Inferring Occupancy from Opportunistically Available Sensor Data

Longqi Yang*
Department of Information Science and Electronic Engineering, Zhejiang University
Hangzhou, China
Email: ylongqi@gmail.com

Kevin Ting*
Department of Electrical Engineering
University of California, Los Angeles
Los Angeles, USA
Email: kting531@ucla.edu

Mani B. Srivastava
Department of Electrical Engineering
University of California, Los Angeles
Los Angeles, USA
Email: mbs@ucla.edu

*Both authors contributed equally to this work

***Abstract*—Commercial and residential buildings are usually instrumented with meters and sensors that are deployed as part of a utility infrastructure installed by companies that provide services such as electricity, water, gas, security, phone, etc. As part of their normal operation, these service providers have direct access to information from the sensors and meters. A concern arises that the sensory information collected by the providers, although coarse-grained, can be subject to analysis that reveals private information about the users of the building. Oftentimes, multiple services are provided by the same company, in which case the potential for leakage of private information increases. Our research seeks to investigate the extent to which easily available sensory information may be used by external service providers to make occupancy-related inferences. Particularly, we focus on inferences from two different sources: motion sensors, which are installed and monitored by security companies, and smart electric meters, which are deployed by electric companies for billing and demand-response management. We explore the motion sensor scenario in a three-person single-family home and the electric meter scenario in a twelve-person university lab. Our exploration with various inference methods shows that sensory information available to service providers can enable them to make undesired occupancy related inferences, such as levels of occupancy or even the identities of current occupants, significantly better than naive prediction strategies that do not make use of sensor information.**

I. Introduction

Modern residential and commercial buildings have a multitude of sensors and meters installed in them by external service providers. In addition to being used for service delivery and billing, the data from these sensors can also give important feedback to the users and help the provider provision value-added services and recommend customized service plans. However, there is also the risk that information from these sensors can be used to make unwanted inferences about occupants and their behavior.

One specific source for potentially privacy-infringing data comes from the sensors set up by security companies. Residential security systems have grown considerably in sophistication as they are now always connected and engage in frequent bi-directional information exchange with monitoring services. In particular, sophisticated services, such as *Alarm.com*[1], have emerged that collect measurements from sensors in real-time, even when the system is not armed, and provide a variety of additional services via smartphone apps and websites. According to *Alarm.com*, over 1 million properties and 20 million individual sensors have been monitored by their security system since October 2012. While home security sensors are anonymous, when combined with publicly available information, there is potential of making non-anonymous inferences about specific residents. Another potential source for inferences about occupancy comes from electrical data available from government-mandated smart meters. According to a 2011 U.S. Energy Information Administration's report, 37,290,374 advanced metering infrastructures have been installed around the country to measure and record electricity usage and provide the corresponding data to both the utility company and customers [1]. These meters can provide power consumption data either every few seconds or every few minutes. Algorithms can then be applied to data collected from these meters to create a model that maps changes in power to changes in occupancy.

The work described here is motivated by the intuition that, by applying the right models and classifiers to data collected from specific sensors, we can infer answers to the following four occupancy questions: 1) Is a particular space occupied? 2) How many people are there in that space? 3) If that space is occupied, what are its occupants' identities? 4) Which particular subspaces do they occupy? In this paper, we seek to quantify the extent of information leakage possible from access to either motion sensors or electric meters. For the motion sensor scenario, we focus on passive infrared sensors, which are the most common sensors found in home security systems. In this context, we present results on exploring occupancy questions 2 and 3 by using data from a three-person family home. For electric meter-based inference, we focus on questions 1 and 2 while using data from a twelve-person university lab. In this paper, we make the following contributions:

- We evaluate the possibility of inferring the number and identity of occupants in a house by only using motion sensors.

[1] http://www.alarm.com/

978-1-4799-3445-4/14/$31.00 ©2014 IEEE

- We characterize the evolution of the occupancy state of a house via the learning based Conditional Hidden Markov Model and the intuitively-designed rule-based method.
- We explore the possibility of non-intrusive occupancy inference by using aggregate smart meter data to infer both binary occupancy and range of occupancy level.

As a roadmap for the rest of the paper, we first introduce the models used in the motion sensor-based context. Specifically, Conditional Hidden Markov Model, Conditional Random Fields, Hidden Markov-Support Vector Machine and a rule-based method are shown in detail in section III-A. Next, in section III-B, we explain our methodology for inferring binary and ranged occupancy in the electric meter-based scenario. Then, in sections IV, we describe the experimental setting for both of our experiments. Finally, in section V, we propose several evaluation metrics and utilize them to compare the performance of different models.

II. Related Work

There is a significant body of literature related to occupancy analysis using various types of sensor data. For example, Agarwal et al. used a finite-state machine driven by motion and door sensor events to infer whether an office room is occupied or not [2]. Motion and door sensors have also been used for activity inference, with Nazerfard et al. using Bayesian Networks to predict the occupant activity in a single-resident home densely instrumented with such sensors [3]. However, neither of those works considered the case of multiple occupants or tried to distinguish their identities.

Prior work that has sought to handle multiple occupants and infer their identity often rely on imaging, with Erickson et al. using a network of low-power embedded cameras [4]. Beyond imaging, Hnat et al. implemented Doorjamb, a system that uses custom sensors combining motion, door contact, and ultrasonic range finders to detect and identify (from height measurement) occupants moving across rooms [5]. While capable of much more sophisticated inferences and higher accuracy, Doorjamb, in its present form, is best suited for special deployments, such as in labs or as part of research studies.

