{hcr_1}, CH_{hcr_2}, \ldots, CH_{hcr_m}$. We prepared different multivariate time series using combinations of temperature, humidity and heating-cooling rate from the examined areas. However, the goal is to perform supervised event detection from these n labeled data sequences. Event detection extracts temporal subsequences of various lengths and assigns positive labels to all instances between starting and ending timestamps. In our case, we considered cooking activities from single length timestamps to higher length. From event detection algorithm we can mark starting and ending timestamps (s_c, e_c) for each of the cooking events c and assigns positive labels to all instances between s_c and e_c.

Consider our prepared temporal sequences from the selected sensor variables are $X = x_1, x_2, ..., x_L$ of length L, where each sequence $x_i \in R_m$ is an d-dimensional vector of readings for d variables starting at timestamp t_i. We train an LSTM autoencoder to reconstruct instances of normal time-series. The LSTM encoder learns a fixed length vector representation of the input time-series and the LSTM decoder uses this representation to reconstruct the time-series using the current hidden state and the value predicted at the previous timestep. The reconstruction errors are then used to obtain the likelihood of a point in a test time-series being anomalous. For each point x_i, a deviation score Dev_i of the point being anomalous is obtained. A higher deviation score indicates a higher likelihood of the point being positive labeled.

Given X, h_E^i is the hidden state of encoder at time t_i for each $i \in 1, 2, \ldots, L$, where $h_E^i \in R_k$, k is the number of LSTM units in the hidden layer of the encoder. The final state h_E^L of the encoder is used as the initial state for the decoder. During training, the decoder uses x_i as input to obtain the state h_D^{i-1}, then predict $\hat{x}_{i-1}$ corresponding to target x_{i-1}. During inference, the predicted value $\hat{x_i}$ is input to the decoder to obtain h_D^{i-1} and predict $\hat{x}_{i-1}$. The model is trained to minimize the objective $\sum_{X \in s_N} \sum_{i=1}^{L} (||x_i - \hat{x_i}||)^2$,where s_N is set of normal training sequences.

We divided X into three sets of time-series: X_{train} , X_{valid}, and X_{test}. Then we separated X_{train_c} (the instances with "cooking" labels) and X_{train_n} (the instances with "normal" labeled) from X_{train}. The set of sequences X_{train_n} is used to learn the LSTM autoencoder reconstruction model. The set X_{valid} is used as validation set while training the encoder-decoder model. The reconstruction error vector for x_i is given by $e_i = |x_i - \hat{x_i}|$. The error vectors for the points in set X_{valid} are used to estimate the mean (μ) and standard deviation (σ) of a Normal distribution $\mathcal{N}(\mu, \sigma^2)$ using Maximum Likelihood Estimation. Then, for any point x_i, the deviation score $dev_i = (e_i - \mu)/\sigma^2$. In a supervised setting, if $dev_i > \tau$, a point in a sequence can be predicted to be "cooking", otherwise "normal". The threshold τ over the likelihood values is learnt to maximize $F_\beta = (1 + \beta^2) \times P \times R/(\beta^2 P + R)$, where P is precision, R is recall, "cooking" is the positive class and "normal" is the negative class. If a window contains an cooking pattern, the entire window is labeled as "cooking". We assume $\beta > 1$ for higher recall since we are interested to find out all the data points of cooking. The parameters τ and k are chosen with maximum F_β score on the validation sequences in X_{valid}. Complete flow of the process is presented as algorithm in algorithm 1.

Algorithm 1 Cooking Event Detection

---

procedure COOKINGEVENTDETECTION (**Input:** temporal instances X, true labels y, $testsize$ **Output:** Predicted cooking Labels $\widehat{Y}$)
    Initialize all model parameters θ for LSTM AutoEncoder
    X_{train}, y_{train}, X_{valid}, y_{valid}, X_{test}, y_{test} $\leftarrow$ TrainTestSplit $(X, y, testsize)$
    $X_{train_n} \leftarrow X_{train_i}$ if $y_{train_i} == 0$
    Train Autoencoder with X_{train_n}
    $\widehat{X_{valid}} \leftarrow$ Autoencoder (X_{valid})
    $\widehat{X_{test}} \leftarrow$ Autoencoder (X_{test})
    ReconstructionError, $E \leftarrow |X_{valid}$ - $\widehat{X_{valid}}|$
    Calculate: $\mathcal{N}(\mu, \sigma^2)$
    DeviationScore $\leftarrow (E_i - \mu)/ \sigma^2$
    Threshold $\leftarrow$ Max$((1 + \beta^2) \times P \times R/(\beta^2 P + R))$
    $\widehat{Y} \leftarrow 0$ if $Score \leq Threshold$; 1 if $Score \geq Threshold$
    return $\widehat{Y}$
end procedure

