# Bayes Filter-Based Occupancy Detection Using Networked SLEEPIR Sensors

Muhammad Emad-ud-din, Zhangjie Chen, Qijie Shen, Libo Wu, and Ya Wang, *Member, IEEE*

Abstract—We have previously developed a synchronized low-energy electronically chopped passive infrared (SLEEPIR) sensor node that can detect both stationary and moving occupants. In this article, we present a Bayes filter (BF)-based network-level algorithm that uses a network of SLEEPIR sensor nodes deployed at a residential apartment to estimate the occupancy of the entire apartment. The method processes the incoming observations from each of the sensor nodes via a sensor model and transforms these observations into Bayesian updates. The sensor model uses a Markov decision process (MDP) formulation to estimate the temporal bounds on the rate of occupancy flow between one occupancy state to another. The overall BF output is a

probability density function (pdf) that represents the occupancy state of the entire observed space. The sensor node adjacency matrix and observation frequency are the key parameters that contribute to the sensor model design. The sensor model uses estimated transition time and probability between occupancy states to filter out observations that do not conform to the constraints set forth by the parameters. Occupancy is established through a thresholding function applied to the output pdf of the BF. A dataset was collected at a residential unit over a period of one month using the SLEEPIR sensor system. Results indicate an average 23.68% occupancy accuracy improvement when compared to the accuracy state delivered by individual SLEEPIR nodes. Results also indicate a 7.74% occupancy accuracy improvement when compared to the accuracy state determined by an already proposed particle filter (PF)-based occupancy estimation algorithm.

Index Terms—Bayes filters (BFs), occupancy detection, passive infrared (PIR) sensors, recurrent neural networks (RNNs), smart devices.
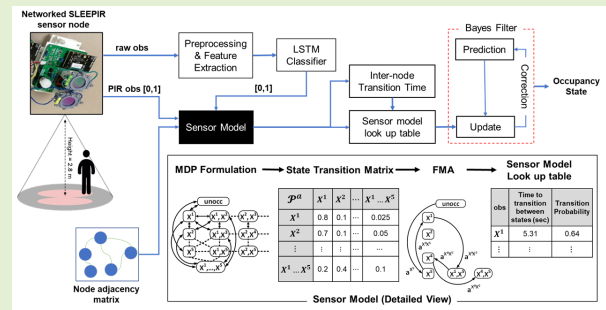
I. INTRODUCTION

TO ADDRESS the issue that standard passive infrared (PIR) sensors can only detect nonstationary occupants, our team previously developed a networked synchronized low-energy electronically chopped PIR (SLEEPIR) sensor

node [1], [2] that can detect both stationary and moving occupants by adding an electronic polymer-dispersed liquid crystal (PDLC) infrared (IR) shutter to a standard PIR sensor [3], [4]. In this article, we propose the Bayes filter (BF)-based algorithm that uses a network of SLEEPIR sensor nodes to improve the otherwise less-than-perfect occupancy detection capability of individual nodes [3].

The BF-based algorithm is more robust to the environmental IR disturbances when compared to our previously reported particle filter (PF)-based algorithm [5]. PF-based occupancy detection approach requires a considerable amount of computational capability, which does not allow such an approach to be implemented over a resource-constrained Internet of Things (IoT) device. Moreover, unlike the PF-based algorithm, the proposed BF algorithm also avoids using the historical sensor data for filtering out node-level noisy observations. The proposed method also allows the use of a minimal number of adjacent sensor nodes and exploits the adjacency information of these nodes to detect the occupancy of an entire covered space of interest.

The proposed BF-based algorithm determines the network-wide occupancy while tracking the occupancy state of the observed area, i.e., the occupancy status for each

sensor node in the system. After binarizing the node-level observations using a proposed long short-term memory (LSTM) network model in Section III-B, these observations are utilized to update a BF, which generates an estimate of occupancy at the network level.

The BF sensor model utilizes a Markov decision process (MDP) formulation [6] to model the indoor occupancy states and occupancy transition probabilities between states. We perform fundamental Markov analysis (FMA) [7] on an underlying MDP, to evaluate transition probability and expected time to travel between two occupancy states. These two parameters play a crucial role in filtering out environmental IR disturbances.

The proposed BF-based occupancy detection algorithm has the following key advantages: 1) superior detection accuracy compared to the baseline algorithms like PF [5] and extended Kalman filter (EKF) [8] based method; 2) superior computational efficiency when compared to the state-of-the-art PF-based method [5]; and 3) independence from historical sensor data to filter out noisy sensor node observations.

A literature review is presented in Section II. Section III provides an overview of the SLEEPIR sensor system and explains the input processing phase of the method. Section IV introduces the proposed method in detail. Section V presents a brief discussion about the method's design, its inherent strengths, and weaknesses. Section VI outlines the dataset collection strategy and highlights the method's performance results. Section VII provides a conclusion to the proposed work.

II. LITERATURE REVIEW

We propose a BF-based network-level algorithm that enhances detection accuracy by utilizing an occupancy picture, comprised of information flowing from multiple networked nodes. Thus, although the proposed method is agnostic to the sensor modality, it is essential to mention that PIR-based sensors are the most widely used sensor modality for occupancy detection [9]. There have been attempts to use light, temperature, sound, CO_2, reed switches, total volatile organic compounds (TVOCs), pressure, humidity, power usage, and Wi-Fi sensors [10] for occupancy detection but wide adoption of these sensors has been stemmed due to problems like slow response, high noise-to-signal ratio, and in some cases, low-correlation between occupancy and sensor observation due to changes in indoor environment. Low-cost camera-based systems have also been suggested [11], yet the lack of privacy and high computational cost to deploy such sensors have inhibited their widespread use.

Although widely used in occupancy detection [1], [3], [12], [13], tracking [14], [15], [16], and counting [17], [18], a major drawback of PIR sensors is that these cannot detect stationary occupants. Individual SLEEPIR [1], [2], [4] nodes address this issue, but at the same time, its accuracy suffers from environmental IR disturbances. Our proposed method improves this degraded accuracy by exploiting the adjacency information for networked SLEEPIR nodes.

Among all the network-level occupancy detection frameworks found in the literature, two major gaps emerged. First, in general, there was a near-linear relationship between the number of sensors and the monitored area in order to maintain the accuracy of the solution [19]. Second, virtually all of the network-level occupancy detection frameworks required historical sensor data (may it be data correlation or training datasets) to be able to filter out the noisy node-level observations. The latter gap is one of the major roadblocks in the way of the widespread adoption of network-level occupancy detection methods.

The closest work to our proposed method in the literature is a BF-based occupancy detection framework [20]. The entire dataset in this work spans over a total of 112 days, out of which 28 days are used to establish prior probabilities. Data were collected at three cities with a different sensor node deployment configuration for each city. Cellphone GPS locations and Bluetooth signal from car key Fob were used to provide occupancy ground truth. This work evaluates the correlation between individual PIR sensor node outputs, the union of PIR node outputs, BF output, and the ground truth. This work concludes that the individual PIR sensor output and the union of PIR sensor outputs have a very low correlation with the ground truth. The result shows a 25% improvement in the correlation measure between BF output and ground truth compared to the correlation between individual node outputs and ground truth. Dependence on historical data for determining prior occupancy probabilities has disadvantages like upfront data-collection costs and the possibility of noise in historical data.

Another similar work is presented in [15] that uses the adjacency relationships between PIR sensor nodes where the exact positions of the sensors are not known and thus irrelevant. The system's state in this work is modeled via a homogenous hidden Markov model (HMM). Multiple hypothesis tracking (MHT) is used to track possible trajectories of occupants using PIR sensor observations. HMMs are better suited for tracking applications as these rely on a time series of sensor observations to produce the most likely sequence of occupancy states that produced sensor observations. Our proposed work follows a different approach compared to an HMM as it only requires the most recent sensor observation to make a hypothesis about the present occupancy state. Moreover, this work requires the placement of many sensors with minimal gaps in the coverage area while leaving little unobserved space. Our proposed method uses a suboptimal configuration of sensor nodes that may leave significant gaps in coverage, and thus, our proposed method deals with a higher level of uncertainty.

We found several interesting sensor model formulations for BFs performing occupancy detection in the literature. For example, in [20], a feed-forward neural network (FFNN) to compute the likelihood function that fuses the occupancy estimate from multiple light-emitting diode (LED) sensor nodes, each of which senses the variance of diffusion reflection caused by the presence of occupants to infer the occupancy level. The redesign of the LED driver achieves the sensing mechanism to leverage LED's photoelectric effect, thus transforming a light emitter into a light sensor [21]. The historical data of six months were required to train the FFNN of 16 sensor nodes that constitute the sensor model. This work differs from [20], as we rely on the sensor node adjacency

information rather than the historical data required to train an FFNN. Another work [22] used a nonhomogeneous Markov model [23] to determine the occupancy state (zero, low, medium, and high) within an office space based on carbon dioxide concentrations. This work uses a single CO_2 sensor node. In contrast, this work considers multiple sensor nodes installed at spread-out locations within an observed space where our occupancy state is a combination of individual occupancy outputs (0,1) of all sensor nodes present in the network.