In terms of basic occupancy estimation, several other systems exist, though they are considered intrusive because they rely on data from sensors that are not usually available to outside service providers [6], [7], [8]. Although Ghai et al.'s experimental context is different from ours, we still utilize parts of their approach when choosing and evaluating different machine learning algorithms on our sensor data [6]. Finally, Chen et al. and Kleiminger et al. have shown that data from a smart meter can be used to infer binary occupancy patterns, but they utilize different algorithms/features and focus on a different setting than we do [9], [10].

III. Methodology

A. Motion Sensor-based Scenario

The first opportunistic sensor data we considered comes from the motion sensors installed by security companies. In this scenario, our goal was to explore the possibility of inferring the number of residents in the house and their identities, at any given time, by using only the motion sensor signals that they trigger. Intuitively this is possible as occupants in a home have movement patterns that are distinct in space and time, for example people may sleep in different bedrooms. This opens the possibility that a sophisticated machine learning algorithm might learn to discriminate movement trajectories belonging to different occupants, thereby identifying the current occupants just from anonymous motion events. In this experiment, we incorporated data from motion sensors and the time of day as inputs into our algorithms. We worked with two different sensor placement scenarios: *rich resources*, where there is one motion sensor in each of the ten rooms of the house and *limited resources*, where there are only motion sensors installed in the bedroom, study room and foyer, the three locations most commonly monitored by modern home security systems.

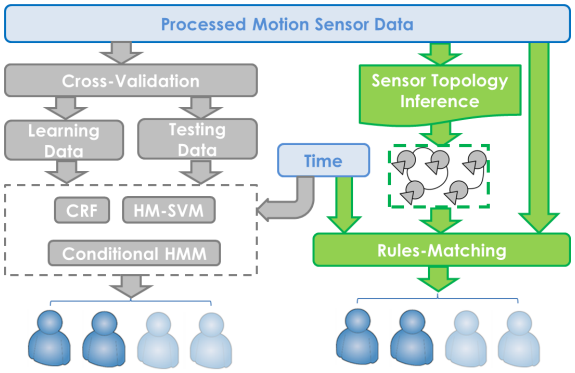


Fig. 1. Overview of the models used in motion sensor based scenario

As an overview, Fig.1 shows the two categories of methods that are proposed under this scenario: learning-based and rule-based. Although the first set of methods requires training data, we still considered it because we do not want to assume limitations on the complexity of the model that an adversarial monitoring company can access. We applied cross-validation on our data with three different models in order to infer the number of occupants and their identities in the home. The three models considered are: Conditional Hidden Markov Model, Conditional Random Fields and Hidden-Markov Support Vector Machine. For the rule-based approach, several rules are presented which do not require any training data.

1) Conditional Hidden Markov model: This model is a proposed Hidden Markov Model [11] based classifier. To avoid the large number of states that a straightforward mapping of the problem structure to a flat HMM would entail, we created a variant, which includes the hour of day as a mechanism to control the transition probability matrix of a HMM. We call this model Conditional HMM. Fig.2 shows the time evolution of the hidden state (top) and motion sensor observation vector (bottom) with the transition governed by the transition probability matrix (horizontal arrow). Specifically, at the top of Fig.2, H_i and $S_i (i = 1, 2, \cdots, T)$ represent the hour of day

and the state number for the time slot i respectively, whereas at the bottom, $M_i^{(1)}, M_i^{(2)}, \cdots, M_i^{(10)}(1, 2, \cdots, T)$ denotes the output of ten motion sensors at time slot i.

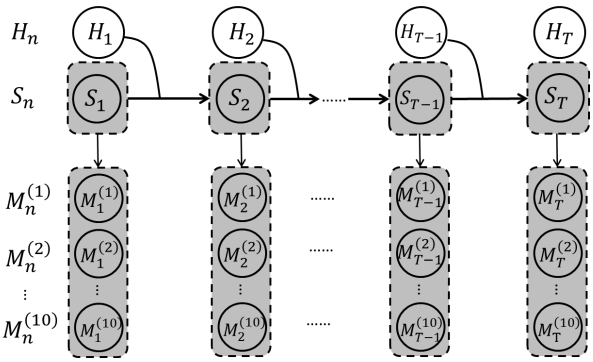


Fig. 2. Time evolution of the Conditional Hidden Markov Model

To make the model more tractable and clear, we defined a mapping between the state number and the occupancy state of the house. In our subject residence, the maximum number of the potential occupants in the house is three. Therefore, having 8 different states is sufficient to represent all possible combinations of the occupants in the house. The mapping is done in a binary-decomposition way and is shown in Fig.3.

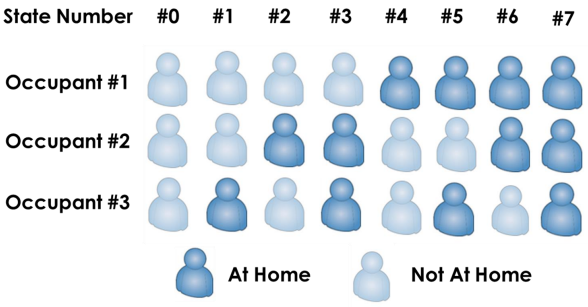


Fig. 3. Mapping between the state number and occupancy state in Conditional HMM

For the parameter-learning part of our Conditional Hidden Markov Model, the criteria of maximum joint probability is adopted. In terms of our model, the joint probability of the hidden states and the observation is derived from (1). By using this criteria, the transition and emission probabilities are learned and the Viterbi algorithm is then applied to infer the most probable sequence of hidden states along T time slots.

$$p(\mathbf{S_n}, \mathbf{M_n^{(1)}}, \cdots, \mathbf{M_n^{(10)}}) = \prod_{i=2}^{T} p(S_i|S_{i-1}, H_{i-1}) p(M_i^{(1)}, \cdots, M_i^{(10)}|S_i) \quad (1)$$

