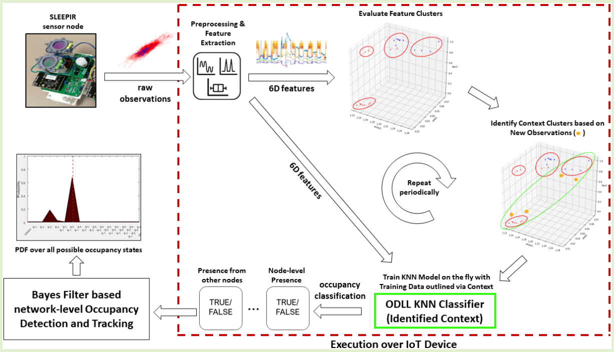
# Context-Aided Occupancy Detection and Tracking Using Networked SLEEPIR Sensors

Muhammad Emad-ud-din and Ya Wang, *Member, IEEE*

***Abstract*—Accurate occupancy detection remains a challenging problem due to dynamic occupancy patterns and varying environments. Traditional machine learning (ML) struggles with this variability as models typically require large datasets and frequent updates as occupancy scenarios are unlimited and continuously change over time. It is thus infeasible to train a single "universal" model for the diverse real-world scenarios given real-world computational constraints. To address these issues, a context-aware hierarchical classification framework is proposed which periodically trains multiple occupancy classifiers on subsets of data delineated by meaningful contexts. When new occupancy data arrive, its context is identified, and a corresponding pretrained classifier is selected for prediction. By focusing each model on more consistent data distributions defined by context, this approach aims to improve classification accuracy compared to baselines trained on static datasets alone. The framework also aims to eliminate the need for offline training on large datasets and frequent over-the-cloud model updates required by traditional ML approaches by performing ML-based training and inference directly on the sensor node via an Internet-of-Things (IoTs) device. The framework is evaluated via datasets collected both in an office and a residential setting, monitored by a network of synchronized low-energy electronically chopped passive infra-red (SLEEPIR) sensors. These sensors, unlike conventional passive infrared (PIR) sensors, can detect stationary occupants. Time-series features are extracted from observations and clustered to discover underlying contextual scenarios. Experimentation resulted in context scenarios which essentially represent varying levels of infrared (IR) noise in observed environment. The proposed framework achieved a 5.03% accuracy improvement over the best baseline algorithm.**


SLEEPIR sensor node
raw observations
Preprocessing & Feature Extraction
6D features
Evaluate Feature Clusters
Identify Context Clusters based on New Observations ( ● )
Repeat periodically
6D features
Train KNN Model on the fly with Training Data outlined via Context
ODLL KNN Classifier (Identified Context)
occupancy classification
Node-level Presence
TRUE/ FALSE
...
Presence from other nodes
TRUE/ FALSE
Bayes Filter based network-level Occupancy Detection and Tracking
PDF over all possible occupancy states
Execution over IoT Device

***Index Terms*— Bayes filter, classification, context-aided classification, *K*-nearest neighbor (KNN) classification, occupancy detection and tracking, on-device learning, passive Infrared (PIR) sensors.**

## I. Introduction

THE problem of occupancy detection is inherently complex as occupancy estimation experiences considerable accuracy deterioration [1], [2] due to constantly evolving environmental and occupancy scenarios. Due to the dynamic nature of occupancy scenarios, it is virtually impossible to collect a comprehensive training dataset that contains patterns encompassing all anticipated occupancy scenarios. Although highly unlikely to exist, such a dataset would also require significant computational power to train a machine learning (ML) model on, due to its size. For the same reason, the ML models are typically trained off-site, and model updates are pushed to the inference engine, which requires over-the-cloud connectivity for the occupancy sensors. For such models, a novel input occupancy pattern that does not belong to the distribution of the training dataset would cause degradation in occupancy detection accuracy. Since occupancy tracking is a higher-order property of occupancy detection [3], it also suffers a loss in tracking accuracy. The overall occupancy detection challenge is a well-investigated topic [4] with widespread applications in health and safety [5], [6], [7], smart energy management [8], [9], [10], heating, ventilation, and air conditioning (HVAC) [11], [12], [13], and security [14], [15] sectors

Manuscript received 17 June 2024; revised 11 August 2024; accepted 20 August 2024. Date of publication 11 September 2024; date of current version 31 October 2024. 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 National Science Foundation (NSF) under Award 2341560. The associate editor coordinating the review of this article and approving it for publication was Prof. Jungyoon Kim. *(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, College Station, TX 77843 USA, also with the Department of Electrical and Computer Engineering, College Station, TX 77843 USA, and also with 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.2024.3452554