Different from non-Markov models [24], [25], a BF state is assumed to follow a first-order Markov process property [26], which states that the future state depends on the present state only. As occupancy change behavior is also a Markov process, Markov chains (a mathematical process that transitions from one state to another within a finite number of possible states) are commonly used to capture the temporal dependence of indoor occupancy [27], [28], [29], [30]. We, thus, use Markov chains to develop a realistic sensor model for our proposed BF.

Although the probability of occupancy for any given location is time-dependent, which can be ideally modeled via analysis of a first-order nonhomogeneous Markov chain (where transition probabilities vary with time) [23], we instead use first-order homogenous Markov chains to determine a statistical model for the change of occupancy state as used in previous works [30], [31]. We do this as we found a marked shift in the probability of occupancy between nighttime and daytime hours due to the shift in occupancy behavior between these hours. We use two separate homogenous Markov models for nighttime and daytime occupancy detection. This will significantly reduce the computational complexity while maintaining high detection accuracy [32].

Given the breadth of systems and techniques proposed to perform occupancy detection given two critical pieces of information, that is: 1) the intrasensor node adjacency relationship with their proximity to the observed area entrance and 2) the SLEEPIR sensor-based observations, we can safely conclude that the proposed algorithm addresses a significant occupancy detection gap with a minimal number of sensor nodes installed in suboptimal configuration.

III. System Notation, Input, and Preprocessing Algorithm

To be able to keep track of what various notations and terms mean, Table I summarizes the frequently used notation throughout this article.

The overall system flowchart is presented in Fig. 1. The raw SLEEPIR sensor observations are extracted from the SLEEPIR sensor node using a Bluetooth communication protocol. The sensor and communication platform details are presented in Section III-A. We present a brief overall algorithm flow in the following that summarizes the flowchart shown in Fig. 1.

1) The raw sensor output (which includes SLEEPIR sensor voltage, PIR sensor binary output, and ambient temperature) is collected from each sensor node via a Bluetooth communication protocol.
2) Raw voltage values from SLEEPIR sensor are preprocessed using an LSTM network-based thresholding

TABLE I
Notation Description

| Notation	Description
U_t	A probability density function (pdf) that is computed based on an incoming observation, by the sensor model, depicting the probability for each occupancy state being occupied at time t. (See section 3)
L_t	A posterior pdf that represents the represents the belief of the BF about all possible occupancy states being occupied at time t. (See section 3)
$V_{out}(t)$	The voltage response by the SLEEPIR sensor depends on the IR radiation power received by the pyroelectric material in the sensor. Units of this value are volts. (See section 3.A)
obs_T	Training dataset where each observation consists of fixed-horizon non-overlapping windows of length l from one sensor node data. (See section 3.B)
$label_T$	Occupancy labels for obs_T (See section 3.B)
$ground\text{-}truth_T$	Occupancy ground truth recorded via camera every second. (See section 3.B)
S and G	The goal state G is the occupancy state detected by the networked SLEEPIR sensor nodes. (See section 4)
$\pi^{\{state\}}$	An optimal sequence of states that need to be navigated to reach the $\{state\}$. (See section 4)
$a^{\{state\}}$	Represents an action that is expected to transition from present state to the intended $\{state\}$. (See section 4.A)
$\mathcal{P}^a_{ss'}$	A transition probability function that represents the world dynamics, telling the MDP model how likely a transition is possible between occupancy state s and s', if action a is chosen. (See section 4.A)
$Pr(G)$	A probability of reaching the goal state when starting from any state in MDP. This probability is assigned to each state in MDP. (See section 4.A)
$MC^{\pi\{state\}}$	Markov chains in our context represent a mathematical process where a sequence of MDP occupancy states are reached non-deterministically when a pre-computed policy $\pi^{\{state\}}$ suggests a sequence of actions. (See section 4. A)
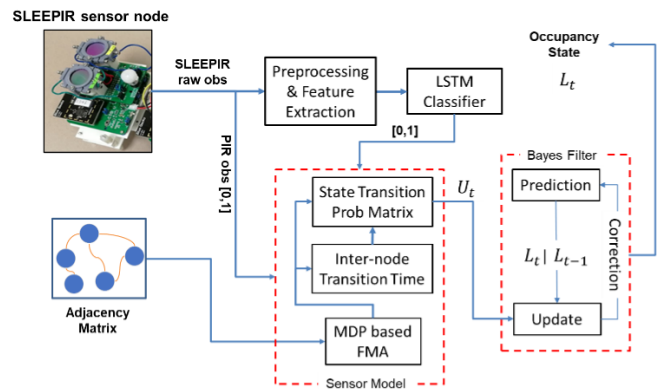
Fig. 1. BF-based occupancy detection method flowchart. Networked sensor nodes generate voltage, ambient temperature, and PIR data. The voltage is converted to binary occupancy observations via the LSTM classifier. The node-level occupancy observations then update a network-level occupancy estimate via BF.

algorithm. This algorithm is detailed in Section III-B. This thresholding algorithm classifies the raw SLEEPIR sensor observations and outputs in binary whether the sensor has detected human occupancy or not.
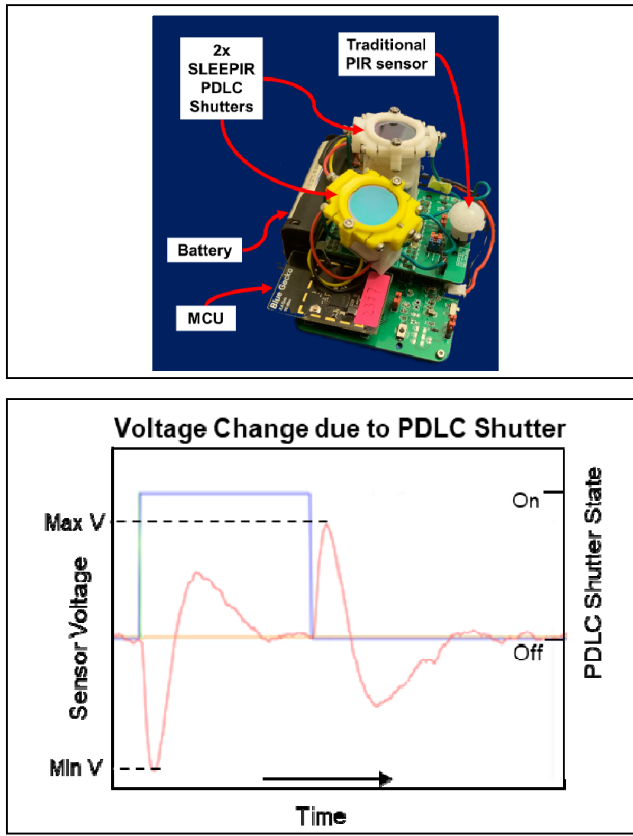
Fig. 2. SLEEPIR sensor node (top). Illustration of the sensor output voltage V_{out} due to the changing transmitted IR radiation when the PDLC shutter turns on or off (bottom).

The traditional PIR sensor output is already binary, so it does not require preprocessing.

3) The binarized observations are converted into a Bayesian update U_t via a sensor model. U_t is used to update our hypothesis L_t that points toward the occupancy state we believe the system to be currently in. This sensor model merges the output from both the SLEEPIR sensor and the traditional PIR sensor into a single BF update. This model is described in detail in Section IV-A.

4) The BF receives these periodic updates from the sensor model function and estimates the probability of human occupancy at each of the locations represented in the BF state. The world constraints are embedded in the sensor model and BF design which serves to produce a robust human occupancy belief. Details of the BF sensor model, state update, and prediction are presented in Section IV-B.