V. EXPERIMENTAL RESULTS

The framework is evaluated on a cooking dataset collected from an apartment unit over 48 days. We deployed Ambi-

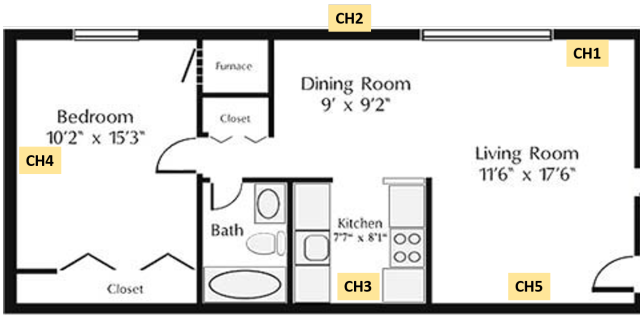


Fig. 2: Sensor placements in the apartment unit

ent weather Thermo-Hygrometer with WS-3000-X5 monitor and five wireless remote sensors to different places of the apartment. The sensors can report temperature in range from $-40°F$ to $140°F$ and humidity in range from 10% and 99%. Weather sensors were placed on the surface of different walls in the apartment. We examined five places of the home in this work. The frequency of the temperature and humidity data is $915MHz$. Temperature and humidity get updated in every 60 seconds. We saved starting and ending time of cooking events by visual observations for 60 days between October and December. We tried to avoid other thermal condition changing human activities i.e., opening windows or doors, during cooking in order to focus only on the thermal condition changing for cooking activities inside the apartment unit. Due to some skipped records of cooking and overlapping of other thermal condition changing activities, we discarded 12 days temperature and humidity data and worked with the remaining 48 days data. Figure 2 shows the orientation of the apartment unit and placement of sensors in the apartment unit.

$CH_1, CH_2, CH_3, CH_4, CH_5$ represents the sensors which provides temperature, humidity, dew point and heat index of the place according to it's placement. We took temperature and humidity from each of the sensors and calculated heating and cooling rate for each of the places we monitored. In this study thermal and moisture condition of each place CH_i is presented by temperature $Temp_i$, humidity $Humid_i$ and heating-cooling rate HCR_i.

The collected temperature and humidity data from weather sensors are noisy and have missing values. For any missing values of temperature, humidity, dew point and heat index, we discarded the entire instance. Figure 3 shows the temperature and humidity data for a day obtained from the apartment unit. The figure shows the outside surface CH_2 has the lowest temperature and the highest humidity for the day. However, temperature series from all other indoor surfaces show two peaks throughout the day and CH_3 has slightly higher temperature than the other channels during this peak. Humidity for indoor surfaces are slightly higher for CH_3 and CH_4. As the apartment unit is centrally air-conditioned, temperature stays almost same all the time except extreme weather or any other human activity which affects the indoor temperature.

We showed temperature and heating-cooling rate for each

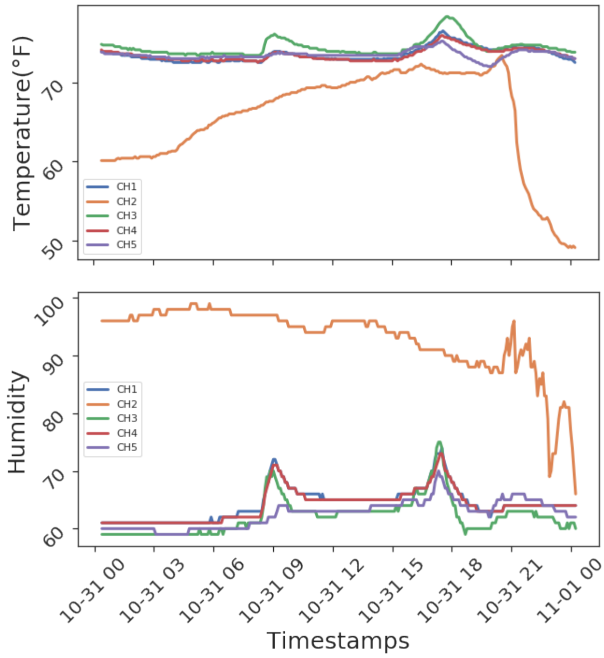
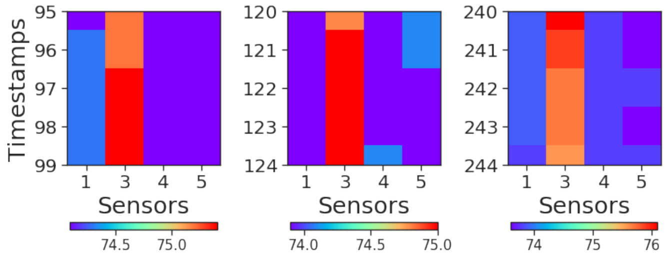