2) Conditional Random Fields: CRF [12] is a commonly used discriminative model to segment and label sequences, which is advantageous in that it does not require the two independence assumptions [13] of HMM: 1) Present state only depends on its immediate predecessor and 2) Present observation only depends on the present state; This allows CRF to adopt more complex feature sets by modeling the posterior probability instead of joint probability. The standard form of CRF can be seen in (2), where F, $\mathbf{x}$ and $\mathbf{y}$ represent the feature function, observation and hidden state respectively. After applying $Z(\mathbf{x})$ to normalize the probability value from 0 to 1, the weighted parameter λ_j can be learned under the criteria of maximum posterior probability. When making predictions under the CRF model, the output sequence which maximizes the posterior probability (2) will always be chosen. In the CRF model we propose, the definitions of state and observation are the same as the ones in Conditional HMM, and each feature function is the sum of the corresponding indicator functions along time (3).

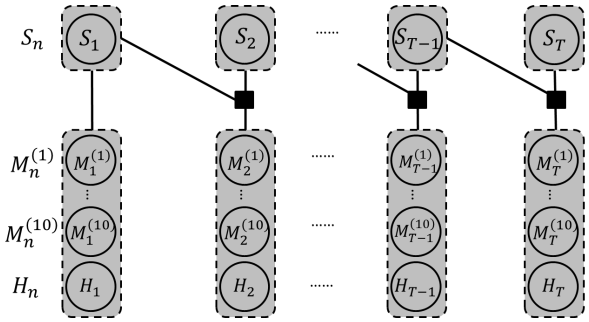


Fig. 4. The structure of Conditional Random Fields

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} exp\{\sum_j \lambda_j F_j(\mathbf{y}, \mathbf{x})\} \quad (2)$$

$$F_j(\mathbf{y}, \mathbf{x}) = \sum_{i=1}^{T} f_j(y_{i-1}, y_i, x_i, i) \quad (3)$$

Specifically, the structure of the CRF model we applied in our problem is presented in Fig.4 where the preceding state is correlated to both the present state and observations. The shaded boxes in Fig.4 are the factor nodes [13] which carry out the calculation of the indicator functions. For the implementation of the CRF model, a standard toolkit called CRF++ [14] was adopted in our problem.

3) Hidden Markov Support Vector Machine: Hidden Markov-Support Vector Machine (HM-SVM) [15] is another discriminative model that we considered. It was chosen for its capacity to combine the strengths of Support Vector Machines and Hidden Markov Models to potentially obtain better results than each of the methods could individually. The features we chose to use with HM-SVM are related to the first-order transition and zero-order emission properties of traditional HMM. To implement HM-SVM, a package called SVM-HMM [16] was used.

4) Rule-based Method: Unlike the learning based methods, the rule-based method does not need a mandatory learning process as a generic set of rules applicable to a set of households with common attributes may be formulated based on background knowledge. This makes the approach applicable

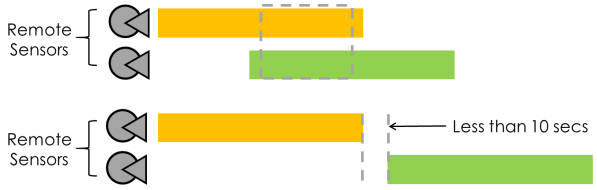


Fig. 5. **NUM** Calculation (Sensors trigged simultaneously or successively in 10 seconds)

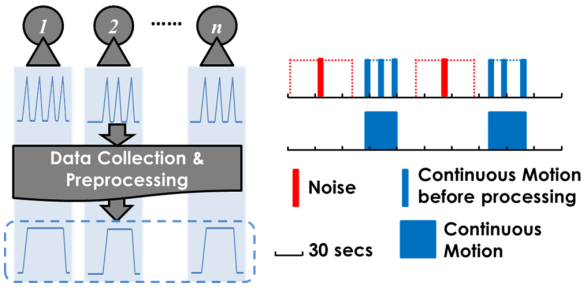


Fig. 6. Methods for the Motion Sensor Data Preprocessing

to a wide variety of households. We designed the rules on the basis of the following two principles: 1) Motion sensors do not provide any clues about occupancy at night when everyone is sleeping. 2) Such sensors embody the behavioral information of the occupants in the daytime, as they are more active then. The proposed method can be divided into four successive steps:

Step 1: Sensor Topology Inference: In this step, we tried to infer the distribution of the motion sensors in the house. The algorithm published by C. Ellis et al. [17] was adopted here. The basic idea is that if two sensors are always triggered successively in 30 seconds, then they are adjacent, otherwise they are remote.

Step 2: NUM Calculation: NUM is defined as the maximum number of remote sensors that are triggered simultaneously or successively within 10 seconds, as illustrated by Fig.5. Intuitively, the number of remote sensors that are triggered at the same time determines the minimum number of residents in the home, because a single person cannot possibly trigger multiple remote sensors at the same time.

Step 3: Number of Occupants Inference: In order to infer the number of occupants with respect to the layout of motion sensors, two rules were designated here: 1) For the daytime period (7 a.m. to 6 p.m.), the value of **NUM** is the number of occupants at each corresponding time slot. 2) For the nighttime period, everyone is present in the house.

Step 4: Identities of Occupants Inference: Finally, after inferring the number of occupants in step 3, a naive and direct rule was applied to guess the identities of the occupants: randomly choose a person when an occupancy related event occurs.

5) Data Pre-processing: The raw data from PIR sensors cannot be directly used as input to our models for the following two reasons: 1) PIR sensors are not perfect and they sometimes generate false events due to outside interference. 2) The motion sensors output discrete events even when there is continuous movement. For our system, the PIR sensor sends a sensor-managing gateway the sequence 0-1-0 whenever motion is detected. This sensor-managing gateway then pushes this data to *Xively*[2].