A. Synergistic SLEEPIR Sensor Node

As illustrated in Fig. 2, top, each SLEEPIR sensor node includes two PDLC shutters covering two analog PIR sensors (EKMC2691111K, Panasonic Inc.), alongside a traditional digital PIR sensor (EKMB1391111K, Panasonic Inc., Kadoma, Japan), a microcontroller unit (MCU) (EFR32BG13, Silicon Labs, Austin, TX, USA), a PDLC driving circuit, an ambient temperature sensor embedded in MCU, and two AA batteries connected in serials (3-V dc voltage supply). The PDLC

shutters cover a pyroelectric sensing element, which is made up of pyroelectric material. It 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 $V_{out}(t)$ for the preamplifier stage is in the following form:

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

Here, p' is the perpendicular component of the pyroelectric coefficient p. A is the area of the sensing element. η represents the emissivity of the sensing element; and $\tau_T = H/G_T$ and $\tau_E = R_{fb}C_{fb}$ represent the thermal and electrical constant, respectively. Here, H, 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 in front of the PIR sensor, changes its transmission periodically, the received radiation $W(t)$ changes periodically as well. This in turn causes the change of the output voltage $V_{out}(t)$. An example output signal is shown in Fig. 2.

B. LSTM Classifier

As the sensor node generates time-series observations consisting of SLEEPIR raw voltage output $V_{out}(t)$ in (1), as shown in Fig. 2, bottom, the ambient temperature $T_{amb}(t)$, and digital PIR sensor output $PIR(t)$, we employ recurrent neural networks (RNNs) to classify these observations to indicate whether the incoming observation represents human occupancy or not. RNNs when compared to the typical FFNNs have been shown to achieve higher accuracy with time-series data [33], as these can process and encode the sequential temporal information contained in time-series data. In our implemented pipeline, first, the incoming time-series data from the sensor node is zero-centered and normalized. We then divide the input time series into predetermined-sized observation windows. Each window is then labeled as either occupied or unoccupied based on the available ground truth gathered via a surveillance camera installed at the testbed. Lastly, we train an LSTM network with the training data. We deploy the trained network (shown in Fig. 3) so that the network can distinguish between the observations indicating occupancy versus those indicating nonoccupancy. We list down the machine learning (ML)-based thresholding algorithm in Sections III-B.1–III-B.3.

1) Input Formatting: The hand-tuned ML features are used widely in the literature with a goal to produce easily distinguishable values for different data classes [34]. A good feature remains invariant to the slight changes in the input pattern for a particular class and tends to produce similar values for patterns belonging to the same class. We could have chosen to quantize the input as input quantization has a proven positive impact on RNN accuracy, provided there is limited information loss [35] but we noticed insignificant accuracy improvement at the
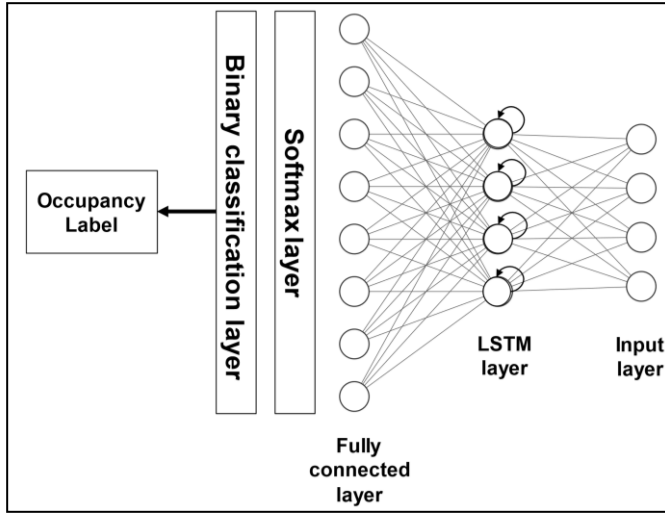
Fig. 3. LSTM network architecture for SLEEPIR raw observation binary classifier.
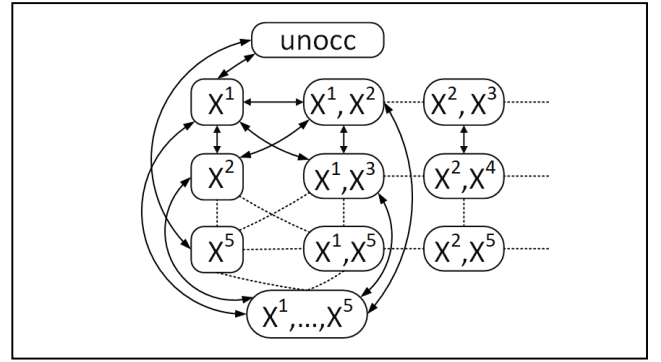

Fig. 4. Underlying MDP used by FMA. Each state s consists of a possible combination of sensor nodes where occupancy can be detected. The MDP consists of a total of 2^n states where n is the number of networked sensor nodes. Actions a connect the states and the probability for each action is defined by the node adjacency matrix and our assumptions about how quickly can the occupancy state change.

cost quantization due to the information loss thus we chose not to follow quantization approach.

2) Sliding Window Input Approach: We initialize the training dataset obs_T where each element is created by sliding a fixed-horizon window of length l over the 4-D training input time series consisting of the following elements $[V_{pp1}(t), V_{pp2}(t), T_{amb}(t), \text{PIR}(t)]$. We then initialize the labels label_T where each element corresponds to each window in obs_T. We set an element to "occupied" if a surveillance camera-based ground-truth$_T$ indicates that the human subject was present for more than 50% of observations in the field of view (FoV) of the sensor. Otherwise, the element is set to "unoccupied." A suitable window length (l) is known to be a critical parameter that has a pronounced impact on the over network accuracy [36]. We will highlight this impact on Section III-B.3.

3) LSTM Network Architecture: We use a highly cited deep forward RNN model proposed in [37], which contains multiple layers of recurrent units that are connected "forward" in time. This model architecture is simple yet powerful enough to produce reliable results over publicly available datasets which consist of time-series data. The online LSTM model shown in Fig. 3 contains a single hidden layer of 16 recurrent neurons. During the evaluation phase, all RNN models use 3, 6, 9, and 16 neurons depending upon the experimental configuration. There are also four input neurons to match the number of input time series from the sensor node, i.e., $[V_{pp1}(t), V_{pp2}(t), T_{amb}(t), \text{PIR}(t)]$. There are two output neurons to match the output classes corresponding to "occupied" and "unoccupied" status.

Emad-ud-din et al. [5] performed a comprehensive search for suitable RNNs for the occupancy detection application of SLEEPIR sensors. Our analysis included testing the collected dataset over LSTM, bidirectional LSTM (Bi-LSTM), continuous-time recurrent neural network (CTRNN), minimal gated unit (MGU) [38], and gated recurrent unit (GRU) networks. We varied the observation window length l over a reasonable range to see if certain networks perform better than others. We found that for $l = 60$ s, the accuracy was highest across all architectures. This result signified that the most effective discriminating features exist over a window length of 60 s. It must be mentioned here that SLEEPIR collects two consecutive observations over a span of 60 s. The analysis in [5] concluded that LSTM and Bi-LSTM outperformed other RNNs in nearly all window length configurations ($l = 30, 45, 60, 90$ s). We thus choose LSTM as our network of choice as it is relatively less expensive in terms of resources when compared to Bi-LSTM.

IV. BF DESIGN

After the node-level observations are binarized via the proposed ML architecture, these observations are used to update a BF which produces a network-level occupancy estimate. BF provides real-time posterior probability density function (pdf) of the state (occupancy belief) based on available information. The BF is thus "optimal" as it seeks the posterior distribution which integrates and uses all of the available information expressed by probabilities [26]. The network-level occupancy detection is a two-tiered algorithm. In the first tier, we shape the occupancy state of the monitored space as an MDP, as shown in Fig. 4. The MDP represents the dynamics of the real world. The transitions (edges) between MDP states are set based upon the adjacency of the occupancy states in real indoor observed area. When MDP is presented with a goal state G, it suggests an optimal policy π (an optimal sequence of states that need to be navigated to reach the goal state while beginning at the start state S) that leads us to G. Start state is the previous occupancy state detected estimated by the proposed BF-based method. As an example, states S or G can look like $\{X^2, X^3\}$ which indicates that occupancy was detected at nodes X^2 and X^3 and nowhere else. For example, if the starting state was {Unoccupied} and the goal state was detected to be $\{X^2, X^3\}$, then MDP would propose an optimal sequence of states π that need to be navigated to reach the goal state while beginning at the start state. The notion of π is useful when we need to evaluate the expected time to transition from the start state S to the goal state G within the MDP.

The second tier consists of a BF that periodically receives the occupancy status from individual SLEEPIR nodes. When individual SLEEPIR nodes point toward an occupancy state, we consider that state as a goal state G. Then, an optimal policy π is generated by the MDP model to reach goal G. Based upon the suggested sequence of occupancy states π, the BF continually adjusts its belief based on the degree of agreement between the observed occupancy state fed via SLEEPIR node observations and the sequence π. If an incoming occupancy observation aligns well with π suggested by the MDP, it is assigned a higher likelihood compared to an observation that does not align well.