Fig. 3: Temperature and humidity data from sensors

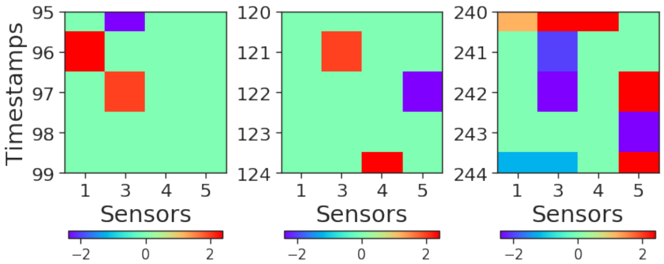
of the places inside the apartment unit during cooking in figure 4. Each subplot in the image shows temperature or heating cooling rate for consecutive five time stamps. Each column represents temperature or heating cooling rate for each of the channels over consecutive time stamps and each row represents temperature or heating cooling rate in all channels for a specific timestamp. Figure 4(a) shows channel CH_3 has always higher temperature (greater than $75°F$) than other channels. Corresponding heating and cooling rate for each channel at the same timestamps is showed in figure 4(b). In all cases, during five timestamps at least one of the channel shows increment of heating-cooling rate (i.e., highlighted with red) during cooking.

We labeled all temperature sensor instances within cooking duration with 1 and other instances with 0 to refer normal condition. We had around 13300 instances which was randomly divided into training and testing set (25%). We used 20% instances as validation set from training set. According to manually labeled dataset, the total number of cooking event in temperature time series is 32 within 48 days. We applied our cooking event detection algorithm on different variable sets with various combination of temperature, humidity and heating-cooling rate from one individual channel or all channels. However, we presented evaluation results from the combinations e.g. temperature series from all channels, humidity series from all channels, temperature and humidity from channel CH_3 and temperature, humidity, heating-cooling rate from all channels.

Our lstm-autoencoder consists of two layers for encoding and two layers for decoding. We trained our autoencoder model for 1000 epochs with batch size 64. We used different number of variables (i.e., 2, 5, 10, 15 etc.) based on the

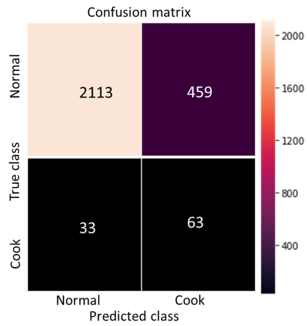


(a) Temperature

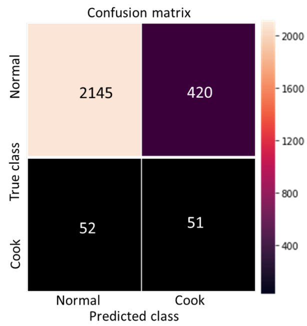


(b) Heating-Cooling rate

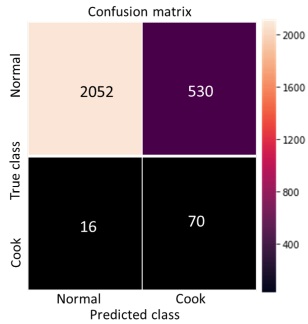
Fig. 4: Visualization of temperature, heating and cooling rate from Sensors



(a) Temperature



(b) Humidity



(c) CH_3 temperature and humdity

(d) All

Fig. 5: Confusion matrices

combinations selected. In this study, we also explored different number of timesteps to lookback for cooking event detection. Empirically we found that 5 timesteps provides the best evaluation metrics. We evaluated our framework in two ways, i.e., one way counts the total number of temperature time series instances are correctly detected as cooking and the another way counts the total number of correctly detected cooking events. We considered true positive (TP) instances or events, false positive (FP) instances or events and false negative (FN) instances or events for calculating different evaluation metrics i.e., accuracy, precision, recall.

