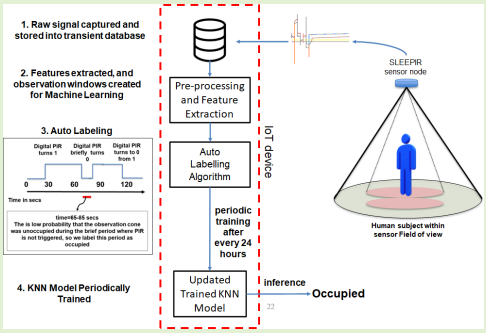
Promoting Occupancy Detection Accuracy Using On-Device Lifelong Learning

Muhammad Emad-Ud-Din and Ya Wang, *Member, IEEE*

***Abstract*—Our recently developed synchronized low-energy electronically chopped passive infrared (SLEEPIR) sensor node enables the stationary occupancy detection capability of traditional passive infrared (PIR) sensors. A machine learning (ML) algorithm reports occupancy based on a locally collected dataset from the sensor node. Though promising, the ML algorithm's detection accuracy depends on the diversity of the collected dataset—provided that the dataset contains a wide variety of infrared (IR) noise and occupancy patterns. Thus, it is challenging to train a universal ML model that contains all possible patterns. We propose an efficient *K*-nearest neighbor (KNN) occupancy classifier that incrementally adapts to the novel data from the sensor. The proposed algorithm ensures that only the relevant noise and occupancy patterns are learned. The fact that training observations are gathered on the same sensor node where the inference is made keeps the proposed classifier accurate even with the bounded size of the dataset. A small dataset and an architecture like KNN both enable the training and inference to be executed on a resource-constrained Internet of Things (IoT) device. Thus, the proposed on-device lifelong learning (ODLL) approach eliminates the need for over-the-cloud ML model updates. The dataset was collected for two distinct floorplans over two months. Results indicate an average occupancy accuracy improvement of 20.8% compared to a statically trained long short-term memory (LSTM) model. The proposed KNN model delivers comparable detection accuracy while remaining orders of magnitude faster in terms of computational performance when compared to the LSTM-based occupancy detection algorithm.**



***Index Terms*— *K*-nearest neighbor (KNN), neural networks, on-device lifelong learning (ODLL), passive infrared (PIR) sensor, smart devices.**

I. Introduction

IN THIS article, we propose an on-device lifelong learning (ODLL) approach [1] to improve the node-level detection accuracy of synchronized low-energy electronically chopped passive infrared (SLEEPIR) occupancy sensors. The SLEEPIR sensor recently developed by our team addresses a long-lasting issue of passive infrared (PIR) sensors—incapable of detecting stationary occupants [2], [3]. The SLEEPIR node consists of a single traditional PIR sensor and polymer-dispersed liquid crystal (PDLC) infrared (IR) shutters, each mounted in front of 2× PIR sensors. The PDLC shutter enables traditional PIR sensors to detect stationary occupancy by intelligently chopping the long-wave (8–12 μm) IR radiation. While the formed SLEEPIR sensor has an advantage in detecting stationary and near-stationary occupants, its performance is limited when it comes to detection range and field of view (FoV) when compared to traditional PIR sensors. To resolve this issue, a long short-term memory (LSTM) classifier has beendeployed [4] in the past to make the occupancy inference more reliable within the range and FoV of the sensor. Attempts have also been made using Bayesian techniques for improving occupancy estimation, yet considerable accuracy deterioration is noted in certain fields [5], [6] due to ever-changing environmental and occupancy scenarios. The problem with such implementations is that they lack a comprehensive training dataset that contains patterns encompassing anticipated occupancy scenarios. Such a dataset, while challenging to collect, would consequently

Manuscript received 15 August 2022; revised 25 February 2023; accepted 14 March 2023. Date of publication 24 March 2023; date of current version 1 May 2023. This work was supported in part by the U.S. Department of Energy, Advanced Research Projects Agency-Energy (ARPA-E), under Grant DE-AR0000945; and in part by the U.S. National Science Foundation (NSF) under Award CMMI 1851635 and Award ECCS 2021081. The associate editor coordinating the review of this article and approving it for publication was Dr. Ashish Pandharipande. *(Corresponding author: Ya Wang.)*

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Institutional Review Board of Texas A&M University under Application No. IRB2018-1681D.

Muhammad Emad-Ud-Din is with the Department of Computer Science, Texas A&M University, College Station, TX 77843 USA (e-mail: emaad22@tamu.edu).

Ya Wang is with the J. Mike Walker '66 Department of Mechanical Engineering, the Department of Electrical and Computer Engineering, and the Department of Biomedical Engineering, Texas A&M University, College Station, TX 77843 USA (e-mail: ya.wang@tamu.edu).

Digital Object Identifier 10.1109/JSEN.2023.3260062

require significant computational power to train and yet may still fail to adapt to novel occupancy patterns detected by the sensor. For the same reason, the machine learning (ML) models are typically trained off-site and model updates require over-the-cloud transfer. Any occupancy pattern that was not part of the off-site training dataset will likely cause degradation in accuracy.

To overcome this challenge, we propose a K-nearest-neighbor (KNN) classifier-based ODLL algorithm that can be trained "near" the SLEEPIR sensor node using an Internet of Things (IoT) device and the training dataset from the same sensor node where the occupancy inference is needed. In any classification problem, an ML model, which is tasked with predicting the class of a sample, is expected to correctly classify any input samples that may deviate by a small margin from the target [7]. However, in the case of occupancy detection, the input samples can vary from the target class, i.e., "occupied" or "unoccupied," by a large margin, depending on the occupancy or the unoccupancy scenario. In other words, for example, the "occupied" class contains several subclasses, which deviates from each other by a large margin as different occupancy scenarios can produce varying IR radiation patterns. Thus, adding more occupancy scenarios adds more subclasses to the classification problem, which impacts the accuracy of the classifier adversely [8].

Traditionally, any ML model that could not be trained at the end-user premises where the sensor node is present needs to be updated via over-the-cloud transfer. Although cloud-based training has fewer challenges in terms of application design, it comes at the cost of latency in data transfer, added connectivity and equipment overheads, and a host of security issues, such as data privacy and network attacks. Our proposed ODLL algorithm ensures that the ML training phase happens locally and that no over-the-cloud transfer is required.

There are multiple factors contributing to rendering any ML model trainable locally. First and foremost is the availability of labeled data. Since most occupancy detection systems deployed at the user premises need the capability of automatically collecting the ground truth via labeling the data, it becomes essential to collect data and train models off-site. Second, even if the ground truth can be collected, expensive computational and memory resources in the form of expensive IoT devices must be deployed for local training due to the large memory requirements to process the collected dataset. The cost of expensive IoT devices can, in turn, drive up the cost of the overall solution. The use of high-end IoT devices is feasible for some applications where accuracy is critical, as training observations are gathered from the same sensor node where the inference is made, but, in our case, where SLEEPIR sensors are applied for occupancy status detection in residential and commercial buildings, the cost and device power consumption are critical factors. This is because competing traditional PIR sensor-based solutions are extremely low-cost and power-efficient. Thus, we propose a unique observation labeling technique that combines temporal constraints on human walking velocity, time-elapsed between two consecutive PIR sensor observations, and observation distribution. This technique allows us to gather ground truth for on-site stationary occupancy without additional equipment or computing power.

For dataset collection, we deployed a total of nine SLEEPIR sensor nodes at two different floorplans. Four nodes were deployed at the floorplan labeled FP1, which resided three subjects (one adult male and two children under the age of 11). Five nodes were deployed on the floorplan labeled FP2, which resided two subjects (both adult males). Both floorplans are shown in Fig. 1. The variety in floorplans and subject profiles was deemed necessary to evaluate the proposed method under a wider variety of occupancy scenarios. The expanse for FP1 is 10 $\times$ 14 m, while for FP2, it is 9.1 $\times$ 11.6 m. As shown in Fig. 1, each node is installed at a height of 2.8 m. Each node is embedded with a traditional PIR sensor that can detect human motion in a circular area of radius 2.4 m. Each node also contains 2$\times$ SLEEPIR sensor modules, each of which can detect stationary and moving occupants in a circular area of radius of 1.2 m. Each node collects one observation every 30 s.

In this article, our focus is on node-level occupancy detection performance improvement. Once completed, we will construct a networked multiple SLEEPIR node system that utilizes the node-level occupancy and reports the accuracy of the network-level occupancy system. We aim to demonstrate network-level accuracy that ensures less than 5% chance of encountering false positives (FPs) or false negatives (FNs) in any given week. The U.S. Department of Energy lists this occupancy sensor performance standard in their Saving Energy Nationwide in Structures with Occupancy Recognition (SENSOR) Program overview [9].