A. MDP and State Transition Matrix Evaluation

Before we describe the elements of BF, we must describe in detail the steps that help us determine the state transition matrix, i.e., the MDP formulation, policy generation and execution, and FMA for Markov chains.

1) MDP Formulation: Each state of the MDP represents a possible combination of the occupancy status of each individual sensor. For example, a state where the entrance and kitchen nodes indicate occupancy and remaining nodes indicate unoccupancy is represented by $X^4 X^5$. Similarly, a state where all nodes are indicating occupancy will be represented by $X^1 X^2 X^3 X^4 X^5$. An MDP with mortality is a tuple as per its standard definition [6]

$$\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$$

where

$\mathcal{S}$ finite set of states s. Each state represents a possible occupancy state for the observed area;

$\mathcal{A}$ finite set of actions where each of its elements a^s is an action intended for a transition to the state s;

$\mathcal{P}$ state transition probability function $\mathcal{P}^a_{ss'} = P[\mathcal{S}_{t+1} = s' | \mathcal{S}_t = s, A_t = a]$;

$\mathcal{R}$ reward function defined by expected value function E: $\mathcal{R}^a_s = E[R_{t+1} | \mathcal{S}_t = s, A_t = a]$. A [6] large positive reward value can be set for any $s \in \mathcal{S}$, to designate s as G or goal state;

γ discount factor where $\gamma \in [0, 1]$. This enables us to model the weight assigned to the future reward at each time step.

Our belief Bel about which states are occupied (or unoccupied), transitions as we move state to state with the MDP via actions $\mathcal{A}$ available to us. $\mathcal{A}$ needs to be defined in detail here. Simply put it is a set of actions for all possible states that add up to 2^5

$$\mathcal{A} = \{a^{\text{unoccupied}}, a^{X^1}, a^{X^2}, a^{X^3}, a^{X^4}, a^{X^5}, a^{X^1 X^2}, \\ a^{X^1 X^3}, \ldots, a^{X^1 X^2 X^3 X^4 X^5}\}.$$

Here, the superscript for each action $a^{\{\text{state}\}}$ represents the state to which the corresponding action will generate a transition.

2) Node Adjacency Matrix and MDP Policy Generation: For an MDP to be able to reach a solution (to generate a policy), a transition probability function $\mathcal{P}$ must assign outgoing probabilities to each state. This function essentially represents the world dynamics, telling the model how likely a transition is possible between any two states within the MDP model. Each of the state in the MDP model is assigned a set of outgoing probabilities by $\mathcal{P}$ depending upon the world dynamics. In the case of indoor human occupancy detection, we assign these probabilities using the node adjacency matrix (ADJ) available to us. The sensor node adjacency matrix must also include the adjacency information for each node to the entrances of the observed space. We use the following expression to assign the probabilities to the state transition matrix:

$$\mathcal{P}^a_{ss'} = \begin{cases} 0.8 \times \Pr_{ss'} \times T^a, & \text{if } (\text{ADJ}(s, s') = 1) \\ 0.2 \times \Pr_{ss'} \times T^a, & \text{if } (\text{ADJ}(s, s') = 0) \\ 1.0 \times \Pr_{ss'} \times T^a, & \text{if } (\text{ADJ}(s, s') = 1) \\ & \&\& \; (s' = \text{unoccupied}). \end{cases} \quad (2)$$

We must mention here that the state transition matrix $\mathcal{P}^a_{ss'}$ mentioned above represents daytime transition probabilities. For nighttime transition probabilities, we modify the last rule to

$$0.5 \times \Pr_{ss'} \times T^a, \quad \text{if } (\text{ADJ}(s, s') = 1) \;\&\&\; (s' = \text{unoccupied}).$$

We make this modification as the occupancy state is less likely to transition to an unoccupied state during the nighttime hours. This will help reduce false negative (FN) detections caused by IR shielding. Here, adj represents the SLEEPIR sensor node adjacency matrix. T^a is the action probability dictated to us by the transition model. The transition model determines the probability for the outcome for action a. We assume that $T^a = 0.7$ when action a is taken and $T^a = 0.3$ when an action a' is taken where $a \neq a'$. Further explanation about transition model is provided in [6]. is the probability of transitioning from state s to state s', evaluated by FMA detailed in Section IV-A.4. We can observe in (2) that the transition probabilities are higher where intersensor proximity is higher.

We also need to assign values to $\mathcal{R}$, which represents a specific reward value associated with each state in the MDP. A higher reward value makes it highly probable for a policy to generate a Markov chain that contains the corresponding state. A lower or negative reward value makes it highly improbable for a policy to generate a Markov chain that contains the corresponding state. We, thus, ensure to set a high reward value for the state that we consider to be a goal state G. For remaining states in the MDP, we associate a value near 0 to indicate that we have no preference vis-à-vis states other than the goal state.

We then use a standard policy iteration-based dynamic programming solution as mentioned in [6] that uses $\mathcal{P}$ and $\mathcal{R}$, to generate a policy π. We generate 2^5 policies by setting each state as our goal state G, one by one, and then generate a corresponding policy for that state so the set of policies forwarded to BF will be as follows:

$$\Pi = \{\pi^{\text{unoccupied}}, \pi^{X^1}, \pi^{X^2}, \pi^{X^3}, \pi^{X^4}, \pi^{X^5}, \ldots, \\ \pi^{X^1 X^2 X^3 X^4 X^5}\}.$$
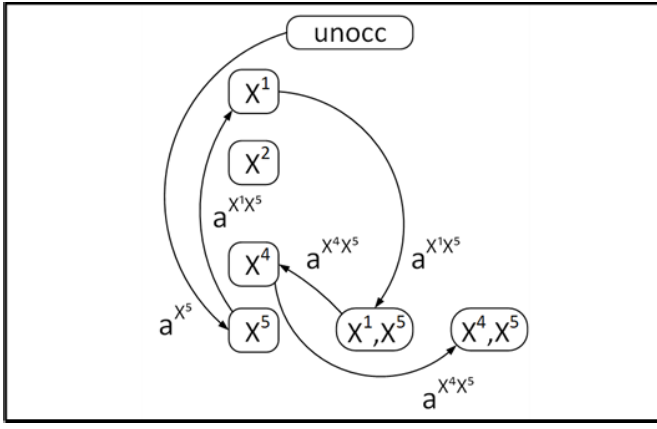
Fig. 5. Sample policy $\pi^{X^4X^5}$ evaluated by the policy iteration algorithm. A corresponding Markov chain MC($\pi^{X^4X^5}$) is generated when the policy is executed. World dynamics essentially dictate that certain state transitions are necessary before the state X^4X^5 can be reached by the occupant when starting from the unoccupied state.

We then forward this set of policies Π to the BF algorithm.

3) MDP Policy Execution: In this phase, at the start of execution, we are provided a starting state which is the unoccupied state, signifying that all observed area is unoccupied. We are also provided a goal state G which corresponds to the present occupancy scenario as observed by the SLEEPIR sensors. For example, if the occupancy is detected at the nodes X^4 and X^5 and no occupancy is detected at the remaining nodes, the chosen goal state is $G = \{X^6X^7\}$. We then create a Markov chain MC$^{\pi^G}$ which is created as per a standard nondeterministic process [6] and uses precomputed policy π^G computed during the last step. More specifically, the process executes the action choice by having the transition from the state s to s' depending on the actions stipulated in the policy at s, i.e., $\sum_{a \in \mathcal{A}} \Pr[\pi^G(s)]\mathcal{P}^a_{ss'}$. We show via Fig. 5, how a sample policy let us say $\pi^{X^4X^5}$ is evaluated. What matters to us the most is that how far into the MC$^{\pi^G}$, the goal state X^4X^5 occurs. If the G occurs too far into the MC$^{\pi^G}$, then it will be unrealistic to assign high credibility to the observation. If G occurs too early into an MC$^{\pi^G}$, then again it would not make sense to assign high credibility to the observation. To be able to determine a sensible threshold of where the G should occur in the MC$^{\pi^G}$, we must do an MDP-based analysis called FMA of Markov chains. We will go into the details of this analysis in Section IV-B. It must be mentioned here that we are evaluating the position of G in the MC$^{\pi^G}$ to be able to assign probabilities to each observation via a sensor model, depending upon how much the observation conforms to real-world temporal and spatial constraints. For the sake of clarity, we must mention that a sequence of nondeterministic actions $\{a^{X^5}, a^{X^1X^5}, a^{X^1X^5}, a^{X^4X^5}, a^{X^4X^5}\}$ was suggested by the policy $\pi^{X^4X^5}$ that resulted in the Markov chain $\{X^5, X^1, X^1X^5, X^4, X^4X^5\}$.