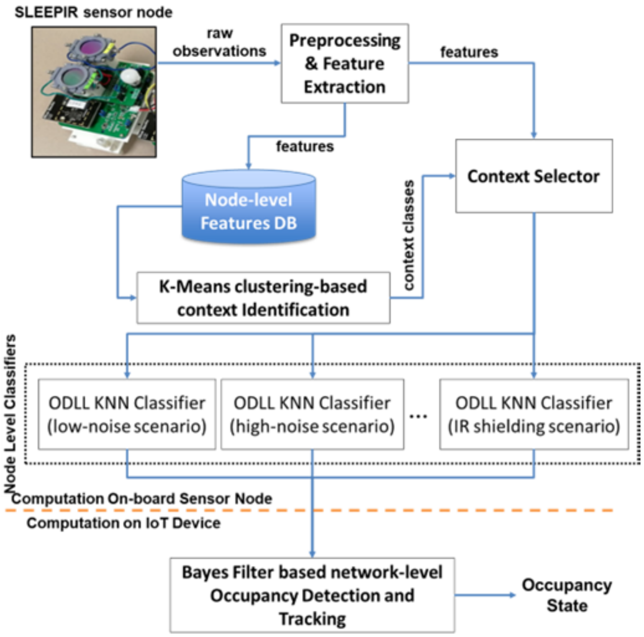

SLEEPIR sensor node
raw observations
Preprocessing & Feature Extraction
features
features
Context Selector
Node-level Features DB
context classes
K-Means clustering-based context Identification
Node Level Classifiers
ODLL KNN Classifier (low-noise scenario)
ODLL KNN Classifier (high-noise scenario)
...
ODLL KNN Classifier (IR shielding scenario)
Computation On-board Sensor Node
Computation on IoT Device
Bayes Filter based network-level Occupancy Detection and Tracking
Occupancy State

Fig. 1. Context-aided occupancy detection and tracking system flowchart. Hand-crafted features are extracted from raw observations from each of the sensor nodes within the system. These features are stored in a limited-term DB. This DB is evaluated for feature clusters based on DBI. Clusters are then assigned to context class via 1-to-many relationship. These classes correspond to various occupancy scenarios. A context selector module then selects a suitably trained ODLL occupancy classifier that establishes occupancy state (presence/no presence). A Bayes filter-based occupancy detection and tracking systems then considers occupancy outputs from all nodes in the system to establish a room-level occupancy detection and tracking estimate every 60 s.

To address occupancy's dynamic and ever-changing nature, on-device lifelong learning (OODL) algorithms [16] aim to maintain test and training distribution similarity by continuously updating datasets. However, these algorithms have memory constraints and limited dataset sizes, as Internet-of-Things (IoTs) and edge AI devices are typically resource-constrained [17]. The proposed method converts the bounded size handicap into an opportunity by using contextual information to create bounded datasets that effectively limit the classification space for algorithms like $K$-nearest neighbor (KNN). A limited classification space guarantees superior classification performance, given certain preconditions are met [18]. We term contextual information as the *context* of occupancy, which can be any information aiding occupancy estimation.

Fig. 1 illustrates the proposed method's overview. Hand-crafted features are extracted from normalized sensor observations and stored in a bounded database (DB). Feature clusters are identified and mapped to context classes representing occupancy scenarios. For new observations, a *context selector* determines the context class. A corresponding periodically trained OODL KNN [16] model is then selected for node-level occupancy determination. Finally, occupancy outputs and node locations are provided to a Bayesian filter (BF) algorithm [19] that fuses the node-level occupancy to generate a system-level estimate for occupancy detection and tracking over time. An in-depth comparison with state-of-the-art occupancy detection methods was provided in a recent study by Emad-ud-din and Wang [20]. This review evaluates the contemporary techniques across key performance indicators such as accuracy, response time, and practical deployment considerations. This work compares the proposed method to the best-performing state-of-the-art methods featured in the review study under discussion.

Steps 2–4 constitute the hierarchical classifier selection (HCS) framework. The key contributions for the proposed work include 1) a unique context-aided hierarchical classification approach for occupancy detection and tracking; 2) context improves accuracy by focusing classification on similar distributions compared to baseline algorithms where a large static dataset is used; 3) a BF algorithm provides robust system-level occupancy state detection and tracking; and 4) The framework eliminates overhead of offline training on large datasets and over-the-cloud ML model updates.

For the remaining article, a literature review is presented in Section II. Section III details the underlying networked sensor nodes used in the dataset collection and describes the steps involved in preprocessing sensor inputs and preparing these for clustering and classification methods. Section IV outlines the significant features and working principles of the HCS framework. Section IV describes the network-level BF algorithm that estimates the system-level occupancy and tracks occupancy. Section V outlines the dataset collection strategy and lists the method performance results. Section VI presents a brief discussion of the method. Section VII presents a conclusion to this article.

## II. Literature Review

While the proposed method can be applied to occupancy features extracted from other sensor modalities, passive infrared (PIR)-based sensors have been the most widely used for occupancy detection due to their low cost [21]. Other sensor modalities explored for occupancy detection include light, temperature, sound, $CO_2$, reed switches, total volatile organic compounds (TVOCs), pressure, humidity, power usage, Bluetooth low energy (BLE), and Wi-Fi sensors [20]. However, the broad adoption of these alternative sensors has been limited due to problems such as slow response times, high noise-to-signal ratios, low sensing resolution, and low correlation with occupancy, given the changes in the indoor environment. Low-cost camera-based systems have also been suggested [22], but lack of privacy and high computational costs have inhibited their widespread use. Issues with alternative sensor modalities have contributed to PIR sensors remaining the primary technology for occupancy detection applications.