Our effort aims to make the following key contributions: 1) we achieve node-level reliable occupancy detection for SLEEPIR sensors by ensuring that training observations are gathered from the same sensor node on which the inference is made, thus limiting the number of occupancy scenarios in the dataset; 2) we use an efficient technique such as KNN-based ODLL that eliminates the need for costly IoT devices; and 3) we eliminate the need for periodic over-the-cloud ML model updates, traditionally needed to address constantly changing occupancy scenarios.

We present a literature review in Section II that outlines the present state of occupancy estimation methods for human occupancy detection. In Section III, we describe the method and provide a brief overview of the SLEEPIR sensor system we use for occupancy estimation. Section IV introduces the dataset collection strategy and method of performance evaluation. Section V presents a brief discussion on addressing the issues encountered during the system design and experimentation phase. We also discuss the impact of various parameters on system accuracy. Finally, Section VII gives a brief conclusion of the presented work.

II. Literature Review

Although recursive neural networks (RNNs) such as LSTM have proven to be superior in accuracy for time-series data when compared to simpler algorithms such as decision trees and KNN, ML algorithms such as KNN are superior in terms of efficiency as these do not require training [10].

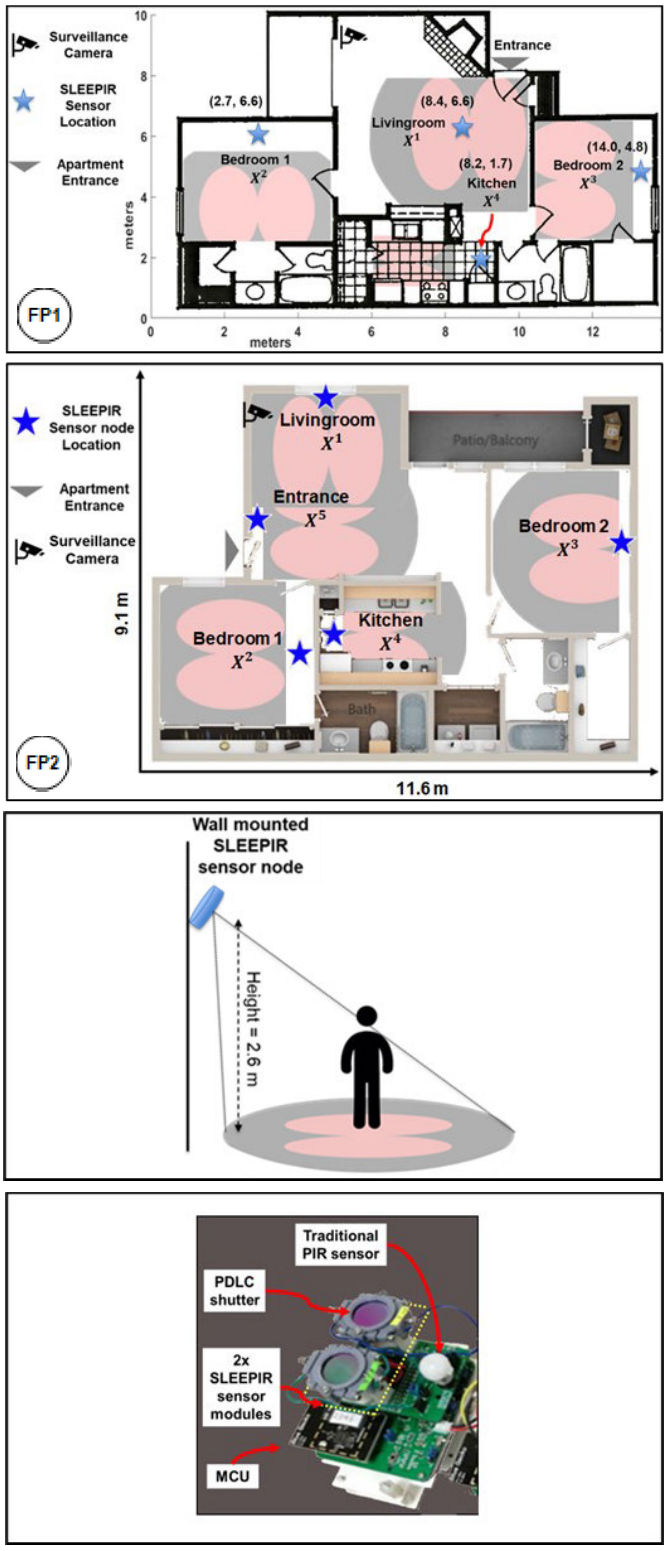


Fig. 1. We use datasets collected at two different two-bed two-bath apartments. We use four SLEEPIR sensor nodes for the floorplan labeled FP1 and five nodes for the floorplan labeled FP2 for dataset collection. Sensor node locations for both apartments are shown in the figure as well. Sensor node installation height, orientation, and footprint are shown in the middle figure. SLEEPIR sensor node is shown in the bottom figure.

The KNN classifier is a conventional nonparametric classifier that provides effective performance for optimal values of the positive integer k. In the KNN rule, a test observation (or sample) is assigned the class most frequently represented among the k nearest training samples. If two or more such classes exist, then the test sample is assigned the class with a minimum average distance. Although the KNN model is not "aware" of the temporal dynamics of the gathered observations such as the RNN model, in case when KNN is provided with a near-identical training and test dataset distribution that are collected using a specific sensor node, the performance of KNN surpasses that of a more sophisticated RNN, which is provided with a nonspecific sensor dataset. This is because every sensor has a limited number of local occupancy scenarios for which training an ML model is comparatively less challenging. We discuss this finding in Section VI. RNN models, such as continuous-time RNN (CTRNN) and LSTM, use backpropagation through time (BPTT) as the training algorithm. The computational complexity of BPTT is of order $O(n^2)$, where n is the number of noninput neurons [11]. The storage complexity of BPTT is potentially unlimited and is proportional to the number of folds in the network [11]. Thus, the computational and storage resource requirements for an unoptimized BPTT algorithm dictate off-site training for RNN models, as most IoT devices cannot perform on-site RNN training [1], [12]. Due to expensive training algorithms such as BPTT, several attempts in the literature, e.g., parameter pruning [13], quantization [14], and gradient compression [15], have been made to reduce the training algorithm complexity to perform RNN learning.

Despite these attempts to optimize on-device RNN learning, only marginal success has been achieved in terms of reducing the RNN training complexity [16]. Most IoT devices are energy constrained. Dynamic random access memory (DRAM) access consumes two orders of magnitude more energy than ON-chip static random access memory (SRAM) access [17]. Compared to other deep neural networks, typical RNNs have orders of magnitude larger memory footprint of activations, which cannot reside over ON-chip SRAM for most IoT devices; thus, DRAM access is needed during training. The training memory for an RNN should strictly fit ON-chip SRAM to achieve on-device training. This is certainly not the case given the large occupancy datasets that usually span from a few days to several months [18]. It can be argued that an already proposed version of transfer learning for the on-device learning [19] (termed TinyTL) has been employed. Since the PIR signal used by our system has unique noise and occupancy scenarios, our initial foray into the transfer learning approach yielded few encouraging results. Instead of focusing on reducing RNN training complexity, we propose to use a less expensive KNN algorithm but with an added ability to automatically label the dataset.

Apart from expensive resource usage, models suggested in our previous works [20], [21], when put to the test, produced a significant number of FPs due to environmental IR noise [20], [22]. We further investigate this finding in Section VI. We propose an automated ground-truth labeling technique that exploits the fact that if the traditional PIR sensor triggers intermittently and frequently, due to nonstationary human presence, there must be a stationary human presence even during the periods when the PIR is briefly in an

untriggered state. This novel ground-truth labeling technique ensures that the human presence IR radiation pattern unique to each sensor gets labeled correctly as true positive. It must be highlighted that in the proposed work, the local ML training dataset is only made possible due to the availability of this labeling scheme that automates the occupancy ground-truth collection. This scheme will be explained further in Section III-C.

One of the attempts made in the literature, to avoid collecting training data altogether, was to determine a general occupancy model via a semi-Markov model [23]. This attempt hinged on the notion that there exist unique Markov chains indicating occupancy in a Markov model, provided that each Markov chain embeds in it, the detection episodes of variance in light, CO_2, humidity, temperature, motion, and acoustic sensor outputs. However, the success in occupancy detection for this work was limited as the work only proposes low-resolution occupancy-centered heating, ventilation, and cooling (HVAC) control schedules generated by the semi-Markov model. Among the works that label the node-level ground truth, one of the most comprehensive publicly available labeled datasets [24] only has 14 days of labeled data. Here, labeling was done via still images captured at 1-min resolution. Another work [25] compares the performance of KNN, support vector machine (SVM), and artificial neural network (ANN) for occupancy prediction. Interestingly, while several statistical ML algorithms were compared, the dataset spanned no more than three days. In [25], the labeling was done manually via monitoring a video feed. The limited availability of labeled datasets for occupancy detection indicates a roadblock in terms of training accurate deep learning-based occupancy classifiers. Alternate established options for ground-truth collection, such as camera or thermopile array-based [26] occupancy tracking, are not feasible because of privacy concerns, high cost, and computation penalty involved. Apart from cameras, sensors, such as inertial measurement unit (IMU), visible light sensors (VLSs) [27], and Wi-Fi sensors, are either too noisy or require expensive prerequisites such and radio signal finger printing [28] in order to be part of a scalable occupancy solution.