The point of the data preprocessing step is to solve the two problems mentioned above. The method we used is presented in Fig.6: When a motion is detected and there is no other activity 30 seconds before or after it, then it is regarded as noise and ignored. Otherwise, time slots with successive motions are turned into a block representing continuous movement.

[2]https://xively.com/

B. Electric Meter-based Scenario

The second source of opportunistic sensor data that we considered are the government-mandated smart meters installed in many homes and office buildings. The idea is that, when a person enters a room, they will cause some recognizable step in the aggregate power consumption by their use of some appliance. Our goal is to map this change in power consumption to changes in occupancy.

In this section, we focus on two specific inferences, binary occupancy (i.e. is the number of occupants zero or non-zero?) and ranged occupancy (i.e. which of the several ranges does the number of occupants lie in?). To infer occupancy, we considered a variety of classifiers within a free machine learning toolkit. A major benefit of our approach is that it can be easily extended to include data from other sources, such as water flow, ambient sound, network traffic, etc.

1) Ranged Approach: Instead of attempting to predict the exact number of people in the lab, we tried to predict the occupancy in terms of ranges or levels. Ranged occupancy prediction is significantly more challenging than binary occupancy prediction but is also much more feasible than trying to predict the exact number of people. The aggregate power measurement of the lab fluctuates too much for us to be able to flawlessly capture the changes that correspond to a single person coming in or out of the lab. Furthermore, sometimes people come in the lab but do not interact with any appliances, therefore remaining invisible to our model. In such cases, any attempt at predicting the exact number of people would fail.

Therefore, in order to better understand the occupancy trends of the lab, we binned a month's worth of ground truth into 12 different bins (because 12 is the maximum number of people in the lab). We then combined the bins into 4 larger bins in order have more instances in each occupancy level. The ranges that were chosen based on the binned data are: 0, 1-2, 3-5, 6-12 people. Note that the distribution of instances is still skewed towards the lower occupancy levels, but after binning, there are at least a few hundred instances in each level.

2) Data Pre-processing and Feature Extraction: To create a dataset that can be used with our machine learning algorithms,

we first took the aggregate time-series data from the smart meter and applied a series of consecutive, non-overlapping time windows to it. Next, we extracted the following eight features from each window: mean, peak, and range for both real power and reactive power, and the difference between the current time window's mean and the previous time window's mean (for both real and reactive power). Then, we included the hour of day and light level for each time window. After that, we appended both a binary occupancy ground truth trace and a ranged occupancy ground truth trace to our dataset. Finally, our datasets were split into training and testing sets. This process involves extracting one day's worth of data from the entire set to be the testing data, while saving the rest for training.

3) Classifiers: For our experiment, we wanted to test our dataset with a variety of classifiers and compare their performances. Several machine learning libraries are publicly available in Matlab, Python, Weka, R, etc. In the end, we chose Weka, a free machine learning toolkit created by the University of Waikato, because of its ease of installation, inclusion of major classifiers, and intuitive GUI.

Within Weka, we considered their implementation of the following classifiers: NaiveBayes, Random Forest, Decision Tree, Multilayer Perceptron, and k-Nearest Neighbor. We chose that set so that we would consider at least one classifier from each of the major machine learning algorithmic categories. We then used ten-fold cross validation on the training set to find the optimal parameters for each classifier. Finally, we applied the learned model to the testing set and recorded various performance metrics.

C. Naive Strategy

All of our methods in both scenarios are compared to a "naive strategy" where no sensor information is used when making occupancy predictions. The naive strategy is important to consider because it provides a benchmark for our classifiers. For the lab, the naive strategy is as follows: Between 10pm and 9am we guess that the lab is unoccupied, between 9am and 6pm we guess that the lab has 4 occupants, and between 6pm and 10pm we guess 1.5 people (1.5 is the mean of the 2nd level of occupancy and 4 is the mean of the 3rd level of occupancy). Similarly, for the binary occupancy problem, the naive strategy guesses that during 10pm to 9am, the lab is unoccupied and on all other times, it is occupied. This strategy was designed based on prior knowledge about our lab's occupancy trends. For the home setting, the naive strategy guesses that everyone is at home throughout the night and that the occupants are totally absent from the residence during the day (7 a.m. to 6 p.m.).

IV. Experimental Setup

A. Three-person House

The motion sensor-based experiment was conducted on a house outfitted with ten PIR sensors, one in each room. We knew apriori that there are three potential occupants in this residence.

Data Collection: PIR sensor data is continuously sent to the cloud data storage service known as *Xively*. For our experiment, we queried *Xively* for nine days worth of motion sensor data, and then used cross-validation to divide the whole data set into learning and testing sets. Meanwhile, we obtained the ground truth by looking at pictures taken by the camera installed in the house and manually creating a ground truth trace.

B. Twelve-person University Lab

The electric power experiment was conducted on a 1200 sq. ft. university lab, which is occupied by 12 students. The lab is equipped with a Veris electric panel monitor, which provides both breaker-level and aggregate electricity consumption data (we use only the latter). Since the ceiling lights in the lab are connected to a non-dedicated panel elsewhere in the building, we used a light sensor to track the status of the ceiling lights and use that as a proxy for the lights' contribution to the whole lab's power consumption. Finally, the lab has a myriad of electrical loads with complex power signatures: desktop computers, peripherals, test instruments, laser cutter, thermal chamber, water dispenser, mini-fridge, specialized test-beds, etc.

Data Collection: The Veris monitor is polled every two seconds, and the resulting time series data is stored by the *Xively* cloud service. To collect ground truth, we deployed two network cameras, each pointed at one of the entry doors, and triggered image capture at a camera whenever the corresponding door is opened. The images captured are then manually analyzed to count the number of people going in or out.