4) FMA of Markov Chains: For a discrete absorbing Markov chain, with one absorbing goal state and $n-1$ transient states, there exists an associated summary of expected temporal behavior that can be characterized via a single $(n-1) \times (n-1)$ matrix called the fundamental matrix [7]. Let us denote the fundamental matrix for MC$^{\pi^G}$ by N^{π^G}. The entry $n_{ij}^{\pi^G}$ of this

matrix gives the expected number of times the occupancy is in a transient state s_j given that it started in a transient state s_i. Assuming the initial state of the occupancy is at s_i, we need to compute the following two pieces of information as per [7].

1) *Interstate Transition Time:* This gives us the expected number of time steps to reach the goal state G. We evaluate the time to reach the goal as: $\tau^G = \sum_{j=1}^{n-1} n_{ij}^{\pi^G}$.

2) *Interstate Transition Probability:* This gives the probability that the chain is absorbed in state G as $\Pr(G) = \sum_{j=1}^{n-1} n_{ij}^{\pi^G} r_{jG}$, where r_{jG} describes the probability of transiting from a transient state s_j to absorbing state G, a submatrix of elements from the transition matrix of MC$^{\pi^G}$.

The interstate transition time parameter essentially serves as the gatekeeper for any observation to be able to update the BF belief, as noisy observations often suggest unrealistic transition times for an occupant to travel from one occupancy state to the other. Similarly, interstate transition probability provides a probability of transitioning between two occupancy states which are crucial to the sensor model of the BF.

B. Bayes Filter

We wish to estimate the occupancy state x for the overall monitored indoor space based on the observations received from the individual SLEEPIR sensor nodes. We treat the estimation problem via a BF. A combined occupancy snapshot from all individual SLEEPIR sensors in the network can be referred to as z_t for the observation at time t, and then, a suitable update equation as per the BF definition [39] will be

$$p(x \mid z_t, \pi^G) \propto p(z_t \mid x, \pi^G)p(x \mid z_{t-1}, \pi^G). \tag{3}$$

In the above expression, we have been explicit in the dependence on policy π. This means that the estimate of the occupancy state is assumed to be dependent upon the latest observation z_t and the policy π^G where state G is the occupancy that the observation z_t is pointing to. For example, if only two sensors X^4 and X^4 are indicating occupancy at time t, we set state G as "X^4X^5." Thus, the above recursive expression gives the complete form of the filter. However, we would like to explain in detail the following essential elements, i.e., observation formulation, concept of time, and sensor model.

1) Observation Formulation: A collective observation z_t generated by a network of SLEEPIR sensor nodes at time t is denoted as follows:

$$z_t = \{z_t^{X_1}, z_t^{X_2}, z_t^{X_3}, z_t^{X_4}, z_t^{X_5}\}, \quad \text{where } z_t^{X_i} \in \{0, 1\}.$$

For example, if there is no human detection at the node level at time $t = 1$, then $z_1 = \{0, 0, 0, 0, 0\}$. Note that $\{0, 0, 0, 0, 0\}$ is a special state referred to as an "unoccupied state" as it represents unoccupancy while all other states represent occupancy. The unoccupied state is shown in Fig. 5.

2) Concept of Time: Since the individual sensors produce at least one observation about human presence every 30 s, the filter can receive z_t every 30 s and can process the observations without any lapses the observations. In case there is a lapse in the observations due to a malfunction, the sensor model can

address such a scenario. It is essential to state here that we set $G = z_t$ whenever each observation arrives. This also means that we will be using a policy π^G, time to absorption τ^G, and $\Pr(G)$ for filtering the observation.

3) Sensor Model: The sensor model, $p(x \mid z_t, \pi^G)$, can now be written in terms of two outcomes of FMA, i.e., time-to-absorption and absorption probabilities

$$p\left(x \mid z_t, \pi^G\right) = \begin{cases} \Pr(G), & \text{if } \Delta t \leq \varepsilon\tau^G \\ 0, & \text{if } \Delta t > \varepsilon\tau^G \\ 0, & \text{if } \Delta t \leq \varepsilon\tau^G \text{ and } z_t = z_t^{\varphi}. \end{cases}$$

The sensor model simply makes sure that 0-occupancy probability is assigned to the observation if the observation state cannot be reached in the time $\varepsilon\tau^G$. Δt is the duration between two consecutive observations. Scalar ε is the safety margin that accounts for any underestimation when it comes to the time-to-absorption calculation by the FMA. The sensor model also assigns a 0-occupancy probability in case the observation points to the unoccupied state. In all other cases where observation indicates an occupied state, a calculated time to absorption $\Pr(G)$ is a suitable estimation of likelihood. In other words, the sensor model helps filter out the observations that do not conform to the expected time suggested by the FMA.

V. Discussion

The occupancy output for (3) is a pdf. This pdf represents the probability of human occupancy over all possible combinations of locations. Whichever location combination has the maximum probability at time t, would be our belief Bel_t about the state of the occupancy. A time plot that represents the shifting belief Bel_t about the human occupancy over time is shown in Fig. 6 (top). This plot shows the progression of the network-level output of our proposed method as the BF belief switches between Unoccupied and the remaining occupied states based on the sensor model and the sensor nodes input. The output is for 24 h of observations collected during the day. It is interesting to show here how traditional PIR output corresponds to the SLEEPIR sensor system output. Fig. 6 (bottom) superimposes the traditional PIR output over the same 24-h ground-truth period, as shown in Fig. 6 (top). Fig. 6 (bottom) shows that the traditional PIR only contributes a small fraction toward overall detected occupancy for the same period. We can also see that during the night hours, PIR activity is limited and mostly restricted to bedrooms. We can also observe FNs during the night hours (see Fig. 6, top), as the ML thresholding algorithm can often not detect the sleeping subject covered in a blanket with little skin exposed due to the IR shielding effect. It can also be observed that entrance PIR is triggered at the time when the switch between occupied and unoccupied states occurs. This is because there is only a single entrance/exit to the apartment and the entrance node registers a PIR observation whenever an entry or exit event happens. Most PIR-based systems depend on such entrance and exit events, but it is difficult to determine whether the event was an entry or an exit in certain scenarios using only traditional PIR sensors. This is because a single PIR sensor can easily get confused between exit and entry given the complex
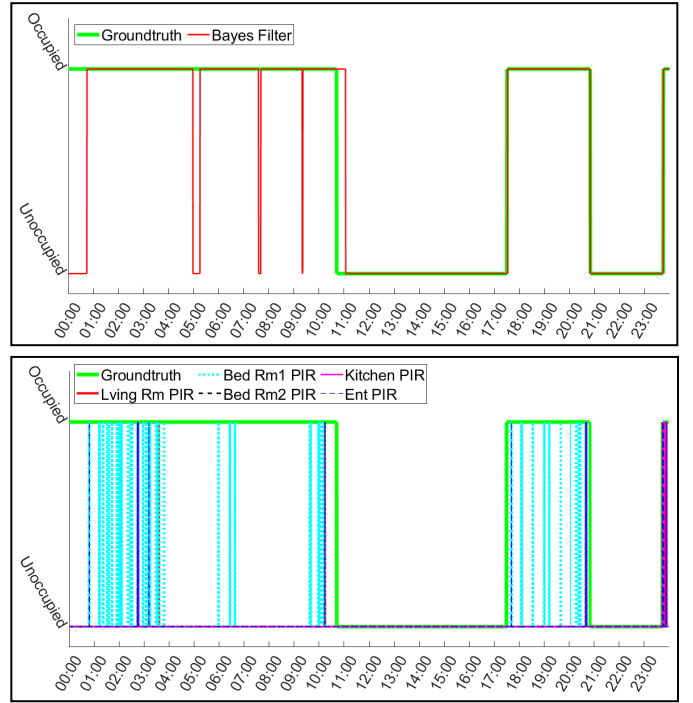

Fig. 6. Time plot that shows the progression of occupancy state through a typical day (top). The BF output is compared with the apartment level occupancy ground truth. Time plot that shows the contribution of traditional PIR observations from each of the sensor nodes toward overall occupancy (bottom). We notice that PIR activations are few and far between especially between 3- and 9-A.M. windows.

human motion behavior, e.g., there may be multiple subjects involved or simultaneous entry and exit events can happen.

Section VI must be preceded by certain assumptions that the proposed method relies on, to provide reliable occupancy, namely: 1) occupants entering the apartment will trigger at least one SLEEPIR sensor node; 2) the SLEEPIR sensors have a limited FoV and consequently have sparse coverage; and 3) all human subjects within FoV do not use any specialized means to shield the emitted body IR radiation.