Context awareness has been established as crucial in developing accurate sensing and control systems [23]. It has already been shown that splitting a dataset based on certain information, such as context, can improve the ML model performance [18], [24]. In [25], various statistical classification models such as linear discriminant analysis (LDA), classification and regression trees (CART), and random forest (RF) are employed for time-series data from light, temperature, humidity, and $CO_2$ sensors. It is shown in the article that by including the information related to the time of the day and week status (weekend, weekdays), a 32% increase in occupancy detection was achieved. While the accuracy peaks at 97% for the work described in [25], it lacks the variety of occupancy scenarios in the dataset. For example, the study indicates whether the room door opened or closed impacted the different sensor readings. Despite this observation, no other occupancy scenarios were tested for an indoor office environment chosen for the study, which can have many possible occupancy scenarios.

The approach, described in [26], is more elaborate than any other work encountered during the survey conducted for this literature review. This approach uses three hierarchical levels of classification. The first level is for occupancy detection, the second is for occupancy density detection, and the third is for head count estimation. Each level provides context for the next layer. The computer usage and meeting schedule provide the context for the first level of the observed office space. Each level outputs a feature vector that contains posterior occupancy probabilities for classification at the next level. KNN and support vector machines (SVM) are used for classification at each level. Then, their results are compared. One of the apparent issues with this approach is the assumption that computer usage and meeting schedules are correlated. The computer can be accessed remotely, while meetings can be skipped. Another issue with this approach is that the context categories, namely, “computer usage,” “meeting schedules,” “occupancy,” “occupancy density,” and “headcount,” are all hard coded in the algorithm. The proposed method, however, is data-driven, and context categories are defined by clusters or, in other words, are data-driven.

Khan et al. [26] corroborates the concerns raised in the introduction of this article that occupancy is dynamic and challenging to predict due to potentially infinite possible scenarios. The occupancy estimation system must also be aware of details on the lag caused by the powerup phase of the heating regulation systems so that the lagging times of the affected room temperature can be optimized. In [27], a context-aware occupancy prediction method is proposed. It is based on the spatiotemporal analysis of historical data and context information. The context information includes the current occupancy state and the sensor location, sensor observation correlation, season, day of the week, and time of day. The context is used to select parameters for the estimation models. This method is suitable for real-time predictive control, for example, for HVAC control. This method implements a Markov Chain model and a Semi-Markov model for occupancy prediction. The proposed work in this article does model the occupancy of the observed area using a Markov Chain model, yet no historical data are utilized in the process.

Emad-Ud-Din and Wang [16] outlines an ODLL approach to continuously update models with new data to maintain test and training distribution similarity. The proposed algorithm builds upon the ODLL algorithm [16] by incorporating contextual information handling and a multilevel classification framework to further improve accuracy for dynamic occupancy scenarios. Both the proposed algorithm and the [16] aim to address the dynamic nature of occupancy detection through continuous learning approaches. However, the proposed algorithm offers a more sophisticated solution by integrating contextual information and a HCS process on top of ODLL.

Emad-ud-din et al. [1] proposes using a particle filter (PF) algorithm for occupancy estimation with networked synchronized low-energy electronically chopped passive infra-red (SLEEPIR) sensors. It evaluates PF performance against a static KNN model. On the other hand, the proposed algorithm provides a more comprehensive evaluation of its proposed context-aided BF method against PF and different baselines like extended Kalman filter (EKF) and long short-term memory (LSTM).

Emad-Ud-Din et al. [19] presents a BF algorithm for occupancy detection using a network of SLEEPIR sensors. It processes observations from each sensor via a sensor model that considers node adjacency, updating the filter. This provides a probability density function (pdf) for occupancy across the monitored space. Evaluated on a residential dataset over 30 days, it improves accuracy over PF and extended Kalman filter baselines by fusing information from networked sensors rather than relying on individual nodes. Compared to [19] the proposed algorithm, it introduces a novel context-aided hierarchical classification framework on top of the BF to handle dynamic occupancy scenarios better. Also, [19] it does not include this contextual reasoning component.

One of the critical issues with context-aware applications is the evaluation of context within the feasible time window through timely data acquisition and efficient processing platforms [28]. The proposed method ensures that the context is evaluated every 30 min. This is achieved via a low-powered IoT device capable of performing onboard learning and inference. It is also observed that multiple works [29], [30], [31] integrate IoT-based devices to facilitate context-aware services and applications. Following the trend, the proposed method also utilizes IoT devices to evaluate and use context for on-device occupancy classification.

## III. Networked Sensor Nodes, Execution Platform, and Input Preprocessing

As per the system flowchart shown in Fig. 1, the raw sensor observations are extracted from the sensor node called the SLEEPIR occupancy sensor. The SLEEPIR sensor was recently developed to address a long-lasting issue of PIR stationary occupants [32], [33]. All the SLEEPIR sensor nodes in the system partially process the input onboard and then use a Bluetooth communication protocol to deliver the output to a central IoT device for network-level processing. This processing boundary is marked in Fig. 1. The sensor node and communication platform details are presented in Section III-A.