V. Evaluation and Analysis

A. Accuracy Evaluation

To evaluate and compare the performance of each model, several corresponding metrics are proposed in this section to measure the ability of each model to make different types of inferences.

1) Symmetric Difference: This metric is used to compare the performances of algorithms trying to infer the identities of the occupants. Specifically, if we denote:
C_a = set of people actually in the space
C_p = set of people predicted to be in the space
Then the error is defined in (4), where SD refers to the symmetric difference.

$$Error = |SD(C_a, C_p)| \tag{4}$$

2) Binary Classification Error: For the binary occupancy inference problem, we consider the following metrics: Average Precision ($\frac{TP}{TP+FP}$), Average Recall ($\frac{TP}{TP+FN}$), and Average F-1 score ($\frac{2\times Precision\times Recall}{Precision+Recall}$) [18], where TP, FP, TN, and FN are the number of instances of true positive, false positive, true negative, and false negative respectively. In the binary occupancy scenario, positive and negative refer to predictions of "occupied" and "unoccupied". Furthermore, in all our experiments, an instance is a single time window considered in

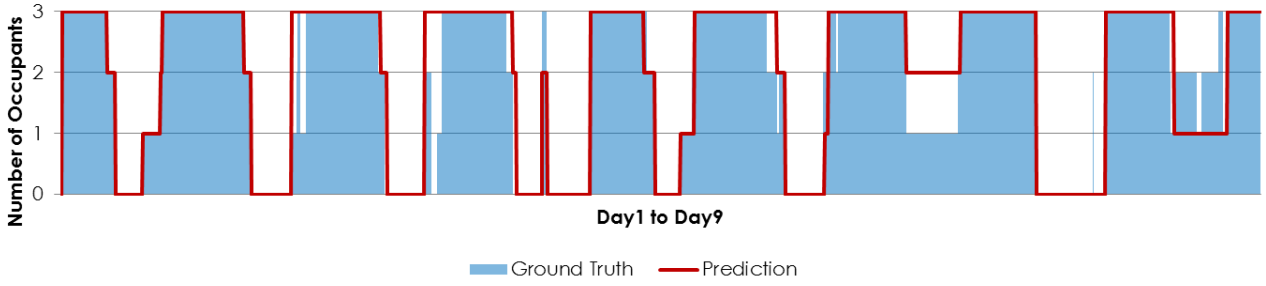
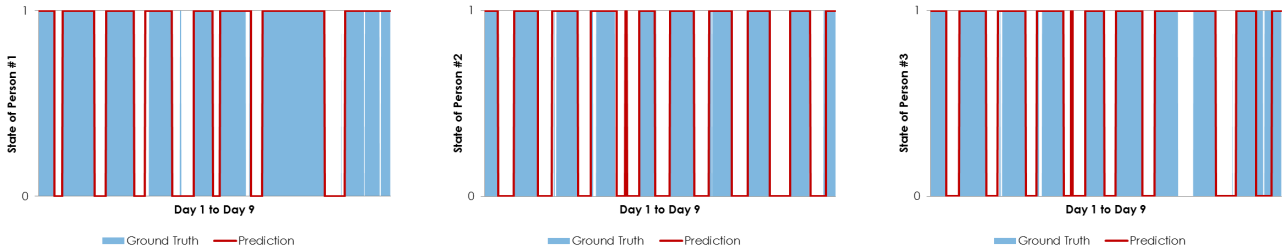


Fig. 7. Ground truth and prediction for the number of occupants using CRF+10 Sensors (9 days)



(a) State Prediction of Occupant 1 (b) State Prediction of Occupant 2 (c) State Prediction of Occupant 3

Fig. 8. Ground truth and prediction for the occupancy state of each potential occupant using CRF+10 Sensors (9 days)

our dataset. Intuitively, higher values for these three metrics mean that a model is able to make accurate predictions for both of the classes.

3) Error in Predicting Occupancy Level: For both the lab and home settings, we used our models to predict the level of occupancy, either the exact number of occupants or one of a priori selected ranges of values it lies in. We used the former for the home deployment where the maximum number of occupants is small, and the latter in the case of the laboratory where the maximum number of occupants is large and a coarser measure of occupancy level is more useful. Although our methods approach occupancy level prediction as a multi-class classification problem with each occupancy level or range corresponding to a different class, for purposes of error analysis the problem is better viewed as one of regression. A good error metric for regression problems is mean absolute error between predicted and actual value of the estimated variable. To handle the case of occupancy ranges, we represent each range by its center value. When expressed in terms of errors in classification, the error metric can be expressed as in (5), where mismatches between classes are weighed by the distance between them, and we therefore refer to the error metric as weighted classification error, or simply weighted error.

$$Weighted\ Error = \frac{\sum_{i=1}^{N}(\delta(C^i_{predict} \neq C^i_{actual}) \times D^i)}{N} \quad (5)$$

B. Motion Sensor-based Inference

1) Number of Occupants Inference: The results for this inference are shown in Fig.9, where 7 different combinations of models and sensor settings were tested for 9 days worth of data. In the best case, the CRF model in the rich-sensor setting obtained an average weighted error of 0.1929, which is almost one third of the naive strategy's error. The ground truth and prediction traces under this best case are shown in Fig.7, which further illustrates the possibility of tracking the occupants' number by solely using the motion sensors installed.

2) Identities of Occupants Inference: The results for this inference problem are presented in Fig.9, where the best performing model and settings turn out to be the same as the experiment above. The average symmetric difference here is 0.202, also lower than one-third of the Naive strategy's symmetric difference. Furthermore, the result of state prediction for each occupant is shown in Fig.8, where state 1 and 0 represents whether a certain occupant is at present or absent. The high prediction accuracy shown in Fig.8 reveals that measurements of motion sensors alone, which were regarded as the indication of binary occupancy previously, possess certain level of private information concerning the identities of occupants.