Table I shows the evaluation metrics of detecting "cooking" labeled instances. There were 2668 instances in test set where 103 instances were labeled as 'cooking'. The table shows

TABLE I: Evaluation of detecting 'cooking' labeled instance

Dataset	Detected	Accuracy	Precision	Recall
$Temp_1, Temp_2, Temp_3, Temp_4, Temp_5$	63	0.82	0.12	0.66
$Humid_1, Humid_2, Humid_3, Humid_4, Humid_5$	51	0.81	0.11	0.50
$Temp_3, Humid_3$	70	0.80	0.12	0.81
All	93	0.95	0.44	0.90

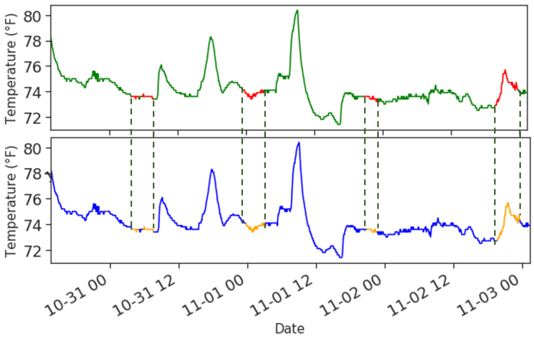
different combination of sensor variables causes difference in the evaluation metric for cooking event detection. The combination with temperature, humidity and heating-cooling rate from all channels work best for accurately detecting cooking events. As we upsampled the temperature and humidity data with 1-minute granularity, we have been able to detect whether any cooking activity is occurred or not for every 1 minute. Table II shows the evaluation metrics of detecting cooking events. This table shows the results for how many events were correctly detected with all timestamps including starting and ending timestamps. We considered one instance tolerance for detecting starting and ending instance for each of the cooking events. Here we counted false positive if any consecutive timestamps are falsely detected as 'cooking' and false negative if any cooking timestamps are falsely detected as 'normal'.

TABLE II: Evaluation of cooking event detection

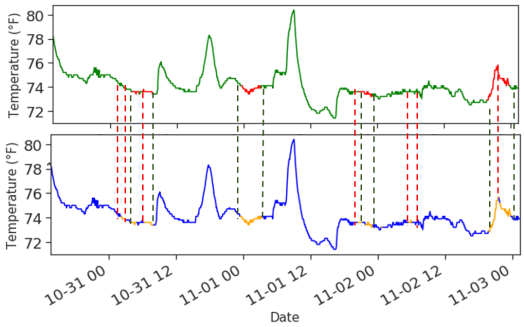
Dataset	Detected	Accuracy	Precision	Recall
$Temp_1, Temp_2, Temp_3, Temp_4, Temp_5$	26	0.81	0.24	0.75
$Humid_1, Humid_2, Humid_3, Humid_4, Humid_5$	20	0.63	0.20	0.60
$Temp_3, Humid_3$	26	0.68	0.22	0.79
All	30	0.93	0.66	0.85

Figure 5 shows TP, FP, FN, TN for each combination of data variables for detecting cooking instances. It is obvious from the tables and confusion matrices that, only temperature or humidity cannot improve the cooking event detection in our experiments. Temperature and humidity from only CH_3 does not improve the result as it increases the false positive instances significantly. It indicates only kitchen area temperature and humidity data is not sufficient to detect cooking activity in a 726 sq^2 feet apartment. For cooking activity detection, we need to consider thermal condition of the whole apartment unit. Considering all temperature, humidity as well as heating and cooling rate from all channels reduce the false negative and false positive instances.

We showed cooking event prediction compared with original



(a) All variables



(b) Temperature

Fig. 6: Prediction of cooking event

event label in figure 6 for three days. Figure 6(a) shows cooking event detection using all variables while figure 6 (b) shows cooking event detection using only temperature series from all channels. The first image shows all variables combination can find all short or long duration cooking events accurately in the timeline. The second image shows temperature combination detects some instances as cooking falsely for the same timeline. Falsely detected events are showed with dotted red colored starting and ending boundaries and correctly detected events are showed with dotted black color.