### A. Synergistic SLEEPIR Sensor Node

In the top figure (Fig. 2), the SLEEPIR sensor node configuration is depicted. Each node consists of two polymer-dispersed liquid crystal (PDLC) IR shutters that cover two analog PIR sensors (EKMC2691111K, Panasonic Inc). Additionally, a traditional digital PIR sensor (EKMB1391111K, Panasonic Inc.), a microcontroller unit (MCU) (EFR32BG13, Silicon Laboratories), a PDLC driving circuit, an ambient temperature sensor embedded in the MCU,

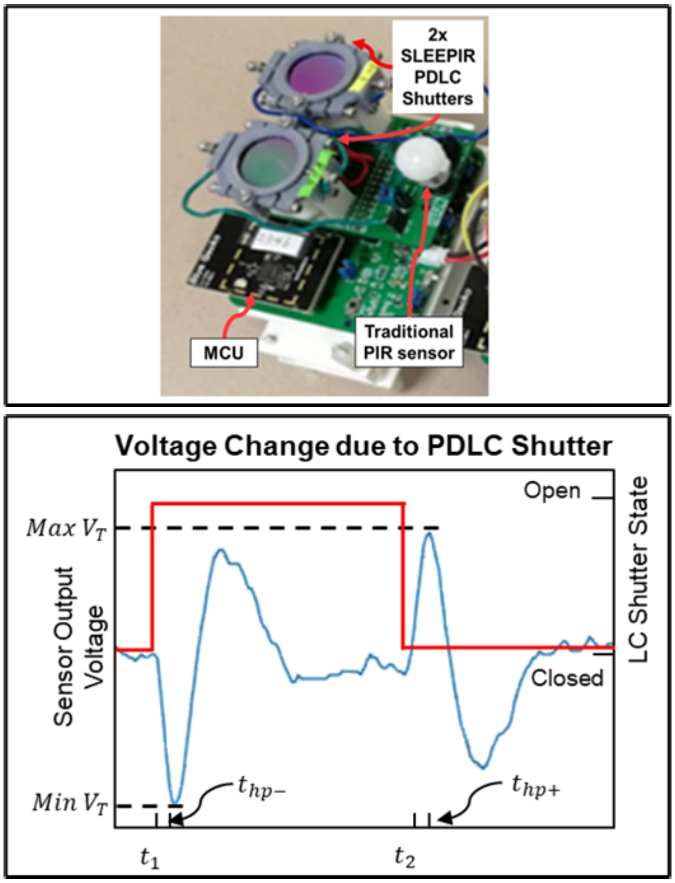


Fig. 2. SLEEPIR sensor node (top). Illustration of the sensor output voltage $V_{\text{out}}$ due to the changing transmitted IR radiation when the PDLC shutter turns on or off. $t_2 - t_1 = 4$ s. $t_3 - t_1 = 8$ s (bottom). $t_{hp+}$ is timestamp when $V_{\text{out}} == (\text{Max}\,V/\sqrt{2})$ while $t_{hp-}$ is timestamp when $V_{\text{out}} == (\text{Min}\,V\sqrt{2})$.

TABLE I
EXECUTION PLATFORM EVALUATION

| Edge AI Platform | Power to CPU Cycle ratio (Avg) | Pros | Cons |
|---|---|---|---|
| Ambiq Apollo3 SoC | 6 µA | ARM Cortex M4F, 1MB Flash, BLE/ADC support, Temp sensor, Cost : $2.97 | 384 KB RAM, 96 MHz CPU |
| Ambiq Apollo3 Blue Plus SoC | 6 µA | ARM Cortex M4F, 768 KB RAM, 2MB Flash, BLE/ADC support, Temp sensor, Cost : $3.95 | 96 MHz CPU |
| SparkFun Artemis | 6 µA | ARM Cortex M4F, 1MB Flash, BLE/ADC support | 384 KB RAM, 96 MHz CPU, Cost: $8.9 |
| ST Micro-electronics 32F746G | 178 mA | ARM Cortex M7, ADC support | 340 KB RAM, No BLE, Cost: $56.23 |
| Arduino Nano 33BLE Sense | 2 mA | ARM Cortex M4, BLE/ADC support | 256 KB RAM,64 MHz CPU, Cost: $22.30 |
| ESP32-PICO-V3-02 | 130 mA, 0.6 A | BLE/WiFi support, 2 MB RAM/4MB Flash, 240 MHz dual core CPU, Cost : $3.20 | High power consumption, Cost: $22.30 |
| Evaluation Kit for RA6M5 | Max 500 mA | 200MHz, Arm Cortex M33, 512KB SRAM, 2MB Flash | High power consumption, Cost: $57.00 |

and two AA batteries connected in series (providing a 3 V dc voltage supply) are included. The PDLC shutters cover a pyroelectric sensing element composed of pyroelectric material, which converts changes in heat flux into current. When the radiation power received by the pyroelectric material is represented as $W(t) = W_0 e^{i\omega t}$, and modulated at frequency $\omega$, the voltage response $V_{\text{out}}(t)$ for the preamplifier stage follows the following form:

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

Here, $p'$ is the perpendicular component of the pyroelectric coefficient $p$. A is the area of the sensing element. $\eta$ represents the emissivity of the sensing element; $\tau_T = H/G_T$ 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-off-the-shelf (COTS) PIR sensors typically include multiple sensing elements (usually two or four) arranged in series with opposite polarizations. Applying the same polarization to the sensing elements and covering these with the PDLC shutter leads to significant voltage signals from the PIR sensor. The PDLC shutter, positioned in front of the PIR sensor, undergoes periodic changes in transmission, resulting in a corresponding periodic change in the received radiation $W(t)$. Consequently, this change influences the output voltage $V_{\text{out}}(t)$. An illustrative example of an output signal is presented in Fig. 2 (bottom).

## B. Execution Platform

The proposed method involves an onboard computational device or an execution platform, a critical component of the execution pipeline. The computational device chosen for this purpose is an Edge AI IoT device called Ambiq Apollo3 blue plus system-on-chip (SoC). This device was chosen after a detailed evaluation study involving several resource-constrained IoT platforms. Power consumption, processing frequency, and onboard memory were major considerations during this evaluation effort. The resultant data gathered from this effort is listed in Table I. It can be observed in Table I that the chosen Edge AI device has a minimal power-to-CPU cycle ratio. The device can also house and process relatively less bulky ML algorithms like KNN, given its 768 KB static RAM. It possesses an embedded temperature sensor and Bluetooth communication module, all of which are critical for the stated needs of the proposed method. Fig. 3 highlights the crucial performance parameters plotted over 24 h for Ambiq Apollo3 Blue Plus SoC. The plot shows no memory leaks and estimates the proposed algorithm's onboard resource usage. As per long-term testing of the SLEEPIR sensor node, its average yearly power consumption is 5.55 E-03 KWh/unit/year. This number is comparable to other well-known PIR-based occupancy sensors such as Panasonic PaPIRs sensors [34] and sensor switch CM10WR [35].

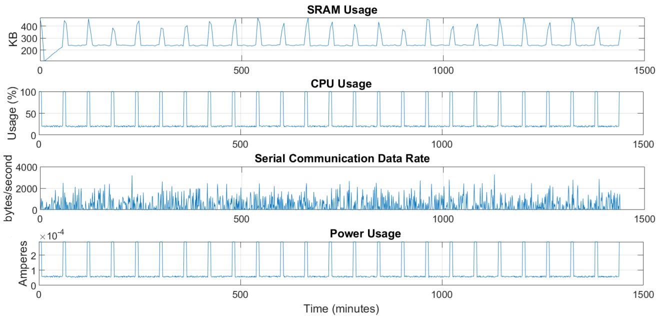


Fig. 3. Various performance metrics plotted for 24 h for Ambiq Apollo3 blue plus edge AI device.

## C. Input Pre-Processing and Feature Extraction

While zero-centering and normalization of features are typical for ML models' performance, stability, and interpretability, segmenting the input time series into windows of features requires justification. Sections III-C1 and III-C2 describe the input formatting, feature extraction, and segmentation in detail.

*1) Input Formatting:* The hand-crafted ML features are used widely in the literature to produce easily distinguishable values for different data classes [36]. 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. Moreover, an optimal feature minimizes the false positives (FPs) and false negatives (FNs) by virtue of being easily distinguishable within feature space. Features described [16] are optimal for occupancy detection as the accuracy exceeds 90% for typical occupancy scenarios using SLEEPIR sensor nodes. Table II describes each feature evaluated during the proposed work. The top six features were chosen to be deployed for the proposed ML model. Some of these features are visually represented in Fig. 2 (bottom panel).

Although input quantization has been shown to improve recursive neural network (RNN) accuracy, we decided against using this approach due to the insignificant accuracy improvement achieved at the cost of information loss [37].

It is important to highlight the effort involved in evaluating several features before finalizing them. Data from both occupied and unoccupied classes were gathered, preprocessed, and windowed. Afterward, this data were processed through a feature evaluation phase where several time and frequency domains and autocorrelation and wavelet features were evaluated using the data. These features were shortlisted from occupancy detection studies in Emad-Ud-Din and Wang [20] that make use of PIR signal. Table II provides the description, suitability, and $p$-value for all evaluated feature to $V_{\text{out}}(t)$ output of the SLEEPIR sensor.

While $p$-value of a feature in this context signifies the statistical significance of the difference in that feature's values between the two classes (occupied and unoccupied), it is important to note that the $V_{\text{out}}(t)$ keeps changing with changing occupancy scenario. Thus, it was ensured that the $p$-value was computed using the entirety of both datasets used in this study which captures a wide variety of occupancy scenarios.