Regarding the duration of the occupancy detection dataset, certain guidelines are presented in the Advanced Research Projects Agency–Energy (ARPA-E) SENSOR program overview [29] under the occupancy sensor testing and evaluation section. The program states that for a sensor to be widely adopted by the industry, an occupancy sensor should have a 95% probability of having no more than two failures per year. The program suggests that given a sensor produces one observation every 30 min, over a year 8760 observations will be required to establish the accuracy of the sensor. In our case, we infer occupancy once every minute instead of 30 min. A 30-min window is set to accommodate slow response occupancy sensors such as CO_2 where the sensor is not expected to respond to human presence in a shorter than 30-min span of time. Since our proposed SLEEPIR sensor and estimation algorithm can infer occupancy once every minute, it would take over six days for our system to collect 8760 observations. We have thus collected a dataset spanning over 60 days and evaluated the performance of our algorithm over a 14-day period.

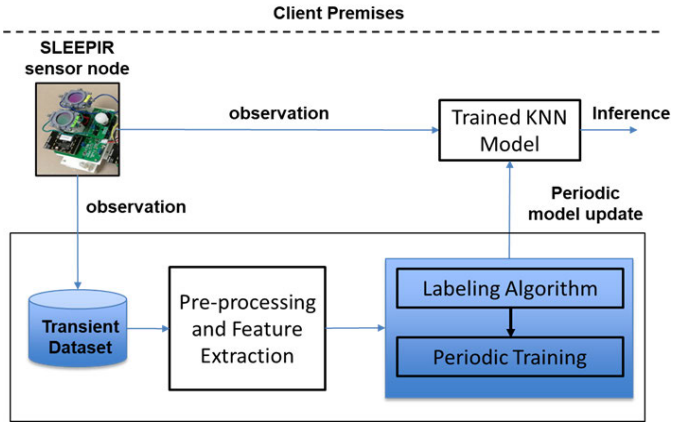


Fig. 2. SLEEPIR node generates voltage, ambient temperature, and PIR signal. The voltage data are converted to binary occupancy inference via a KNN binary classifier. Selected observations from a transient dataset are then used to periodically adapt the KNN classifier through an on-site IoT device.

Thus, after a careful review of relevant literature, it can be concluded that a constrained dataset-based classifier, such as KNN, is presently the only viable alternative to expensive RNN models, for on-device training and inference. Moreover, the proposed automated labeling scheme addresses the gap of collecting ground truth locally via utilizing the onboard PIR sensor.

III. System Design

The overall occupancy detection system flowchart is presented in Fig. 2. We present a brief algorithm flow below to summarize Fig. 2.

1) The raw sensor node inputs, which include SLEEPIR sensor module voltage, PIR sensor binary output, and ambient temperature, are collected from each sensor node via Bluetooth low-energy (BLE) communication protocol. The sensor and communication platform details are presented in Section III-A.
2) An observation consisting of raw voltage values from the SLEEPIR sensors, ambient temperature value, and traditional PIR sensor value is normalized and zero-centered. A window of l observations is forwarded to the KNN binary classifier. The binary classifier then interprets the window of SLEEPIR sensor observations and outputs in binary whether the sensor has detected human occupancy or not. This step is explained in Section III-B.
3) As and when novel labeled observations are received from the automated labeling algorithm, the KNN classifier training dataset is updated. The automated labeling algorithm is detailed in Section IV-B. The primary function of the labeling algorithm is to accept or reject incoming observations from the sensor node into a transient training dataset. This is based on predetermined criteria.

A. Working Principle of SLEEPIR Sensor Node

As shown in Fig. 1, each SLEEPIR sensor node includes two SLEEPIR sensor modules. Each sensor module consists of an analog PIR sensors (EKMC2691111K, Panasonic Inc.) mounted behind a PDLC shutter with two germanium windows that can significantly reduce the power consumption, weight, volume, and noise level, compared to mechanical choppers [30], [31], [32]. Optimal design parameters of such a sensor are provided in our previous work [22]. Onboard Silicon Labs microcontroller unit (MCU) with model no. EFR32BG13 reads the analog signals from both the SLEEPIR and the PIR sensors via analog-to-digital converter (ADC) at a sampling frequency of 20 Hz. These observations are later downsampled to 1 Hz for efficiency purposes. Afterward, the collected values are sent out as observations to a server IoT device (Raspberry Pi) via BLE connection for permanent memory storage. The FoV of the SLEEPIR sensor and the PIR are 100° (horizontal) × 100° (vertical) and 110° (horizontal) 93° (vertical), respectively.

Alongside SLEEPIR sensor modules and MCU, a traditional PIR sensor (EKMB1391111K, Panasonic Inc.), a PDLC driving circuit, and two AA batteries (3-V dc voltage supply) are present onboard.

Within the analog PIR sensor, a pyroelectric sensing element, which is made up of pyroelectric material, converts the change of heat flux to current. If the radiation power received by the pyroelectric material is $W(t) = W_0 e^{i\omega t}$, which is modulated at frequency ω, then the voltage response $V_{\text{out}}(t)$ for the preamplifier stage is in the following form:

$$V_{\text{out}}(t) = \frac{R_{\text{fb}}\eta p' A\omega}{G_T\left(1+\omega^2\tau_T^2\right)^{\frac{1}{2}}\left(1+\omega^2\tau_E^2\right)^{\frac{1}{2}}} W(t). \quad (1)$$

Here, p' is the perpendicular component of the pyroelectric coefficient p, A is the area of the sensing element, η represents the emissivity of sensing element, and $\tau_T = H/G_T$ and $\tau_E = R_{\text{fb}}C_{\text{fb}}$ represent the thermal and electrical constant, respectively. Here, G_T, R_{fb}, and C_{fb} stand for thermal capacity, thermal conductance, feedback resistance, and capacitance, respectively. Commercial-of-the-shelf PIR sensors usually consist of two or four sensing elements placed in series with opposite polarizations. By covering the sensing elements with the same polarization, the transmission change of the PDLC shutter would introduce noticeable voltage signals from the PIR sensor. When the PDLC shutter, which is mounted in front of the PIR sensor, changes its transmission periodically, the received radiation $W(t)$ changes in synchronization as well. This in turn causes the change of the output voltage $V_{\text{out}}(t)$.

B. Data Preprocessing of Sensor Signals

The SLEEPIR sensor node generates time-series observations consisting of SLEEPIR sensor module raw voltage outputs $V_{\text{out}}(t)$ (see Section III-A) and off-the-shelf digital PIR sensor output. To process these observations and infer whether the observed area is occupied or not, RNNs are an obvious choice, but these are expensive to train and usually require a significantly large dataset as they must be trained over a large number of time-series observations spanning days, if not weeks, for better accuracy. To realize the ideal scenario of local training and inference using an IoT device such as Raspberry Pi that has constrained computational power and memory, a computationally inexpensive algorithm is needed. A potential candidate is KNN that is trained over a bounded dataset. We present a detailed computational comparison among KNN and RNN algorithms in Section IV-C.

In general, KNN-based classifiers assume that the set of labeled training data is already provided and contains enough training samples to describe the class distributions in the feature space. In our case, the KNN classification is effective as the extracted features form discernable clusters in feature space. In other words, observations gathered for the cases where there is occupancy, in the feature space, should not be in proximity to the observations that are gathered, while there is no occupancy. This places emphasis on the feature determination process, which we discuss in Sections III-B.1–III-B.3.

1) Time-Series Selection and Formatting: The goal of hand-tuned ML features used widely in the literature is to produce easily distinguishable values for various data classes. A good feature remains invariant to the slight changes in the input pattern for a particular class and tends to produce roughly similar values for patterns belonging to the same class. The elements of the observation obs(t) collected at time t from the sensor node include raw voltage signals from two ADC channels of SLEEPIR sensor, i.e., $[V_{\text{out1}}(t), V_{\text{out2}}(t)]$ and a binary traditional PIR signal PIR(t). Notice that we do not collect ambient temperature as part of our dataset. The reasons for not doing so are discussed in detail in Section VI-C. We then initialize the training dataset by normalizing and zero-centering all $\text{obs}_{\mathbf{T}}$ present in the training dataset so that it has a zero mean and a standard deviation of 1.