VI. Results

A. Dataset

The experimental testbed for dataset collection was a two-bed two-bath, first-floor residential apartment. There were at least two occupants who used this apartment as their primary residence. The sensor node layout configuration is shown in Fig. 7. The experimentation was completely uncontrolled. We did not explore the optimal sensor node configuration for best coverage since we expect the sensor node network to be installed by a nonexpert user who may choose to deploy sensors in a suboptimal configuration. We, thus, used a single nonoptimized deployment configuration. We installed five SLEEPIR sensor nodes in the months of April, May, and June where the average outdoor temperatures range between 59 and 91 °F. Each node collects the observation every 30 s. The SLEEPIR observations were evaluated using the raw SLEEPIR sensor voltage values. Certain thresholds were used to remove noisy observations as per the literature presented in [1]. Manual logging was used to label the ground truth.
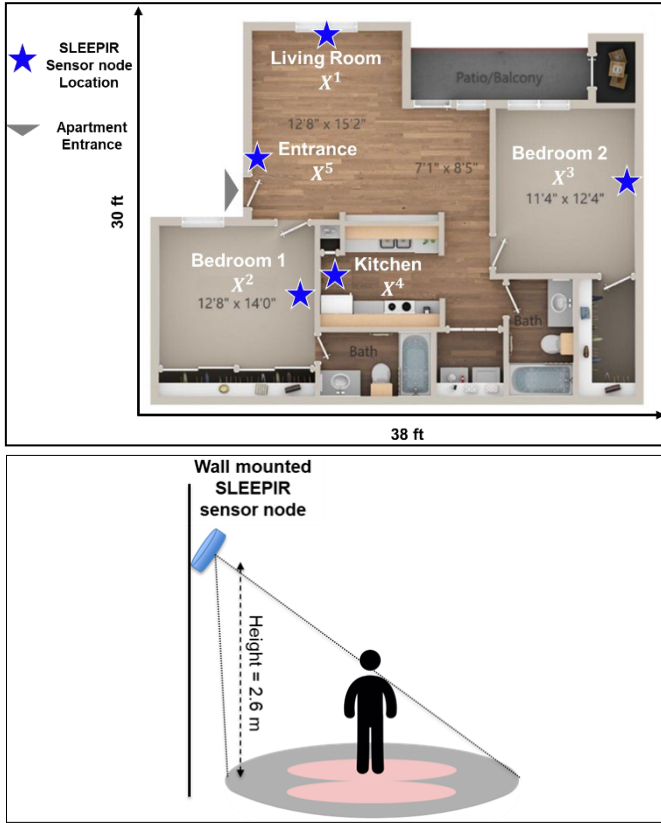
Fig. 7. Human occupancy is estimated for the locations X^1 through X^5 (top). Apartment-level occupancy ground truth is collected via manual entries to a log register. The pink footprint is shown for SLEEPIR sensor modules (bottom). The gray footprint is shown for traditional PIR sensor.
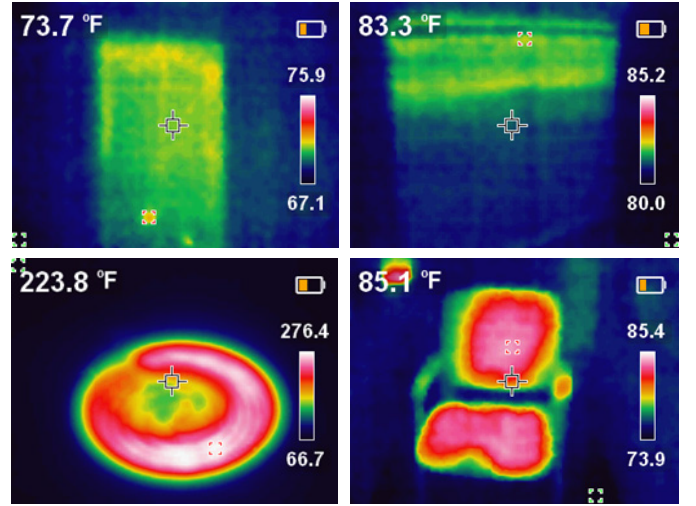

Fig. 8. Common sources of IR noise in the dataset included kitchen stove (bottom left), human body heat transferred to the chair (bottom right), solar IR entering through the living room window (top left), and bedroom 2 window (top right).

TABLE II
IMPACT OF REDUCTION OF SENSOR NODES IN THE NETWORK

Method	1-node Accuracy (X^5)	3-node Accuracy (X^5, X^2, X^3)	5-node Accuracy (*All nodes*)
EKF	33.27±9.77%	64.97±7.63%	76.52±5.01%
PF	37.19±6.81%	77.20±4.81%	84.29±4.38%
BF	40.05±4.06%	88.17±4.59%	92.04±4.40%

Apartment-level occupancy observations were noted down as logbook entries whenever anyone entered or left the apartment. Data for a total of 30 days was collected. We sampled the observations down to one observation per minute. This provided us with a total of 43 200 observations for each sensor, within the dataset. For the ML thresholding algorithm, we use 80% of observations for training, 10% for validation, and 10% for testing using fivefold cross-validation. At least two university students (both young males) were the primary subjects for the dataset. The subjects used the apartment as their residence.

Fig. 8 shows sources of IR noise in the dataset environment. A stove that is cooling down, with certain parts of the stove frame emitting temperature values similar to that of human skin (albeit momentarily), the window blinds warming up due to sunlight, and a chair that remains warm even after use, all exemplify the highly noisy nature of the environment.

B. Accuracy Results

The proposed method claims to achieve superior accuracy and execution time when compared to already proposed state-of-the-art sensor nodes network-based occupancy detection methods, i.e., EKF and PF [5]. We chose an EKF implementation [8] to be a baseline method for comparison because it is a Gaussian approximation method and in fact a special case of

BF with linear, quadratic, and Gaussian assumptions. Moreover, EKF has been compared to BF frequently in [40], [41], and [42] in terms of algorithmic efficiency. This study aims to establish that while the proposed BF method bears similarities to EKF yet is more accurate and efficient compared to existing implementations. We also evaluate the impact of the number of sensor nodes on the occupancy detection accuracy. The goal of this evaluation is to prove the efficacy of the proposed method while keeping the cost and infrastructure footprint limited. We thus present the accuracy results that compare the performance of one-node, three-node, and five-node networks. The results of this comparative experiment are listed in Table II. The node subsets in Table II are selected to maximize coverage in the case of a 3,5-node combination. For the one-node case, the entrance node was selected as it maximizes the information about the occupancy of the apartment. Table III shows the comparison of the average accuracy of the proposed BF to baseline methods. It also compares the average execution time for processing a single occupancy estimate by the proposed and the baseline methods. The execution time includes the run times for signal preprocessing, feature extraction, LSTM inference, sensor model query, and filter update and prediction steps. The proposed model error is broken down and assigned to known error sources (ESs) as illustrated in Fig. 9. The error is also broken down into false positives (FPs) and FNs, and Fig. 10 illustrates the average FPs and FNs reported by the proposed and each of the baseline methods.
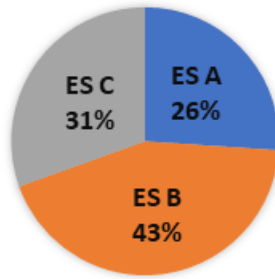
Fig. 9. Pie chart illustrating the contribution of each ES toward the FPs and FNs reported by the proposed method.

TABLE III
ACCURACY AND EXECUTION TIME COMPARISON BETWEEN BASELINE AND PROPOSED METHODS

Method	Accuracy (%)	Avg Execution time (ms)
EKF	76.52±5.01%	87±1.15
PF	84.29±4.38%	698±122.02
$X^1|X^2|X^3|X^4|X^5$	68.36±9.73%	14±0.0
BF	92.04±4.40%	59±1.35

TABLE IV
IMPLEMENTATION COMPARISON BETWEEN BASELINE AND PROPOSED METHODS

Metrics	EKF	PF	**BF**
Response time (scenario change)	45.7sec±3.79%	298.5sec±6.12%	43.5sec ±5.61%
Accuracy (high sensor noise)	63.74±7.81%	81.82±9.26%	88.13±9.02%
Accuracy (low sensor noise)	84.01±4.75%	87.59±5.22%	97.48±3.37%
Memory use	87 ±0.35% KB RAM	752±14.27% KB RAM	53 ±0.41% KB RAM

C. Results Discussion

The presented results in Section VI-D underscore some key points. First, BF consistently shows superior performance both in terms of accuracy and execution time when compared to EKF [8] and PF [5] implementations. It is important to highlight that each of these implementations has unique strengths and weaknesses and cannot be considered representative of the whole class of EKF or PF implementations. While the PF implementation is similar to the proposed BF implementation, it is important to highlight some important features of the EKF implementation. The EKF implementation is a real-time EKF-based networked sensor-based occupancy estimation algorithm. This system originally estimated the number of occupants in each room of the monitored building. We modified the EKF system to estimate the binary occupancy and use its output to compare with our proposed BF output. This EKF system handles the nonlinearity in the occupancy detection data by placing certain constraints on the model like placing upper bounds and lower bounds on exit/entrance rates, placing upper bounds on occupant flow from one room to another, and conservation principle on the number of people in the building.