TABLE II
EVALUATED FEATURE DESCRIPTION AND SUITABILITY TO SIGNAL

| Evaluated Features | Description (Feature computed over $win_T$) | Suitability (p-value) |
|---|---|---|
| Max $V_T$ | Maximum $V_{out}$ | $V_{out}$ presents peak values when sensor is exposed to radiated IR (0.029) |
| Min $V_T$ | Minimum $V_{out}$ | $V_{out}$ presents low values when sensor shutter is closed (0.010) |
| Half-Power Bandwidth for +ve peak (HPB+) | $t_{hp+} - t_2$ where $t_{hp+}$ is timestamp when ( $V_{out} ==$ $(Max\ V)/\sqrt{2})$ ) Minimum $V_{out}$ | $V_{out}$ rises sharply when sensor is exposed to radiated IR (0.025) |
| Half-Power Bandwidth for -ve peak (HPB-) | $t_{hp-} - t_1$ where $t_{hp-}$ is timestamp when ( $V_{out} ==$ $(Min\ V)/\sqrt{2})$ ) | $V_{out}$ drops sharply when sensor shutter is closed after exposure to radiated IR (0.027) |
| Windowed mean (mean $V_T$) | Mean $V_{out}$ | $V_{out}$ maintains higher central tendency during non-occupancy (0.040) |
| Windowed Std. Dev (std $V_T$) | Standard Deviation for $V_{out}$ | $V_{out}$ maintains high signal variability during human presence (0.0018) |
| Peak Count | Number of peaks above 0.5 std. deviation value | $V_{out}$ has less signal variability towards positive side during human presence (0.062) |
| Spectral Entropy | Measure of frequency distribution randomness | $V_{out}$ is predictable in terms of frequency during most scenarios (0.096) |
| Mean autocorrelation | Measure signal self-similarity over time | $V_{out}$ is not self-similar over-time especially when IR noises are added or ambient temperature changes (0.13) |
| Wavelet Detail Entropy | Measure of complexity or randomness at multiple scales | $V_{out}$ remains relatively less random at different scales during most scenarios (0.098) |

*2) Sliding Window Input Approach:* We initialize a base training dataset $\text{obs}_{\boldsymbol{B}}$ where each element is created by sliding a fixed-horizon nonoverlapping window of length $l$ over the raw voltage output $V_{\text{out}}$ from the SLEEPIR sensor. We thus extract a 6-D training input time-series consisting of the following elements [Max $V_t$,Min $V_t$, mean$V_t$, std$V_t$, HPB$_t$+, HPB$_t$−]. Suitable observation window length ($l$) is a critical parameter that has a pronounced impact on the over-network accuracy [38]. Detailed impact assessment of this parameter has already been accomplished in [16] and an optimal value of $l$ was found to be 60 s for typical occupancy scenarios. The labels $\text{label}_B$ are initialized where each element corresponds to each window in $\text{obs}_B$. The labeling is done as per the *automated labeling algorithm* proposed in [16]. It must be reiterated here that initially, the labels (*occupied* and *unoccupied)* are obtained from calibration data collected by the end user. The calibration is done using a smartphone app where the user labels 20 observation windows, with 10 windows labeled as "occupied" and 10 as "unoccupied" based on human presence within the field of view (FoV) of the SLEEPIR sensor node.

The algorithm utilizes the time difference between consecutive PIR activations for a sensor for automatic labeling. If the time difference is less than or equal to $l$ (60 s), it assumes the presence of a human subject within the sensor range and FoV. This assumption is generally true, and the training datasets labeled based on this assumption provide high occupancy detection accuracy [16]. It may also be mentioned here that labeling of observations is required when the output of the traditional PIR sensor onboard the SLEEPIR node is 0 for the entire observation window of length $l$. The traditional PIR sensor is assumed to be reliable enough not to produce FPs.

## IV. HCS Framework

Determining an occupancy classification model context involves clustering the observation features based on Euclidean distance. Each feature cluster approximately corresponds to a specific occupancy scenario or activity performed by subjects. For example, clusters may form when the observed subject is in bed. Such a cluster can be associated with a context labeled "subject sleeping." This context outlines a cluster containing observations of subjects under blankets at night. During experimentation, it was found that an occupancy classifier trained only on this "sleeping" data subset that outperformed a baseline classifier trained on all data where the subjects performed other activities besides sleeping. Thus, the proposed method trains multiple ML models, with each trained on the observations within an identified context cluster.

When classifying new observations, the appropriate pretrained model is selected based on the observation's context cluster. This approach leverages context to focus models on more consistent data distributions, improving classification accuracy over baselines.

The purpose of the HCS framework described below is to train and select the most accurate occupancy classifier among the set of continuously trained classifiers given a context. Here, context is the information that is pivotal in selecting the optimal occupancy classifier—Section IV-A details how the context is evaluated.

### A. Context Generation Through Data Clustering

Fig. 1 shows that the node-level feature DB contains a labeled base training dataset with labels. Clustering aims to identify the subsets of the base training dataset that enables more accurate occupancy classification. Fig. 4 explains why an appropriately clustered feature space is easier to classify than an unclustered one. The concept of using clustering to improve classification has been investigated thoroughly [18], [39]. This begs the following questions: 1) How do feature clusters look like for an actual SLEEPIR sensor signal? 2) Can the feature data be clustered in a meaningful way so that it facilitates classification? Fig. 5 answers these questions by showing that the evaluated occupancy features, when clustered, can be easily classified into occupied and unoccupied classes. Fig. 5 also indicates that the raw observation clusters overlap and thus have a significantly higher susceptibility to FPs and FNs.