2) Sliding Window Input Approach: Following the normalization, the observation time series obs_T, which consists of the following elements $[V_{\text{out1}}(t), V_{\text{out2}}(t), \text{PIR}(t)]$, is divided into nonoverlapping windows win_T. Here, each element win_T is created by sliding a fixed-horizon window of length equaling 8 s over the 3-D obs time series. Here, subscript T is the timestamp at which the PDLC shutter permits increased IR radiation to reach the sensor. The shutter remains in this state for 4 s. We term the PDLC shutter to be in the open state for 4 s. The remaining time the shutter is termed to be in the closed state. Thus, our window duration corresponds to 8 s where the sensor completes its response to the 4-s PDLC shutter modulation. Fig. 3 shows the SLEEPIR sensor response to the PDLC shutter where IR radiation that reaches the PIR sensor changes every 4 s.

3) Feature Extraction: For each win_T, we computed six ML features. These features are handcrafted and were tested to perform better in terms of KNN classification compared to a host of other features that were considered during the feature selection effort. Fig. 3 describes the measures used in the feature evaluation, Fig. 4 shows the features in the time-series format and Table I provides a description for each feature.

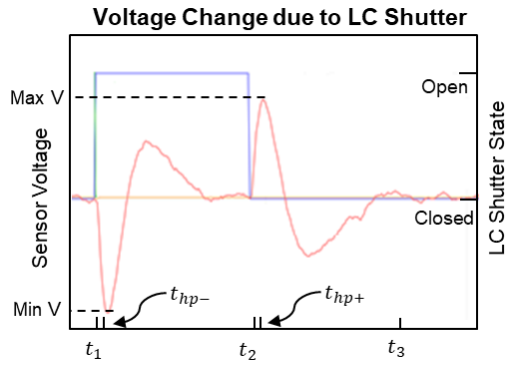


Fig. 3. Illustration of the variation of the transmitted IR radiation when the PDLC shutter opens or closes. $t_2 - t_1 = 4$ s. $t_3 - t_1 = 8$ s. t_{hp+} is timestamp when $V_{\text{out}} = \text{Max}V/\sqrt{2}$, while t_{hp-} is timestamp when $V_{\text{out}} = \text{Min}V/\sqrt{2}$.

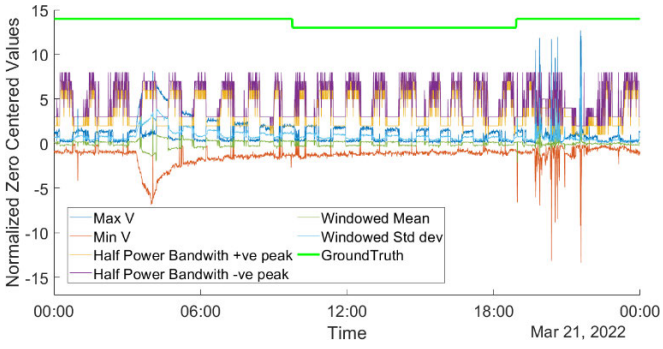


Fig. 4. Statistical features are evaluated over a 60-s observation window and are plotted over 24 h to illustrate the portions where occupancy was observed, i.e., 12 A.M.–6 A.M. and 6 P.M.–12 A.M. These features are used for training/indexing the proposed KNN algorithm.

TABLE I
FEATURE DESCRIPTION

SLEEPIR Features	Description
Max V_T	Maximum V_{out} computed over win_T
Min V_T	Minimum V_{out} computed over win_T
Half-Power Bandwidth for +ve peak (HPB+)	$t_{hp+} - t_2$ where t_{hp+} is timestamp when ($V_{out} == (Max\ V)/\sqrt{2}$))
Half-Power Bandwidth for -ve peak (HPB-)	$t_{hp-} - t_1$ where t_{hp-} is timestamp when ($V_{out} == (Min\ V)/\sqrt{2}$))
Windowed mean (mean V_T)	Mean V_{out} computed over win_T
Windowed Std. Dev (std V_T)	Standard Deviation for V_{out} computed over win_T

IV. KNN-Based Locally Trained Occupancy Classifier

A. KNN Network Architecture

KNN is a nonlinear, distance-based method, supervised classification technique. It is a direct classification method that does not require a learning process. Instead, it requires the indexed storage of the whole data. Given a training dataset (x_T, y_T), where $T = [t_1, t_1 + 30, t_1 + 60, \ldots,]$ and a test sample x_{test}, the distance, d_m, between x_{test} and x_T can be calculated as in the following equation:

$$d_m = \|x_{\text{test}} - x_T\| \tag{2}$$

where $||\cdot||$ is the distance. One of the most widely applied distance calculations is Euclidean distance. After obtaining the distance d_m, the labels of $\boldsymbol{k}$ training samples with the smallest distance can be used. Then, majority voting will be performed to determine the label of the testing sample. It must be highlighted here that as a new sample is assigned to a class, the computation time increases as a function of the existing samples in the dataset [33].

Our implementation, however, keeps the training dataset bounded via a cap on the total number of observations. This is done by periodically eliminating observations that are farthest (in terms of Euclidean distance) from the respective cluster center. We use the *Elbow method* search [34] to determine the optimal number of neighbors $\boldsymbol{k}$, which is a crucial parameter for KNN inference. This search is performed periodically rather than at every inference. This method calculates the within-cluster sum of squared errors (WSS) for different values of k neighbors using which WSS is evaluated. We then choose k for which WSS starts to diminish for the first time. In the plot of WSS-versus-k, this is visible as an elbow.

B. Automated Labeling Algorithm

For the initial training of the KNN classifier, we train the model with observations that are labeled via calibration data collected by the end user. We then initialize labels y_l where each element corresponds to each observation $x_l = [\text{Max}V_T, \text{Min}V_T, \text{HPB+}, \text{HPB-}, \text{mean}V_T, \text{std}\ V_T]$. The calibration labels are collected via a smartphone app where the end user labels 20 observation windows. Ten windows are labeled as "occupied" while the end-user ensures that there is human presence within the FoV of the SLEEPIR sensor node while ten windows are labeled as "unoccupied" while there is no human presence within the FoV of the SLEEPIR sensor.

For automatic labeling, the range and FoV of traditional PIR sensor embedded within the SLEEPIR node are critical. Experiments conducted in our earlier works [2], [3] discuss in detail the sensor installation height and orientation choices. As a result of experimentation in [2], we found that for sensor installed at a height of 2.8 m, the radius of the SLEEPIR sensor node footprint is 1.2 m, while the radius of concentric PIR sensor footprint is 2.4 m. Each sensor generates a timestamped log of occupancy status for traditional PIR sensor as follows:

$$D_T^{i\text{PIR}} = \left\{(i, T) : i \in N, T \in \mathbb{R}^+\right\}. \tag{3}$$

In (3) (i, t) denotes that the PIR sensor at location X^i was triggered at time T. Fig. 1 shows the set of locations denoted by index i, i.e., X^1, X^2, X^3, and X^4. The labeling algorithm exploits the time difference between two consecutive PIR activations for a sensor. We assume that a human subject is present within the sensor range and FoV if $D_{TT} = D_{T+1}^{i\text{PIR}} - D_T^{i\text{PIR}}$ is $\leq$60 s. In other words, we assume that the human subject did not leave the sensor footprint area if two consecutive PIR activations for the same node are $\leq$60 s apart in time.

The assumption that there certainly would be a stationary occupant in the FoV of sensor if two consecutive PIR triggers are ≤60 s is evaluated to be generally true as the training datasets labeled based on this assumption provide us with high occupancy detection accuracy. The labeling algorithm is better explained via the algorithm steps mentioned in Algorithm 1. The input includes all x_{test} feature observations that were recorded between T and $T + 1$ whenever $D_{TT} \leq 60$ s.

Algorithm 1 KNN Training Set Label Generator

Input: x_{test}, x_l, y_l, $thresh\_occ$, $thresh\_unocc$
Output: Updated Training set x_l, y_l for KNN classifier

$k_{opt} =$ **Elbow_search**(x_l, y_l);
for all i **in** $x_l where y_l == occupied$
  $[clust\_cents\_occ, loc\_occ] =$ **KMeans**(x_l, k_{opt})
for all i **in** $x_l where y_l == unoccupied$
  $[clust\_cents\_unocc, loc\_unocc] =$ **KMeans**(x_l, k_{opt})
$[dist_{occ}, idx_{occ}]$=**farthest_occ_sample**(x_l, y_l);
$[dist_{unocc}, idx_{unocc}]$=**farthest_unocc_sample**(x_l, y_l);
for all x_{test}
  for all k **in** $clusters$
    if dist$(x_{test}, clust_cents_occ) < thresh_occ$
      $y_{test} = occupied$;
      if dist$(x_{test}, clust_cents_occ) < dist_{occ}$
        if (size$(x_l) > 1000$)
        $[x_l, y_l] =$ **replace_in_dataset**$(x_{test}, y_{test}, idx_{occ})$
        else
        $[x_l, y_l] =$ **add_to_dataset**(x_{test}, y_{test})
    if dist$(x_{test}, clust_cents_unocc) < thresh_unocc$
      $y_i = unoccupied$;
      if dist$(x_i, clust_cents_unocc) < dist_{unocc}$
       if(size$(x_l) > 1000$)
       $[x_l, y_l]$=**replace_in_dataset**$(x_{test}, y_{test}, idx_{unocc})$
       else
       $[x_l, y_l] =$ **add_to_dataset**(x_{test}, y_{test})

There exists another criterion that needs to be satisfied by the feature observation x_{test} to qualify as an input to Algorithm 1. This criterion ensures that the voltage difference $\text{Max}V_T - \text{Min}V_T$ calculated over the observation window win_T must be greater than an empirically set voltage threshold termed "*zero-presence voltage difference*" or ΔV_{Thresh}. This qualifier is important as we can observe in Fig. 3 that the difference between Min V_{out} and Max V_{out} is maximum over the duration of time window when the PDLC shutter is opened and then closed. In case our assumption that a human subject is present within the sensor range and FoV while $D_{TT} = D^{i\text{PIR}}_{T+1} - D^{i\text{PIR}}_{T}$ is ≤60 s is wrong, this secondary check will ensure that the sensor is exposed to a minimal level of IR, which is expected to be radiated by a human subject. It must be mentioned here that the value of ΔV_{Thresh} will need to be minimal as, under certain scenarios, stationary human subjects are known to produce a minimal change in V_{out} [4].