Table II highlights an important aspect of the proposed BF system, i.e., scaling down to a smaller number of sensor nodes. If a sensor node needs to be dropped from the system, the proposed method considers the observations from such sensor nodes as unoccupied. Any node that has an unoccupied status does not determine the present state of the system, e.g., for the three-node scenario, the status of nodes X^1, X^4 is set to unoccupied permanently. Hence, these nodes do not contribute toward the states of the system by design.

The higher accuracy of the proposed BF demonstrated in Table III is primarily because of the sensor model deployed in the proposed method. The sensor model was able to integrate

the internode and entrance proximity information into the Bayesian updates rendering the proposed method simple yet effective enough to surpass an inherently superior PF-based method. The execution time results for BF are not very different from EKF, and both methods use effectively similar steps to reach an estimate. Matrix inversion remains the most computationally expensive step for EKF while the transition model matrix model evaluation is the most time-consuming phase for the proposed BF method. It should be mentioned here that the transition model matrix is only evaluated once for each node deployment configuration.

The row for the method titled "$X^1|X^2|X^3|X^4|X^5$" in Table III denotes the performance results for the system level output where the algorithm is simply a union of binary occupancy status from each node. This binary status is generated by the ML thresholding algorithm for each sensor node. For this system-level union algorithm, it can be observed that each of the baseline estimation algorithms (EKF, PF, and BF) is significantly more accurate than this simple rule-based method.

Table IV highlights the application-specific implementation issues encountered during the method evaluation phase. Metrics like root-mean-square error (RMSE) are not suitable for occupancy detection performance comparison while the estimator's response to high noise levels and sudden changes in the occupancy scenarios is critical. It can be observed in Table IV that the proposed BF-based method performs optimally in terms of memory usage and robustness to high-noise metrics. PF is superior in terms of low-accuracy drop as an occupancy scenario change happens between two activities, e.g., sleeping (low-noise) to cooking (high-noise).

The execution time listed in Table III is based on the Raspberry Pi 4 board. Since memory usage is a critical parameter in any IoT application, we tested the platforms listed in Table IV for memory use in the IoT platforms listed in Table V. We also found the maximum power draw a result of the execution of the proposed and baseline network-level estimation methods.

D. Error Breakdown

The ESs illustrated in Fig. 9 were determined by attributing the errors to the location and time of certain IR anomalies like IR noise due to cooking, warm water in the sink, or IR
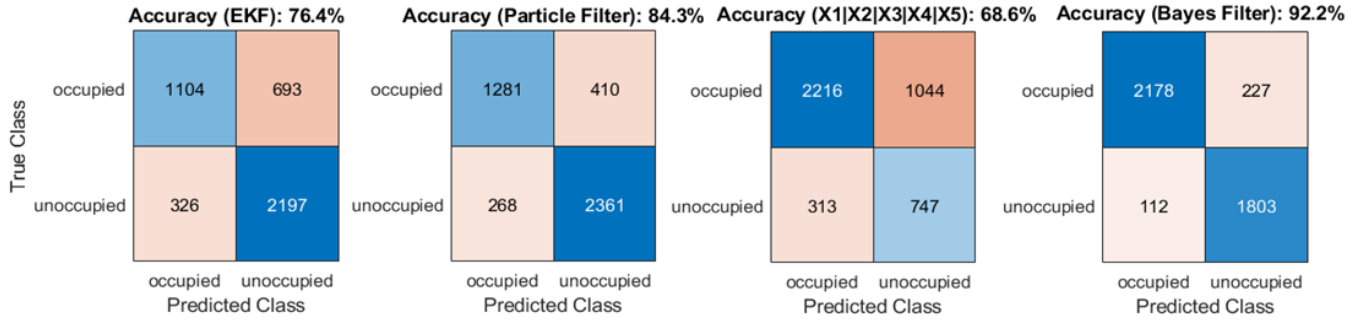
Fig. 10. Confusion matrices showing the performance comparison between the EKF, PF, simple node output union algorithm, and proposed BF method.

TABLE V
POWER CONSUMPTION COMPARISON BETWEEN BASELINE AND PROPOSED METHODS

Platform Name	Avg Power Consumption	Pros	Cons
Raspberry Pi 4	600 mA @ 5V	- 4 GB RAM - 1.5 GHz Quad Core processor	- Large form factor - Power consumption - No ADC support - Cost: $35
SparkFun Edge Development Board	6 µA @ 3.6V	- BLE 5 capable - ADC support	- 384 KB RAM - 48 MHz CPU - Cost: $16.50
Arduino Nano 33BLE Sense	2 mA @ 5V	- BLE 5 capable - ADC support	- 256 KB RAM - 64 MHz CPU - Cost: $22.30
STMicroelectronics 32F746G	178 mA @ 5V	- ADC Support	- 340 KB RAM - ARM Cortex M7 - No BLE - Cost: $56.23

shielding effect [43] due to blanket or extra layers of clothing covering the entire length of the body. The error categories are as follows.

1) *ES A:* FP caused due to the IR noise in the environment. Sources include: a) stove, hot utensils; b) warm water tap; c) laptops, chargers, and other electronics that can warm up; d) space heater; and e) objects that retain body heat after human contact.

2) *ES B:* FN caused due to low human IR radiation reaching the sensor. This happens either due to IR-blocking materials like blankets shielding human subjects or due to sensors failing to detect human IR radiation due to limited FoV.

3) *ES C:* Errors caused by wrong assumptions in network-level estimation algorithms.

We can observe in Fig. 9 that error category B dominates the pie chart as our subjects spend a significant amount of time sleeping covered in blankets during their sleep hours.

VII. CONCLUSION

We present a BF-based occupancy detection method that detects both stationary and moving subject occupancy using a network of SLEEPIR sensor nodes. Our claims of installing a minimal number of SLEEPIR nodes in suboptimal

configuration to achieve high accuracy occupancy detection are proven by the results that are evaluated over a long-term dataset that spans over 30 days. Results indicate an average 23.68% occupancy accuracy improvement when compared to the accuracy state delivered by individual SLEEPIR nodes. Results also indicate an average 7.74% occupancy accuracy improvement when compared to the accuracy state determined by the previously proposed PF-based occupancy estimation algorithm. Furthermore, the proposed BF-based method is shown to be faster by orders of magnitude when compared to a competing PF-based occupancy detection implementation.

Code Repository:

The code for C++ implementation for the proposed method is available at the following repository:

https://github.com/em22ad/system_level_occupancy_det

REFERENCES

[1] 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.

[2] 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.

[3] L. Wu, Z. Chen, and Y. Wang, "Occupancy detection using a temperature-sensitive adaptive algorithm," *IEEE Sensors Lett.*, vol. 5, no. 12, pp. 1–4, Dec. 2021, doi: 10.1109/LSENS.2021.3126285.

[4] 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.

[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. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach* (Pearson Series in Artificial Intelligence), 4th ed. Hoboken, NJ, USA: Pearson, 2021.

[7] I. Bradley and R. L. Meek, *Matrices and Society: Matrix Algebra and Its Applications in the Social Sciences*. Princeton, NJ, USA: Princeton Univ. Press, 2014.

[8] T. Miyazaki and Y. Kasama, "Multiple human tracking using binary infrared sensors," *Sensors*, vol. 15, no. 6, pp. 13459–13476, Jun. 2015.

[9] (2021). *Occupancy Sensor Market: Global Industry Trends, Share, Size, Growth, Opportunity and Forecast 2022–2027*. [Online]. Available: https://www.imarcgroup.com/occupancy-sensor-market

[10] M. Emad-Ud-Din and Y. Wang, "Indoor occupancy sensing via networked nodes (2012–2022): A review," *Future Internet*, vol. 15, no. 3, p. 116, Mar. 2023, doi: 10.3390/fi15030116.

[11] L. Romeo, R. Marani, A. Petitti, A. Milella, T. D'Orazio, and G. Cicirelli, "Image-based mobility assessment in elderly people from low-cost systems of cameras: A skeletal dataset for experimental evaluations," in *Ad-Hoc, Mobile, and Wireless Networks*. Cham, Switzerland: Springer, 2020, pp. 125–130.