C. Electrical Power-based Inference

1) Binary Occupancy: For this inference, we considered 8 days worth of data and used a 1-minute time window for feature extraction. Five-fold cross validation was used on the dataset and results for each fold were averaged to obtain the final values. The results are shown in Fig.10. The Multilayer

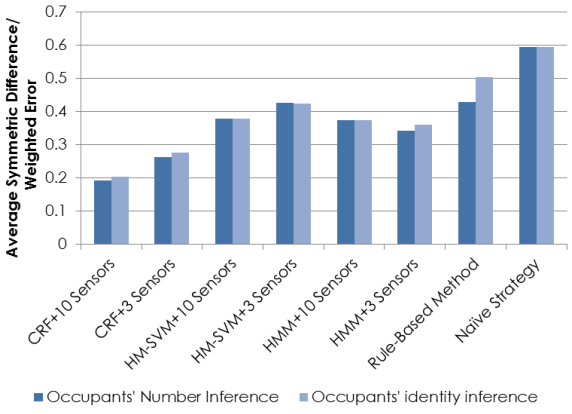


Fig. 9. Prediction error for the motion sensor-based scenario

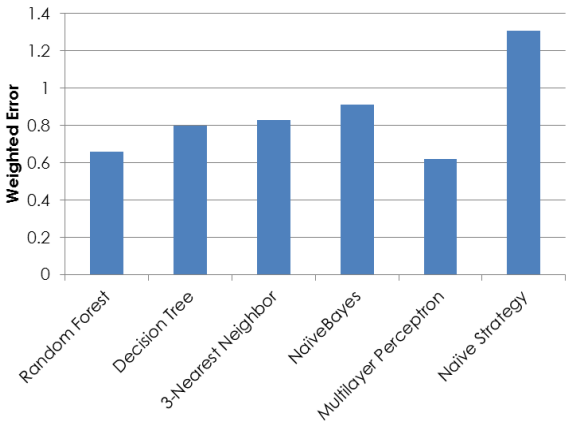


Fig. 11. Prediction error for the electric meter-based scenario (ranged occupancy inference)

Perceptron and Random Forest classifiers performed the best, with Precision, Recall, and F1 score measuring about 0.1 higher than those resulting from the naive strategy. The reason why Precision, Recall, and F1 score were so similar for each of the classifiers is because the models did not predict many False Positives or True Negatives.

The ground truth and prediction traces under the Random Forest classifier are shown in Fig.12. These low error values were expected because, intuitively, binary occupancy is easy to predict with the features that we have. Specifically, by taking into account the time of day and the light levels, we can already predict with high confidence when the lab is occupied or not.

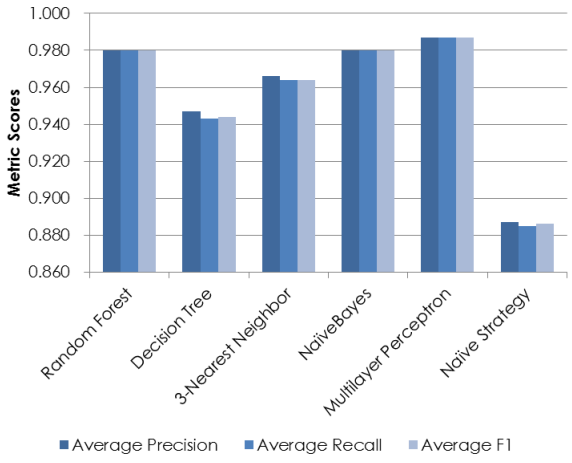


Fig. 10. Prediction error for the electric meter-based scenario (binary occupancy inference)

2) Ranged Occupancy: This experiment used the same settings as the binary occupancy scenario. The results shown in Fig.11 show that once again, the Multilayer Perceptron and Random Forest classifiers performed the best. Their average weighted errors are about half of the naive strategy's weighted error. The ground truth and prediction traces under the Random Forest classifier are shown in Fig.13.

3) Sample Complexity Experiment: The purpose of this experiment was to explore how varying the training set size would affect our classifiers' prediction error. For this experiment we varied the number of days included in the training set while keeping the testing set at one day. We used the Random Forest classifier with a time window of 1-minute for all choices of dataset sizes. We considered datasets that include the following number of days: 3, 5, 8, 10, 15, and 20.

We chose 3 days- August 21, 23, and 25 as our testing days. For each testing day we took its preceding 3, 5, 8, 10, 15, and 20 days and used those days as the training set. We then ran the classifier and averaged the weighted error for all three days. The results are shown in Fig.14.

These results are interesting and show that a large training set may not be ideal. From Fig.14, we see the error decrease until the training set is about 8 days and then start to increase again. This makes sense because a certain day's power fluctuations is more likely to be similar to the power fluctuation of days close by rather than a long time before it. By taking into account a whole preceding month's worth of power data when building a model, we introduce extra variation that decreases prediction abilities. On the other hand, by taking into account too few days, we do not capture enough of the variation that is required to build an accurate model. The optimal point seems to be around a week's worth of data.