Algorithm 1 starts by extracting the cluster centers for multiple clusters formed in feature space via a typical K-means algorithm. Each of these clusters can belong to either an occupied or unoccupied class. The data being clustered belong to the initial training dataset x_l, y_l recorded via user calibration. The job of Algorithm 1 is to update the initial training dataset to a more comprehensive training dataset that includes occupancy (and unoccupancy) patterns that were not captured during the calibration time period. An unclassified observation is evaluated for proximity to the clusters in the initial training dataset. If the observation is within a threshold distance from the center of a certain class, the observation is assigned the corresponding class to which the cluster belongs. Algorithm 1 bounds the size of the dataset to a limit value of 1000. An example of an updated training dataset divided into "occupied" and "unoccupied" clusters is shown in Fig. 5. This figure plots the distributed stochastic neighbor embedding (t-SNE) projection [35] of the observations. t-SNE gives us an intuition of how the data are arranged in a high-dimensional space.

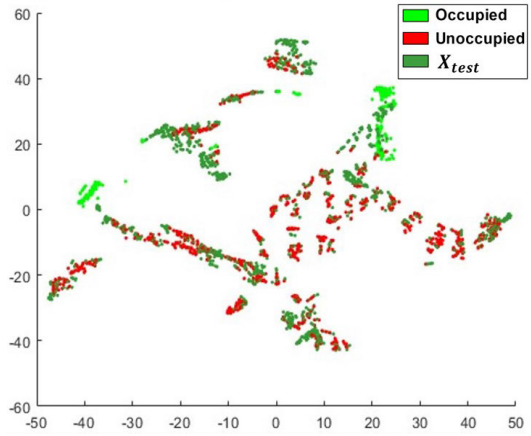


Fig. 5. T-SNE feature space plot for training dataset x_l, y_l for kitchen sensor X_4. X_{test} represents the labels that are yet to be labeled.

C. Performance Evaluation Comparison Between KNN and Other RNNs

To evaluate a performance gain for using dataset bound KNN classifier when compared to an RNN, we performed a comprehensive comparison of the KNN classifier and RNN classifier performance. Our analysis included testing the collected dataset over LSTM, CTRNN, and proposed KNN architectures. We also varied the observation window length l over a reasonable range to see whether certain networks performed better than others. We found that for $l = 60$ s, the accuracy was highest across all architectures. This indicates that the most effective discriminating features exist over a period of 60 s. It is important to mention here that SLEEPIR collects two consecutive observations over a span of 60 s. Table II outlines the performance evaluation results for comparison. Although for an unbounded dataset, KNN is not as effective in terms of accuracy as LSTM or CTRNN, it does not require expensive BPTT training as is the case with its RNN counterparts. Training durations for each of the tested algorithms are also provided in Table II. The training durations were measured on Raspberry Pi 4 using a 64-quad-core Cortex-A72 (ARM v8) processor. Table III lists the power consumed by the Raspberry

TABLE II
AVERAGE ACCURACY AND TRAINING DURATION FOR 16 NEURONS HIDDEN LAYERRNN MODELS (LSTM, CTRNN) AND FOR FIVE-NEAREST NEIGHBOR MODEL FORA TOTAL OF 1000 OBSERVATIONS

Obs Window length	LSTM		CTRNN		Proposed KNN	
	acc (%)	training duration (sec)	acc (%)	training duration (sec)	acc (%)	indexing duration (sec)
30 sec	96.4	5314	92.3	4109	91.7	19
60 sec	97.1	8722	95.5	8264	94.8	23
90 sec	91.3	9945	88.0	10427	87.3	24
120 sec	82.8	16618	7.1	15730	78.7	28

TABLE III
AVERAGE POWER CONSUMPTION FOR TRAINING 16 NEURONS HIDDEN LAYER RNNMODELS (LSTM, CTRNN) AND FOR FIVE-NEAREST NEIGHBOR MODEL FOR ATOTAL OF 1000 OBSERVATIONS

Obs Window length	LSTM	CTRNN	Proposed KNN
	avg consumption (mAh)	avg consumption (mAh)	avg consumption (mAh)
30 sec	1.03	0.96	0.0049
60 sec	2.13	1.71	0.0045
90 sec	2.25	1.98	0.0050
120 sec	3.74	3.06	0.0052

Pi 4 platform during the duration of training for LSTM, CTRNN, and KNN algorithms for a training dataset consisting of 1000 observations. Raspberry pi consumed between 3.8 and 5.5 W depending on the number of processing cores used during the training process. It may be highlighted here that the accuracy and power consumption values in Tables II and III are an average for all four nodes deployed in the system.

V. DISCUSSION

We must mention certain facts about the training set accumulated by the automatic labeling algorithm. We exclude the input observations x_{test} where traditional PIR is triggered. This is because the observations captured while the occupant in the sensor FoV is moving belong to a very different distribution compared to the ones where the subject is stationary. Since the embedded traditional PIR can determine occupancy when the occupant is moving, the KNN does not need to classify observations under such a scenario. Moreover, the observations that are obvious outliers are not included in the training dataset. Comparisons at lines 10 and 17 of Algorithm 1 ensure that whenever the incoming feature observation has a large Euclidean distance from the centers of class clusters, such an observation is discarded. By doing this, we only keep training data points in the dataset that have less than a threshold distance (thresh_occ or thresh_unocc) to the center of class clusters. It is useful to highlight here that multiple clusters can belong to a single class. Multiple clusters belonging to a single class in feature space exist because SLEEPIR voltage response is not the same to occupancy in varying circumstances, e.g., varying ambient temperature or varying environmental IR noise.

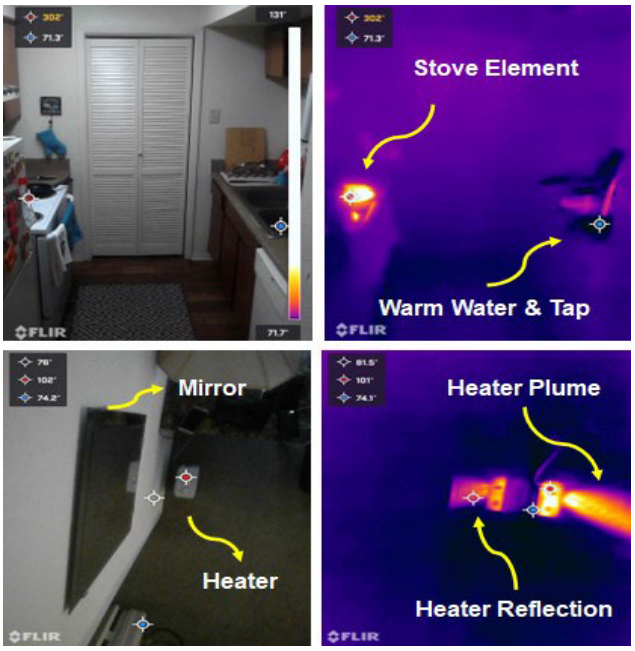


Fig. 6. Common sources of IR noise include kitchen stove, warm tap water, space heater, and electronic devices such as laptops and chargers.

We observe that multiple sources of IR noise contribute toward the FPs. We found that sources, such as tap running with warm water and warm laptop, present us with challenges in terms of false detections. Essentially, the introduction of IR noise in sensor FoV will add another cluster in the feature space that is more than likely to be adjacent to the occupancy cluster. The intercluster distance between a cluster formed by the IR noise and a cluster formed by human presence is crucial for KNN to avoid FPs. Most of our research effort was spent toward feature engineering effort that increases this intercluster distance. Fig. 6 presents us with examples of multiple IR sources within sensor FoV responsible for false detections.