[12] T. Labeodan, C. De Bakker, A. Rosemann, and W. Zeiler, "On the application of wireless sensors and actuators network in existing buildings for occupancy detection and occupancy-driven lighting control," *Energy Buildings*, vol. 127, pp. 75–83, Sep. 2016, doi: 10.1016/j.enbuild.2016.05.077.

[13] 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.

[14] S. Narayana, R. V. Prasad, V. S. Rao, T. V. Prabhakar, S. S. Kowshik, and M. S. Iyer, "PIR sensors: Characterization and novel localization technique," presented at the 14th Int. Conf. Inf. Process. Sensor Netw., Seattle, WA, USA, 2015.

[15] H. Pulkkinen, "Improving energy efficiency with occupant tracking," M.S. thesis, Dept. Appl. Phys., Aalto Univ., Espoo, Finland, 2016.

[16] R. Rabiee and J. Karlsson, "Multi-Bernoulli tracking approach for occupancy monitoring of smart buildings using low-resolution infrared sensor array," *Remote Sens.*, vol. 13, no. 16, p. 3127, Aug. 2021, doi: 10.3390/rs13163127.

[17] A. P. Singh, V. Jain, S. Chaudhari, F. A. Kraemer, S. Werner, and V. Garg, "Machine learning-based occupancy estimation using multivariate sensor nodes," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2018, pp. 1–6, doi: 10.1109/GLOCOMW.2018.8644432.

[18] 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.

[19] C. Wang, J. Jiang, T. Roth, C. Nguyen, Y. Liu, and H. Lee, "Integrated sensor data processing for occupancy detection in residential buildings," *Energy Buildings*, vol. 237, Apr. 2021, Art. no. 110810, doi: 10.1016/j.enbuild.2021.110810.

[20] D. L. Tryon, "Bayes' network and smart sensors—Occupancy detection," Ph.D. dissertation, Dept. Construct. Eng., Univ. Nebraska Lincoln, Ann Arbor, MI, USA, 2020.

[21] 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.

[22] C. Jiang, Z. Chen, R. Su, M. K. Masood, and Y. C. Soh, "Bayesian filtering for building occupancy estimation from carbon dioxide concentration," *Energy Buildings*, vol. 206, Jan. 2020, Art. no. 109566, doi: 10.1016/j.enbuild.2019.109566.

[23] P. Brémaud, "Non-homogeneous Markov Chains," in *Markov Chains: Gibbs Fields, Monte Carlo Simulation and Queues*, P. Brémaud, Ed. Cham, Switzerland: Springer, 2020, pp. 399–422.

[24] Z. Zhang and J. C. Moore, "Autoregressive moving average models," in *Mathematical and Physical Fundamentals of Climate Change*, Z. Zhang and J. C. Moore, Eds. Boston, MA, USA: Elsevier, 2015, ch. 8, pp. 239–290.

[25] Y. Mishura and M. Zili, "Mixed fractional and mixed sub-fractional Brownian motions," in *Stochastic Analysis of Mixed Fractional Gaussian Processes*, Y. Mishura and M. Zili, Eds. Amsterdam, The Netherlands: Elsevier, 2018, ch. 3, pp. 75–154.

[26] Z. Chen, "Bayesian filtering: From Kalman filters to particle filters, and beyond," *Statistics*, vol. 182, pp. 1–69, Jan. 2003, doi: 10.1080/02331880309257.

[27] Z. Chen, Q. Zhu, M. K. Masood, and Y. C. Soh, "Environmental sensors-based occupancy estimation in buildings via IHMM-MLR," *IEEE Trans. Ind. Informat.*, vol. 13, no. 5, pp. 2184–2193, Oct. 2017, doi: 10.1109/TII.2017.2668444.

[28] B. Dong et al., "An information technology enabled sustainability testbed (ITEST) for occupancy detection through an environmental sensing network," *Energy Buildings*, vol. 42, no. 7, pp. 1038–1046, Jul. 2010, doi: 10.1016/j.enbuild.2010.01.016.

[29] Z. Chen, J. Xu, and Y. C. Soh, "Modeling regular occupancy in commercial buildings using stochastic models," *Energy Buildings*, vol. 103, pp. 216–223, Sep. 2015, doi: 10.1016/j.enbuild.2015.06.009.

[30] J. Page, D. Robinson, N. Morel, and J.-L. Scartezzini, "A generalised stochastic model for the simulation of occupant presence," *Energy Buildings*, vol. 40, no. 2, pp. 83–98, Jan. 2008, doi: 10.1016/j.enbuild.2007.01.018.

[31] C. Wang, D. Yan, and Y. Jiang, "A novel approach for building occupancy simulation," *Building Simul.*, vol. 4, no. 2, pp. 149–167, Jun. 2011, doi: 10.1007/s12273-011-0044-5.

[32] A. Berchtold, "Markov chain computation for homogeneous and non-homogeneous data: MARCH 1.1 users guide," *J. Stat. Softw.*, vol. 6, no. 3, pp. 1–81, 2001, doi: 10.18637/jss.v006.i03.

[33] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," 2015, *arXiv:1506.00019*.

[34] Z. Chen and Y. Wang, "Remote recognition of in-bed postures using a thermopile array sensor with machine learning," *IEEE Sensors J.*, vol. 21, no. 9, pp. 10428–10436, May 2021, doi: 10.1109/JSEN.2021.3059681.

[35] J. Suto, S. Oniga, and P. P. Sitar, "Feature analysis to human activity recognition," *Int. J. Comput. Commun. Control*, vol. 12, no. 1, pp. 116–130, 2017.

[36] R. Hua and Y. Wang, "Robust foot motion recognition using stride detection and weak supervision-based fast labeling," *IEEE Sensors J.*, vol. 21, no. 14, pp. 16245–16255, Jul. 2021, doi: 10.1109/JSEN.2021.3075151.

[37] N. Y. Hammerla, S. Halloran, and T. Plöetz, "Deep, convolutional, and recurrent models for human activity recognition using wearables," 2016, *arXiv:1604.08880*.

[38] J. Scott et al., "PreHeat: Controlling home heating using occupancy prediction," in *Proc. 13th Int. Conf. Ubiquitous Comput.*, Sep. 2011, pp. 281–290.

[39] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics* (Intelligent Robotics and Autonomous Agents). Cambridge, MA, USA: MIT Press, 2005.

[40] N. Bellotto and H. Hu, "Computationally efficient solutions for tracking people with a mobile robot: An experimental evaluation of Bayesian filters," *Auto. Robots*, vol. 28, no. 4, pp. 425–438, May 2010, doi: 10.1007/s10514-009-9167-2.

[41] M. F. Bugallo, S. Xu, and P. M. Djurić, "Performance comparison of EKF and particle filtering methods for maneuvering targets," *Digit. Signal Process.*, vol. 17, no. 4, pp. 774–786, Jul. 2007, doi: 10.1016/j.dsp.2006.10.001.

[42] C. H. Tong and T. D. Barfoot, "A comparison of the EKF, SPKF, and the Bayes filter for landmark-based localization," in *Proc. Can. Conf. Comput. Robot Vis.*, May/Jun. 2010, pp. 199–206, doi: 10.1109/CRV.2010.33.

[43] 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, p. 32, Apr. 2020, doi: 10.1038/s41427-020-0213-z.

Muhammad Emad-ud-din received the B.S. degree in software engineering from Foundation University, 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 Fulbright Scholarship.

Zhangjie Chen received the B.S. degree from Shanghai Jiao Tong University, Shanghai, China, in 2014, the M.S. degree in mechanical engineering from the University of Florida, Gainesville, FL, USA, in 2016, and the Ph.D. degree in mechanical engineering from Texas A&M University, College Station, TX, USA, in 2021.

His research interests include occupancy sensing, activity recognition, control and actuation, and machine learning.

Qijie Shen received the B.S. degree in electronic engineering from Nanjing University, Nanjing, Jiangsu, China, in 2016, and the M.S. degree in computer engineering from Texas A&M University, College Station, TX, USA, in 2021.

He is currently a Senior Engineer with Qualcomm Technologies, Inc., San Diego, CA, USA. His recent duties focus on 5G, DSP, and firmware development.

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 the 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, 63 journal articles, and 40 conference proceeding papers, and holds 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.

Libo Wu received the B.S. degree in applied physic from the University of Science and Technology of China, Hefei, Anhui, China, in 2015, and the Ph.D. degree in mechanical engineering from Texas A&M University, College Station, TX, USA, in 2021.

His research interests include pyroelectrical sensors, thermopile sensors, and their applications for occupancy detection in indoor environment.