4) Time Window Experiment: Finally, we wanted to explore how the size of the time window affects a classifier's performance. We varied the time windows from which we extract features from 1 minute to 30 seconds to 10 seconds in order to see its effects. Intuitively, a well-chosen time window is important because it can help the model better capture changes in power consumption that correspond to changes in occupancy. For example, if the time window is too large, an occupancy-related step in the aggregate power could be masked by other fluctuations when features, such as the mean, are extracted from the window. For this experiment, we stuck with the 8 day dataset previously described and ran the Random Forest algorithm to see how its performance

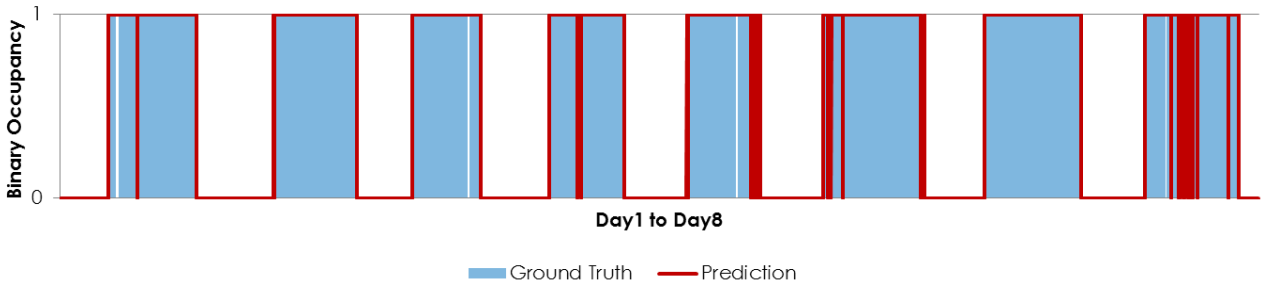


Fig. 12. Ground truth and prediction for binary occupancy using electric meter data (8 days)

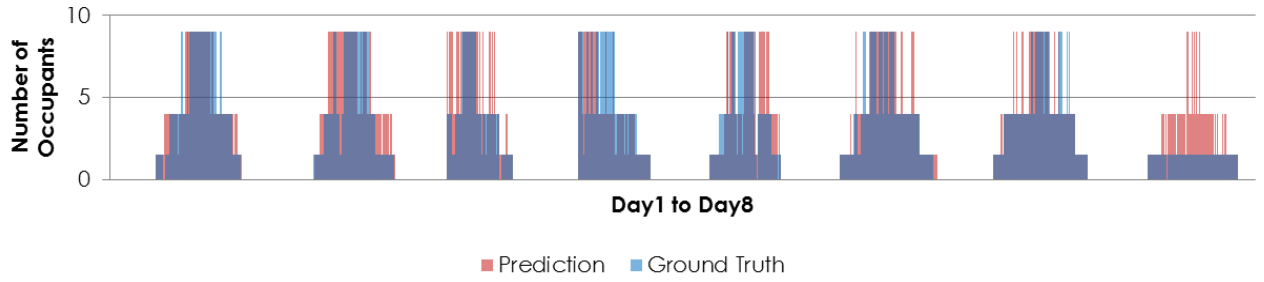


Fig. 13. Ground truth and prediction for ranged occupancy using electric meter data (8 days)

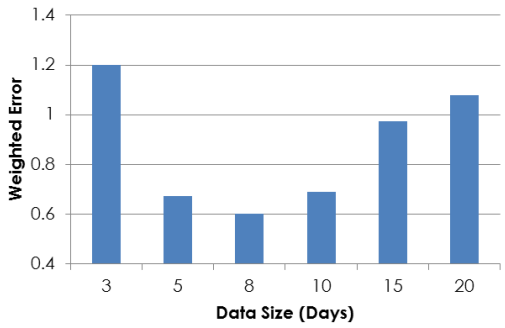


Fig. 14. Average weighted error for different training set sizes

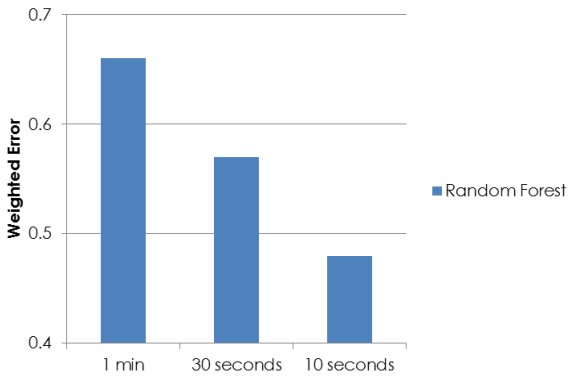


Fig. 15. Average weighted error for different time windows

varies. Results are shown in Fig.15. The results show that a shorter time window results in lower error. This implies that occupancy-related power changes most likely occur within 10 seconds of an entry or exit event.

VI. Future Work

A. Improving Motion Sensor-based Inference

1) Room-level Occupancy Inference: So far, for the motion-sensor based scenario, all experiments were done on a home-level setting. We want to explore this same problem but on a room-level setting. Specifically, our goal is to identify the exact occupants in each room. We believe this is possible as the motion sensors can record the diurnal movements of the occupants and we can infer their identities by learning different room-usage patterns of each occupant in the residence. To exploit this problem, we need more complex models and also fine-grained room-level ground truth.

2) Utilizing More Inference Models: The machine learning models presented in this paper were chosen for their capacities to describe the motion sensor scenario's time-series events. Apart from our models, a multitude of classifiers, like the ones available in Weka, are also worth trying in the future.

B. Improving Ranged Occupancy Inference

Even though our methods are able to produce inferences that have half the error of the naive strategy's inferences, we believe that more advanced techniques can decrease the error even more and help us better understand the extent of privacy leakage.

1) Load Disaggregation and Markov-related models: One idea is to apply load disaggregation techniques to the aggregate power data to see if we can better separate non-occupancy correlated loads from occupancy correlated loads. These disaggregation techniques are well studied in the field of non-intrusive load monitoring; Their goal is to take an aggregate power trace and separate it into individual appliance power traces. With these disaggregation expanded datasets, we want to apply Markov-related algorithms, such as the ones used in the motion sensor setting, to evaluate their performance.