VI. RESULTS

A. Dataset

We used two datasets that employ nine SLEEPIR sensor nodes, as shown in Fig. 1. Certain thresholds were used to remove noisy observations as per the literature presented in [2]. A single surveillance camera for each dataset was used to label the ground truth for all rooms as entrances of all rooms and the apartment are visible in the camera FoV. Data for 60 days were collected (30 days for each floorplan). We used 23 days of data for training and the remaining seven days of data for testing for each floorplan. We downsampled the 20-Hz raw observations to 1 Hz to optimize the processing time. Thus, 864 000 observations for each sensor node are retained in the dataset. It is also helpful to mention that the proposed algorithm makes an occupancy inference every minute. The accuracy results are reported for the kitchen sensor (X^4) and the living room (X^1) sensors. The kitchen sensor has the highest incidence of IR noise in the collected observations, i.e., frequent usage of stove, tap water, and the presence of appliances that produce IR radiation. On the other hand,

TABLE IV
ACCURACY COMPARISON BETWEEN THE PROPOSED KNN MODEL (FP1) AND THESTATIC LSTM MODEL

Date	Static LSTM Classification Accuracy X^4	Proposed KNN Classification Accuracy X^4	Static LSTM Classification Accuracy X^1	Proposed KNN Classification Accuracy X^1
15 April	76.1%	89.9%	81.0%	95.5%
16 April	83.4%	94.8%	86.8%	94.9%
17 April	69.3%	97.6%	75.3%	97.6%
18 April	62.5%	98.0%	70.9%	88.4%
19 April	56.9%	97.2%	71.6%	93.8%
20 April	74.2%	87.5%	79.2%	96.8%
21 April	82.1%	92.6%	83.5%	97.1%

TABLE V
ACCURACY COMPARISON BETWEEN THE PROPOSED KNN MODEL (FP2) AND THESTATIC LSTM MODEL

Date	Static LSTM Classification Accuracy X^4	Proposed KNN Classification Accuracy X^4	Static LSTM Classification Accuracy X^1	Proposed KNN Classification Accuracy X^1
18 June	70.9%	92.5%	83.6%	95.3%
19 June	67.4%	93.1%	76.0%	87.7%
20 June	69.4%	91.5%	72.3%	84.1%
21 June	61.6%	85.5%	77.0%	95.8%
22 June	63.1%	83.8%	75.1%	94.2%
23 June	83.6%	95.1%	90.8%	96.3%
24 June	81.3%	94.5%	92.2%	96.8%

the living room sensor only has a relatively limited number of appliances within its FoV that produces IR noise such as laptops, certain appliances, and step-down transformers in device chargers. A comparative accuracy analysis between the two rooms with different IR radiation profiles is expected to bring further insights into the impact of IR noise on the occupancy detection performance.

B. Accuracy Analysis

This work has a unique claim to train the model locally and eliminate the need for periodic over the cloud ML model updates. The work also claims to minimize the computational resource usage by the ML model while delivering comparable human occupancy accuracy when compared to a traditional RNN method. For this purpose, we used a previously deployed static LSTM model [20] trained at the laboratory for human occupancy. We compared it to the proposed KNN model, which is dynamically updated every 24 h. It must be highlighted here that the laboratory environment and IR noises were different in many respects from the local environment of the apartment. The results of this comparative study are listed in Tables IV and V. As mentioned earlier, this study only involves a kitchen sensor (X^4) and a living room sensor

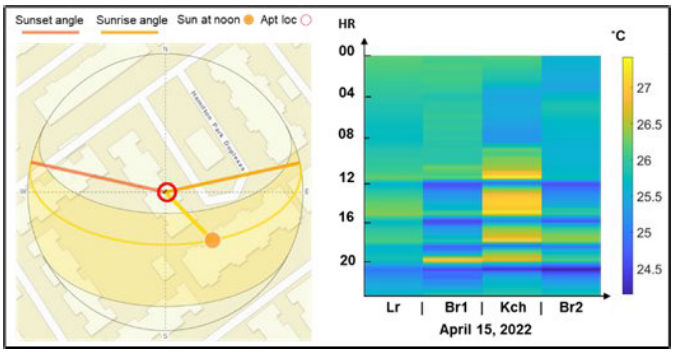


Fig. 7. (Left) Apartment location depicted along sunrise and sunset angles on April 15, 2022. (Right) Temperature profile recorded by each node for the same day. Living room exposed to sunlight IR via large glass sliding door thus warmer. Kitchen was used for cooking primarily during the day. Alternating periods of blue indicate HVAC setpoint triggers through noon until evening.

(both of which capture very different occupancy scenarios within the target apartments. The data from the remaining sensors contained infrequent IR noise sources and would have skewed the detection accuracy to be high. The accuracy analysis presented in this work is based on a sensor placed in an environment where IR noise is encountered frequently, thus representing the model performance in challenging environments. Although the IR noise encountered by the living room sensor is relatively low, it is useful to evaluate this sensor as it provides a baseline for the accuracy analysis for nodes that encounter high IR noise.

C. Results Discussion and Future Work

Despite locally collected training set, we still observe more than 10% inaccurate classifications for April 15 and 20, for the kitchen node (X^4) for the dataset FP1. We investigated the reasons and found that both for April 15 and 20, we had unusually busy days in terms of cooking and dishwashing. The extracted features from observations for both these days rendered observations far away from occupied/unoccupied cluster centers present in the training data. The maximum ambient temperature for the kitchen sensor for both these days was over 82 °F when the misclassifications occurred. We noticed that the static dataset was collected, while the average ambient temperature was 75.1 °F. Although ambient temperature significantly impacts the total IR radiation reaching the SLEEPIR sensor [20], it is not strictly correlated with occupancy. Ambient room temperature in our dataset has a constantly changing profile caused due to weather, HVAC setpoints, the difference in daytime and nighttime outdoor temperatures, and seasonal shifts in sunrise and sunset angles. Fig. 7 shows this profile for a specific day within our dataset. Moreover, IR noise sources also tend to change ambient temperature. We thus conclude that ambient temperature must be excluded from the input features for the proposed method as it has no obvious correlation to occupancy.

We also observe that on April 18 for the living room node (X^1) for the dataset FP1. On June 19 and 20 for the living room node (X^1) for dataset FP2, there are more than 10% misclassifications. If we look at Figs. 8 and 9, we observe

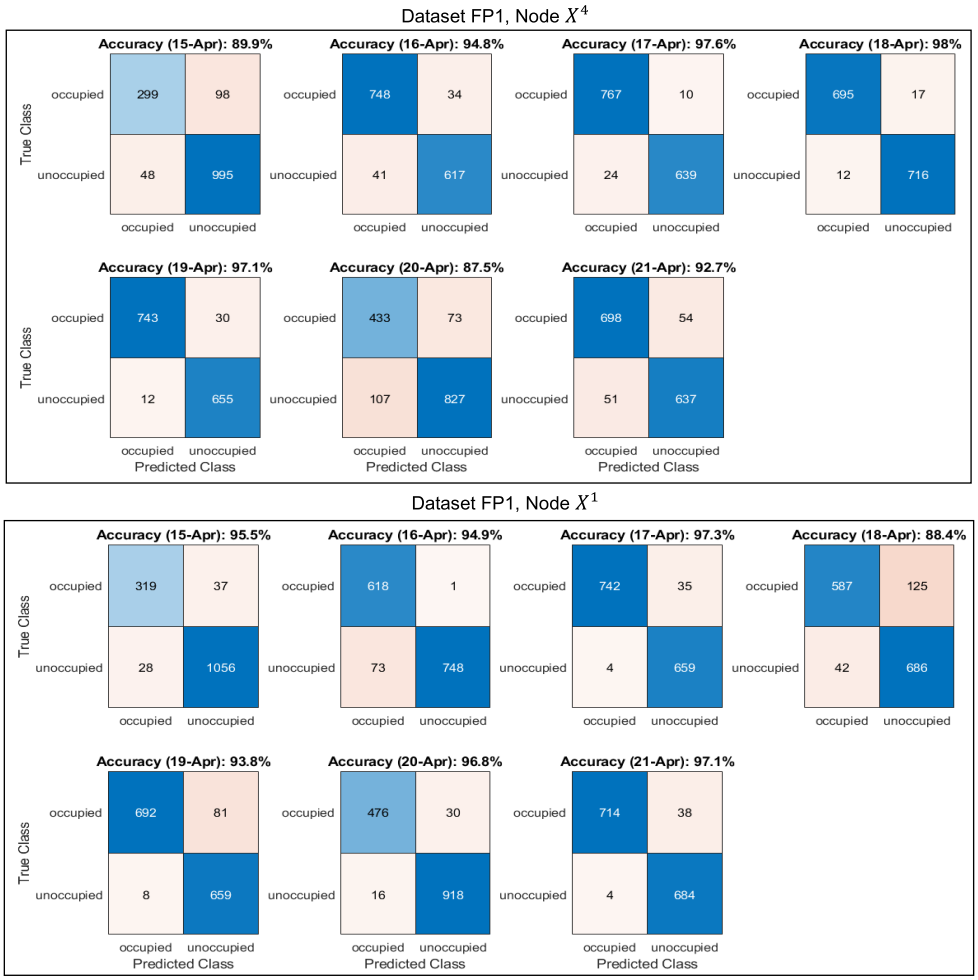


Fig. 8. Confusion matrices showing the performance for the proposed KNN classification method for kitchen node (X^4) and living room node (X^1) for dataset FP1.

that for all these days, an unusually large number of FNs are reported. The KNN was unable to recover from these FNs as these are caused due to the subjects' IR being blocked by an occlusion caused by temporary move of furniture in one case and by subject being near the edge of the detection range in the second case.