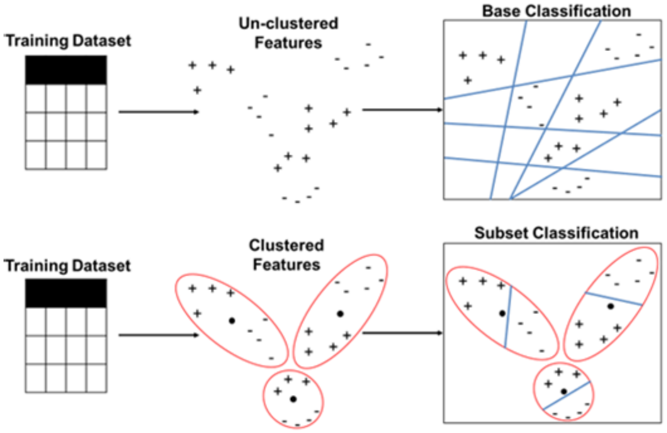


Fig. 4. Unclustered features being classified for occupancy (+) and nonoccupancy (−) (top). The clustered features are classified (bottom). The classification is much simpler in case only a single cluster is considered at a time. The caveat is that the clustering needs to be meaningful and have minimal outliers.

To evaluate meaningful clusters, that is, clusters that correspond to an occupancy scenario, we utilize a clustering technique based on the $K$-means algorithm. Apart from identifying the clusters attributed to each scenario, this technique is also designed to minimize the variance in clustering solutions obtained from $K$-means in the presence of randomized centroid initialization. The Davies–Bouldin index (DBI) [40] was utilized to reduce the impact of random initialization on the $K$-means clustering results by tuning the number of clusters $k$ in a manner less sensitive to the initial cluster centroids. To be precise, the DBI was calculated for different values of $k$ as $K$-means was executed multiple times with randomly initialized centroids. The DBI seeks to minimize the ratio of within-cluster to between-cluster distance. By plotting the DBI against $k$ for each run, the value of $k$ at which the DBI curve plateaus or reaches a minimum was identified. This $k$ value corresponded to a clustering solution demonstrated to be more robust to initialization, although the individual cluster assignments differed slightly between runs. It was observed from the experiments conducted that each context naturally tended to generate clusters with low intracluster distance and high intercluster distance.

It must be highlighted here that one occupancy scenario can cause multiple feature clusters. A single KNN-based subclassifier is trained with features contained within a single scenario. Multiple occupancy scenarios are denoted by varying colors in Fig. 6. A 1-to-1 correspondence is shown between the scenarios and trained subclassifiers in Fig. 6 as well.

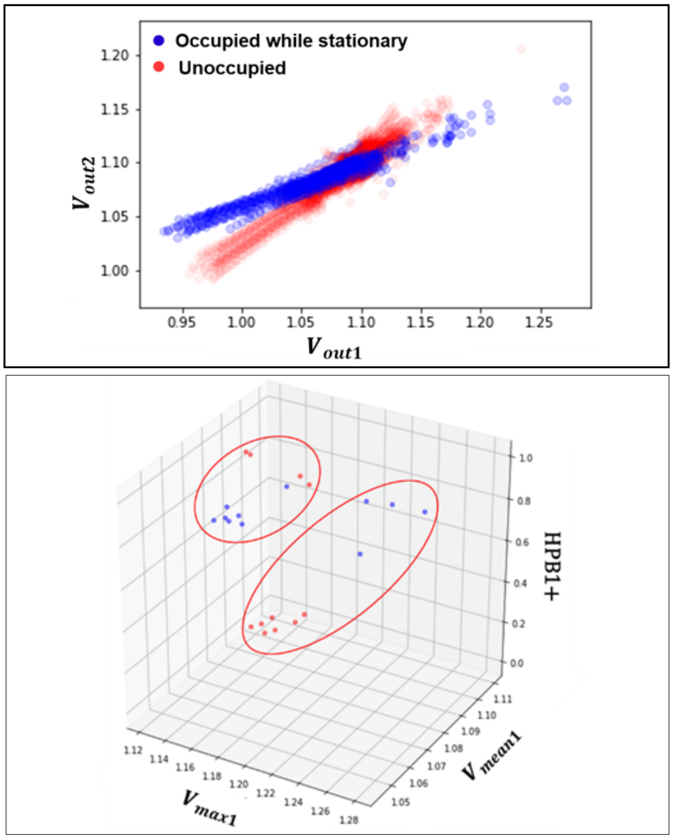


Fig. 5. Raw observations from the SLEEPIR sensor node for a 24-hr period (top). Clustered features for SLEEPIR sensor module 1, evaluated from the automatically labeled raw Fig. 1 (bottom). Context-aided occupancy detection and tracking system flowchart. Hand-crafted features are extracted from raw observations for each of the sensor nodes within the system. These features are stored in a limited-term DB. This DB is evaluated for feature clusters based on DBI. Clusters are then assigned to context class via 1-to-many relationship. These classes correspond to various occupancy scenarios. A context selector module then selects a *suitably* trained ODLL occupancy classifier that establishes occupancy state (presence/no presence). A Bayes filter-based occupancy detection and tracking system then considers occupancy outputs from all nodes in the system to establish a room-level occupancy detection and tracking estimate every 60 s.