2) Sensor Fusion: Another idea is to integrate other sources of sensor data that are also available to external service providers, to our inference process. For example, in the university lab, we could include aggregate network traffic data with our electric meter data to potentially decrease our prediction errors. Furthermore, for the home setting, we could fuse water flow data with electrical power data to make binary and ranged occupancy inferences.

VII. Conclusion

In this paper, we sought to provide insights into the risks of leaking private information presented by sensor data streams that are opportunistically available to companies providing utility and other services to a building. Our empirical work with data from motion sensors and electric meters shows that undesirable inferences about occupancy are certainly feasible, and the risk of this happening will only increase as machine learning methods advance in sophistication and as companies gain access to more background data from public sources and from other companies as part of business arrangements. Occupancy related inferences are only the tip of the iceberg, and other inferences such as specific activities being associated with specific occupants are also possible. While laws limiting how service providing companies can use such sensory data are certainly helpful (e.g. SB1476 law in California regulates the use of smart meter data by utilities), they focus on disclosure and liability, and do not prevent companies from analyzing the data to draw additional inferences for business use. A challenge, perhaps for the research community, is to develop sensing system architectures that better balance risk vs. utility by providing users with the following: visibility to the flow of data from their sensors to the service providers, an understanding of the risks that such data presents, and control over that flow via automated mechanisms as part of a "sensor firewall" that would suitably sanitize the sensor data to maintain both privacy and utility.

Acknowledgment

This research is based upon work supported in part by the NSF under awards CNS-1143667 and CNS-1213140, by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via US Navy (USN) SPAWAR Systems Center Pacific (SSCPac). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF, ODNI, IARPA, and USN SSCPac, or represent the official policies of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes not withstanding any copyright notation hereon. The first author conducted this research at UCLA while being funded as a visiting undergraduate student through the Cross-disciplinary Scholars in Science and Technology program.

References

[1] U. E. I. Administration. (2013, Jan.) Frequently asked questions. [Online]. Available: http://www.eia.gov/tools/faqs/how-many-smart-meters-are-installed-us-and-who-has-them

[2] Y. Agarwal, B. Balaji, R. Gupta, J. Lyles, M. Wei, and T. Weng, "Occupancy-driven energy management for smart building automation," in *Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building*. ACM, 2010, pp. 1–6.

[3] E. Nazerfard and D. J. Cook, "Using bayesian networks for daily activity prediction," 2013.

[4] V. L. Erickson, M. Carreira-Perpinan, and A. E. Cerpa, "Observe: Occupancy-based system for efficient reduction of hvac energy," in *Information Processing in Sensor Networks (IPSN), 2011 10th International Conference on*. IEEE, 2011, pp. 258–269.

[5] T. W. Hnat, E. Griffiths, R. Dawson, and K. Whitehouse, "Doorjamb: unobtrusive room-level tracking of people in homes using doorway sensors," in *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*. ACM, 2012, pp. 309–322.

[6] S. K. Ghai, L. V. Thanayankizil, D. P. Seetharam, and D. Chakraborty, "Occupancy detection in commercial buildings using opportunistic context sources," in *Pervasive Computing and Communications Workshops, 2012 IEEE International Conference on*. IEEE, 2012, pp. 463–466.

[7] Z. Yang, N. Li, B. Becerik-Gerber, and M. Orosz, "A multi-sensor based occupancy estimation model for supporting demand driven hvac operations," in *Proceedings of the 2012 Symposium on Simulation for Architecture and Urban Design*, ser. SimAUD '12. San Diego, CA, USA: Society for Computer Simulation International, 2012, pp. 2:1–2:8. [Online]. Available: http://dl.acm.org/citation.cfm?id=2339453.2339455

[8] A. N. R. G. Y. A. Bharathan Balajiy, Jian Xuy, "Sentinel: Occupancy based hvac actuation using existing wifi infrastructure within commercial buildings," in *Proceedings of the 11th ACM Conference on Embedded Network Sensor Systems*. ACM, 2013.

[9] D. Chen, S. Barker, A. Subbaswamy, D. Irwin, and P. Shenoy, "Non-intrusive occupancy monitoring using smart meters," in *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*. ACM, 2013, pp. 1–8.

[10] W. Kleiminger, C. Beckel, T. Staake, and S. Santini, "Occupancy detection from electricity consumption data," in *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*. ACM, 2013, pp. 1–8.

[11] W. Zucchini and I. L. MacDonald, *Hidden Markov models for time series: an introduction using R*. CRC Press, 2009.

[12] J. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," 2001.

[13] C. Sutton and A. McCallum, "An introduction to conditional random fields for relational learning," *Introduction to statistical relational learning*, vol. 93, pp. 142–146, 2007.

[14] T. Kudo. (2013, Feb.) Crf++: Yet another crf toolkit. [Online]. Available: http://crfpp.googlecode.com/svn/trunk/doc/index.html?source=navbar

[15] Y. Altun, I. Tsochantaridis, T. Hofmann *et al.*, "Hidden markov support vector machines," in *ICML*, vol. 3, 2003, pp. 3–10.

[16] T. Joachims. (2008, Aug.) Svmhmm: Sequence tagging with structural support vector machines. [Online]. Available: http://www.cs.cornell.edu/people/tj/svm_light/svm_hmm.html

[17] C. Ellis, J. Scott, I. Constandache, and M. Hazas, "Creating a room connectivity graph of a building from per-room sensor units," in *Proceedings of the Fourth ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*. ACM, 2012, pp. 177–183.

[18] josverwoerd. (2012, Dec.) Predicting with my model: Is it safe? [Online]. Available: http://blog.bigml.com/2012/12/03/predicting-with-my-model-is-it-safe/