We will now explain in detail the reasons for misclassification due to sudden changes in ambient temperature and introduction of IR shielding [36] effect on the output of the proposed algorithm. The training set update via Algorithm 1 slowly modifies the calibration data clusters and may even form new clusters. Since KNN allows more visibility into the classification process, we found that IR noise in combination with high ambient temperature caused the new feature points closer to the cluster center of the false class. The local occupancy patterns observed over time manage to modify the calibration dataset clusters in the long term. Still, sudden unusual shifts in the occupancy patterns and IR noise cause FPs or FNs in the short time. The confusion matrices in Figs. 8 and 9, belonging to each test day within the dataset, provide an insight into the FPs caused by IR noise and FNs caused primarily by the IR shielding effect. Certain IR noises, such as warm water from a tap in the kitchen sink, may produce near identical features to that of the human subject. Such features need to be clarified for the classifier to produce FPs. As a future work, new features must be formulated to distinguish certain IR noises from human subjects using only a privacy-aware and cost-effective sensor such as an SLEEPIR.

It must be mentioned here that IR noise in the environment gets labeled as "occupied" in the training dataset, if it coexists with a human subject within sensor FoV. Thus, periodic training ultimately resolves FPs or FNs over time. The problem occurs when the IR noise is transient and short-term. The training dataset cannot cope with such in-consistent noise, as an example. The IR noise caused by a permanent space heater placement in the sensor FoV can be handled. In case the space heater is frequently moved in and out of the sensor FoV, Algorithm 1 cannot correctly integrate the IR noise into the dataset. This happens because a temporary IR noise observation will be included in the dataset if the value for thresh_occ is set with a higher allowance. This inclusion will render clust_cents_occ to move drastically toward the included noise point. In case the IR noise is removed from FoV and the dataset cannot update itself in case, no x_{test} feature observations are recorded between T and $T + 1$ as $D_{TT} > 60$ s. Thus, the classifier is expected to perform

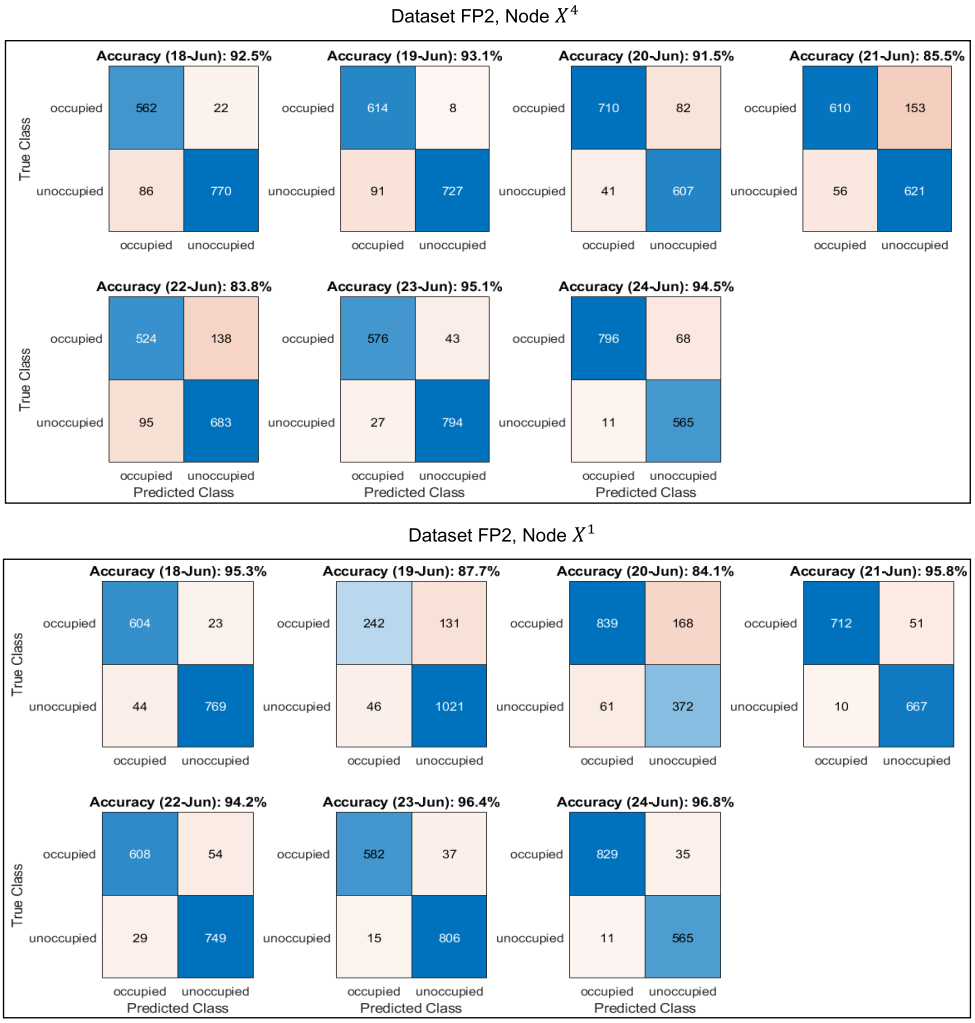


Fig. 9. Confusion matrices showing the performance for the proposed KNN classification method for kitchen node (X^4) and living room node (X^1) for dataset FP2.

inaccurate classifications until valid updated observations are integrated into the dataset.

VII. Conclusion

The proposed ODLL KNN-based occupancy classification method claims to provide superior accuracy and training efficiency compared to static RNN models. The higher accuracy and significantly efficient training hinges on the KNN model being adapted to the novel observations representing new occupancy scenarios. This is only made possible as the training dataset is labeled locally with the help of the proposed automated labeling algorithm and an initial calibration dataset. The resultant occupancy classification can deal with a host of IR noises and occupancy patterns as the training observations are gathered from the same sensor node where the inference is made. A 20.8% average improvement in accuracy was achieved over the most noise-prone sensor, while a 14.2% average improvement was achieved over the slightest noise-prone sensor. This was achieved due to the ability to train the model locally under local occupancy scenarios. The training duration for a limited training set of 1000 records was cut short by the orders of magnitude compared to traditional RNN methods. Apart from accuracy and efficiency gains, the proposed method eliminates the need for over the cloud ML model updates that are usually carried out to update the model to classify newer occupancy patterns.

References

[1] J. Lin, L. Zhu, W.-M. Chen, W.-C. Wang, C. Gan, and S. Han, "On-device training under 256KB memory," 2022, *arXiv:2206.15472*.

[2] L. Wu and Y. Wang, "Stationary and moving occupancy detection using the SLEEPIR sensor module and machine learning," *IEEE Sensors J.*, vol. 21, no. 13, pp. 14701–14708, Jul. 2021.

[3] L. Wu, F. Gou, S.-T. Wu, and Y. Wang, "SLEEPIR: Synchronized low-energy electronically chopped PIR sensor for true presence detection," *IEEE Sensors Lett.*, vol. 4, no. 3, pp. 1–4, Mar. 2020.

[4] Z. Chen, "Data processing for device-free fine-grained occupancy sensing using infrared sensors," Ph.D. dissertation, Dept. Mech. Eng., Texas A&M Univ., Ann Arbor, MI, USA, 2021.

[5] M. Emad-ud-Din, Z. Chen, L. Wu, Q. Shen, and Y. Wang, "Indoor occupancy estimation using particle filter and SLEEPIR sensor system," *IEEE Sensors J.*, vol. 22, no. 17, pp. 17173–17183, Sep. 2022, doi: 10.1109/JSEN.2022.3192270.

[6] S. Ali and N. Bouguila, "Towards scalable deployment of hidden Markov models in occupancy estimation: A novel methodology applied to the study case of occupancy detection," *Energy Buildings*, vol. 254, Jan. 2022, Art. no. 111594, doi: 10.1016/j.enbuild.2021.111594.

[7] C. Geng, S.-J. Huang, and S. Chen, "Recent advances in open set recognition: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3614–3631, Oct. 2021, doi: 10.1109/TPAMI.2020.2981604.