With all sensor variables from all channels, we detected starting and ending instances of cooking events correctly up to 80% without ± 1 instance tolerance. For 95% of the total cooking events, instances were correctly detected as 'cooking' at least after 2-minute of actual starting timestamps. We noticed cooking duration and number of stoves or cooking unit has significant impact on the amount of temperature change. Combination of one stove and oven increases temperature up to $5°F$ in 1 hr cooking duration while only one stove achieves the same amount of temperature increment in 1.5 hr. Temperature increment is significant for using two or more than two stoves. In one hour two stoves increases up to $8°F$. Temperature doesn't start to decrease just after stop cooking, rather it starts to decrease after quite some time. Therefore, we obtained higher number of false positive instances which leads to low precision, using only temperature or humidity from all channels while it raised significantly up to 44% using all temperature and humidity variables from all channels.

VI. Conclusion

In this paper, we presented cooking events as rare events in the thermal data of a residential home. We inferred the isolated cooking events using lstm autoencoder from ambient weather variables (i.e., temperature, humidity, heating-cooling rate). We evaluated and demonstrated our cooking event detection algorithm with different combination of weather variables. We suggested to use temperature and humidity along with the temperature change rate of whole apartment area to infer cooking activity. In future work, we plan to recognize other human activities i.e., door, window interaction which changes thermal condition of residential homes from contextual weather data.

VII. Acknowledgment

This research is partially supported by the following grants; NSF CAREER 1750936, NSF CPS 1544687, NSF CNS 1640625, ONR N00014-18-1-2462, and Alzheimer's Association AARG-17-533039.

References

[1] H. Saha, A. R. Florita, G. P. Henze, and S. Sarkar, "Occupancy sensing in buildings: A review of data analytics approaches," *Energy and Buildings*, 2019.

[2] H. Elkhoukhi, Y. NaitMalek, A. Berouine, M. Bakhouya, D. Elouadghiri, and M. Essaaidi, "Towards a real-time occupancy detection approach for smart buildings," *Procedia computer science*, vol. 134, pp. 114–120, 2018.

[3] B. Abade, D. Perez Abreu, and M. Curado, "A non-intrusive approach for indoor occupancy detection in smart environments," *Sensors*, vol. 18, no. 11, p. 3953, 2018.

[4] K. J. Lomas and R. Giridharan, "Thermal comfort standards, measured internal temperatures and thermal resilience to climate change of free-running buildings: A case-study of hospital wards," *Building and Environment*, vol. 55, pp. 57–72, 2012.

[5] S. Bansal and M. Schmidt, "Energy disaggregation methods for commercial buildings using smart meter and operational data," in *Workshops at the Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[6] S. Mudie, E. A. Essah, A. Grandison, and R. Felgate, "Electricity use in the commercial kitchen," *International Journal of Low-Carbon Technologies*, vol. 11, no. 1, pp. 66–74, 2013.

[7] "Us home cooking statistics habits." https://www.reportlinker.com/insight/americans-cooking-habits.html. Accessed: 2019-12-03.

[8] I. Ruiz-Mercado, E. Canuz, and K. R. Smith, "Temperature dataloggers as stove use monitors (sums): Field methods and signal analysis," *Biomass and Bioenergy*, vol. 47, pp. 459–468, 2012.

[9] D. L. Wilson, M. I. Adam, O. Abbas, J. Coyle, A. Kirk, J. Rosa, and A. J. Gadgil, "Comparing cookstove usage measured with sensors versus cell phone-based surveys in darfur, sudan," in *Technologies for Development*, pp. 211–221, Springer, 2015.

[10] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele, "A database for fine grained activity detection of cooking activities," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1194–1201, IEEE, 2012.

[11] S. Urabe, K. Inoue, and M. Yoshioka, "Cooking activities recognition in egocentric videos using combining 2dcnn and 3dcnn," in *Proceedings of the Joint Workshop on Multimedia for Cooking and Eating Activities and Multimedia Assisted Dietary Management*, pp. 1–8, ACM, 2018.

[12] V. Reppa, M. Michaelides, M. Christodoulou, C. Panayiotou, and M. Polycarpou, "A distributed framework for contaminant event detection and isolation in multi-zone intelligent buildings," in *2014 European Control Conference (ECC)*, pp. 1637–1642, IEEE, 2014.

[13] C. Ranjan, M. Reddy, M. Mustonen, K. Paynabar, and K. Pourak, "Dataset: rare event classification in multivariate time series," *arXiv preprint arXiv:1809.10717*, 2018.

[14] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long short term memory networks for anomaly detection in time series," in *Proceedings*, p. 89, Presses universitaires de Louvain, 2015.