[8] A. Jha, M. Dave, and S. Madan, "Comparison of binary class and multi-class classifier using different data mining classification techniques," in *Proc. Int. Conf. Advancements Comput. Manag. (ICACM)*, 2019.

[9] D. M. Sofos. (2022). *Saving Energy Nationwide in Structures With Occupancy Recognition*. [Online]. Available: https://arpa-e.energy.gov/?q=arpa-e-programs/sensor

[10] C. Wang, W. Du, Z. Zhu, and Z. Yue, "The real-time big data processing method based on LSTM or GRU for the smart job shop production process," *J. Algorithms Comput. Technol.*, vol. 14, pp. 1–9, Jan. 2020, doi: 10.1177/1748302620962390.

[11] J. Schmidhuber, "A fixed size storage $O(n^3)$ time complexity learning algorithm for fully recurrent continually running networks," *Neural Comput.*, vol. 4, no. 2, pp. 243–248, Mar. 1992, doi: 10.1162/neco.1992.4.2.243.

[12] D. K. Sari, D. P. Wulandari, and Y. K. Suprapto, "Training performance of recurrent neural network using RTRL and BPTT for gamelan onset detection," *J. Phys., Conf.*, vol. 1201, no. 1, May 2019, Art. no. 012046, doi: 10.1088/1742-6596/1201/1/012046.

[13] C. L. Giles and C. W. Omlin, "Pruning recurrent neural networks for improved generalization performance," *IEEE Trans. Neural Netw.*, vol. 5, no. 5, pp. 848–851, Sep. 1994, doi: 10.1109/72.317740.

[14] B. Jacob, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 2704–2713.

[15] Y. Lin, S. Han, H. Mao, Y. Wang, and W. Dally, "Deep gradient compression: Reducing the communication bandwidth for distributed training," in *Proc. ICLR*, 2017, pp. 1–14.

[16] A. Kusupati, M. Singh, K. Bhatia, A. Kumar, P. Jain, and M. Varma, "FastGRNN: A fast, accurate, stable and tiny kilobyte sized gated recurrent neural network," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 1–12.

[17] H. Cai et al., "Enable deep learning on mobile devices: Methods, systems, and applications," *ACM Trans. Design Autom. Electron. Syst.*, vol. 27, no. 3, pp. 1–50, 2022, doi: 10.1145/3486618.

[18] W. Zhang, Y. Wu, and J. K. Calautit, "A review on occupancy prediction through machine learning for enhancing energy efficiency, air quality and thermal comfort in the built environment," *Renew. Sustain. Energy Rev.*, vol. 167, Oct. 2022, Art. no. 112704, doi: 10.1016/j.rser.2022.112704.

[19] H. Cai, C. Gan, L. Zhu, and S. Han, "TinyTL: Reduce memory, not parameters for efficient on-device learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 1–13.

[20] L. Wu, Z. Chen, and Y. Wang, "Occupancy detection using a temperature-sensitive adaptive algorithm," *IEEE Sensors Lett.*, vol. 5, no. 12, pp. 1–10, Nov. 2021, doi: 10.1109/LSENS.2021.3126285.

[21] M. Emad-ud-Din, Z. Chen, L. Wu, Q. Shen, and Y. Wang, "Indoor occupancy estimation using particle filter and SLEEPIR sensor system," *IEEE Sensors J.*, vol. 22, no. 17, pp. 17173–17183, Sep. 2022.

[22] L. Wu and Y. Wang, "Performance optimization of the SLEEPIR sensor towards indoor stationary occupancy detection," *IEEE Sensors J.*, vol. 21, no. 21, pp. 23776–23786, Nov. 2021, doi: 10.1109/JSEN.2021.3111877.

[23] B. Dong and B. Andrews, "Sensor-based occupancy behavioral pattern recognition for energy and comfort management in intelligent buildings," in *Proc. 11th Int. IBPSA Conf. Building Simul.*, 2009, pp. 1444–1451.

[24] L. M. Candanedo and V. Feldheim, "Accurate occupancy detection of an office room from light, temperature, humidity and CO_2 measurements using statistical learning models," *Energy Buildings*, vol. 112, pp. 28–39, Jan. 2016, doi: https://doi.org/10.1016/j.enbuild.2015.11.071.

[25] W. Wang, J. Chen, and T. Hong, "Occupancy prediction through machine learning and data fusion of environmental sensing and Wi-Fi sensing in buildings," *Autom. Construct.*, vol. 94, pp. 233–243, Oct. 2018, doi: 10.1016/j.autcon.2018.07.007.

[26] Z. Chen, Y. Wang, and H. Liu, "Unobtrusive sensor-based occupancy facing direction detection and tracking using advanced machine learning algorithms," *IEEE Sensors J.*, vol. 18, no. 15, pp. 6360–6368, Aug. 2018, doi: 10.1109/JSEN.2018.2844252.

[27] Y. Yang, J. Hao, J. Luo, and S. J. Pan, "CeilingSee: Device-free occupancy inference through lighting infrastructure based LED sensing," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. (PerCom)*, Mar. 2017, pp. 247–256, doi: 10.1109/PERCOM.2017.7917871.

[28] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, "SpotFi," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 4, pp. 269–282, Sep. 2015, doi: 10.1145/2829988.2787487.

[29] *Saving Energy Nationwide in Structures With Occupancy Recognition (SENSOR) Program Overview*, ARPA Energy, Dept. Energy, Washington, DC, USA, 2017, Accessed: Feb. 24, 2023. [Online]. Available: https://arpa-e.energy.gov/sites/default/files/documents/files/SENSOR_ProgramOverview_FINAL.pdf

[30] H. Liu, Y. Wang, K. Wang, and H. Lin, "Turning a pyroelectric infrared motion sensor into a high-accuracy presence detector by using a narrow semi-transparent chopper," *Appl. Phys. Lett.*, vol. 111, no. 24, 2017, Art. no. 243901.

[31] L. Wu, Y. Wang, and H. Liu, "Occupancy detection and localization by monitoring nonlinear energy flow of a shuttered passive infrared sensor," *IEEE Sensors J.*, vol. 18, no. 21, pp. 8656–8666, Nov. 2018.

[32] L. Wu and Y. Wang, "A low-power electric-mechanical driving approach for true occupancy detection using a shuttered passive infrared sensor," *IEEE Sensors J.*, vol. 19, no. 1, pp. 47–57, Jan. 2019.

[33] S. K. Trisal and A. Kaul, "K-RCC: A novel approach to reduce the computational complexity of KNN algorithm for detecting human behavior on social networks," *J. Intell. Fuzzy Syst.*, vol. 36, no. 6, pp. 5475–5497, Jun. 2019, doi: 10.3233/JIFS-181336.

[34] D. Marutho, S. H. Handaka, E. Wijaya, and Muljono, "The determination of cluster number at K-mean using elbow method and purity evaluation on headline news," in *Proc. Int. Seminar Appl. Technol. Inf. Commun.*, Sep. 2018, pp. 533–538, doi: 10.1109/ISEMANTIC.2018.8549751.

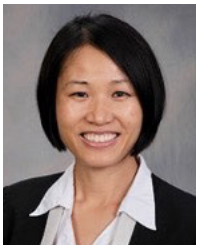
[35] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008. [Online]. Available: http://jmlr.org/papers/v9/vandermaaten08a.html

[36] S.-M. Jeong et al., "Development of a wearable infrared shield based on a polyurethane–antimony tin oxide composite fiber," *NPG Asia Mater.*, vol. 12, no. 1, p. 32, Dec. 2020, doi: 10.1038/s41427-020-0213-z.

Muhammad Emad-Ud-Din received the B.S. degree in software engineering from Foundation University Islamabad, Islamabad, Pakistan, in 2004, and the M.S. degree in computer science from the University of Southern California, Los Angeles, CA, USA, in 2008. He is pursuing the Ph.D. degree in computer science with Texas A&M University, College Station, TX, USA.

His research interests include machine learning, robotics, sensor modeling, and human activity recognition and tracking.

Mr. Emad-Ud-Din was a recipient of the Fulbright Scholarship.



Ya Wang (Member, IEEE) received the B.S. degree in mechatronics from Shandong University, Jinan, Shandong, China, in 2004, the M.S. degree in mechanical engineering from the University of Puerto Rico, Mayaguez, Puerto Rico, in 2007, and the Ph.D. degree in mechanical engineering from Virginia Polytechnic Institute and State University, Blacksburg, VA, USA, in 2012.

She is currently an Associate Professor of Mechanical Engineering, Electrical and Computer Engineering, and Biomedical Engineering with Texas A&M University, College Station, TX, USA. She has authored one book chapter, 59 journal articles, and 40 conference proceeding papers, and filed four U.S. utility patents and two provisional patents. Her recent research interests include infrared sensing, signal processing, and machine learning, with applications in occupancy detection, human identification, and activity tracking.