## B. Sub-classifier Architecture and Training

The node-level or subclassifier forms the second layer of the HCS framework, as shown in Fig. 1. These classifiers are ODLL classifiers having training datasets that evolve as the occupancy scenarios consistently change in almost every real indoor setting. The training dataset for each classifier is an assigned cluster from the context generation phase detailed in Section IV-A. For example, while the subject sleeps, the SLEEPIR observations are frequently clustered together for the collected dataset. Thus, these clustered observations can serve as the training dataset for a dedicated ODLL KNN classifier in this subclassifier layer of the HCS framework.

Similarly, the subject working on a laptop produces feature clusters that are distinct from the "sleep scenario cluster." The "sleep scenario cluster" and "laptop work scenario" clusters may be close in terms of Euclidean or Cosine distance which may cause misclassifications. When both scenarios are evaluated for occupancy by two separate classifiers, each trained on its respective cluster, the results show marked improvement. The architecture for these subclassifiers is inspired by the similar KNN classifier deployed in [16].

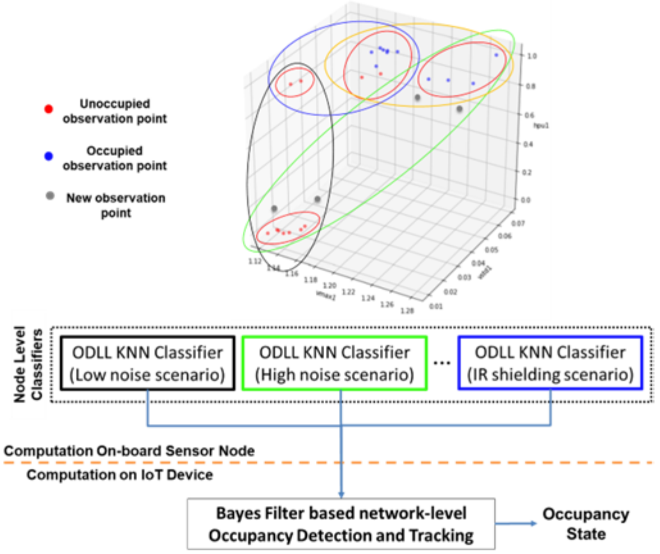

Unoccupied observation point
Occupied observation point
New observation point
Node Level Classifiers
ODLL KNN Classifier (Low noise scenario)
ODLL KNN Classifier (High noise scenario)
...
ODLL KNN Classifier (IR shielding scenario)
Computation On-board Sensor Node
Computation on IoT Device
Bayes Filter based network-level Occupancy Detection and Tracking
Occupancy State

Fig. 6. 6-D Features points (shown in 3-D) divided into clusters (in red) via DBI. Based on observations from past 30 min (in gray), closest clusters are chosen (in green) to form a context. A dedicated subclassifier is pretrained with feature points contained in green ellipsoid for occupancy classification.

In general, KNN is a supervised classification technique that operates on the principles of nonlinear distance-based analysis. Unlike other methods, KNN doesn't involve a learning process but relies on direct classification. It requires the indexed storage of the entire training dataset.

Given a training dataset that contains a cluster $C_1$ $(x_{C_1}, y_{C_1})$, and a new observation $x_{\text{new}}$, the distance is calculated, denoted as $d_m$, between $x_{\text{new}}$ and $x_{C_1}$ using the following equation:

$$d_m = \left\| x_{\text{new}} - x_{C_1} \right\|. \tag{2}$$

Distance calculation typically employs the widely used Euclidean distance measure. Once the distance $d_m$ is obtained, the labels of the $k$ training samples with the smallest distances are selected. A majority voting scheme is then applied to determine the label of the new observation. As the number of existing samples in the dataset increases, the computation time for assigning a new sample to a class also increases [41].

A limit is set on the total number of observations in the training dataset (as determined by the cluster) to keep the size of the training set bounded. This is achieved by periodically removing observations that are farthest from their respective cluster centroid, determined by the Euclidean distance. To determine the optimal number of neighbors ($k$), a critical parameter for KNN inference the Elbow search method [42] is used to determine the optimal number of neighbors ($k$), a critical parameter for KNN inference. This method involves periodically calculating the within-cluster-sum of squared errors (WSS) for different values of $k$ neighbors and evaluating the WSS. While it is true that the plot of WSS versus $k$ can be visually assessed for an elbow. A numerical evaluation route was undertaken to determine the elbow point for the sake of process automation. This was done by calculating

**Algorithm 1** Context Selector

```
Input:
feature observations $x_{test}$, base clusters list $c_B$, thresh, $N$
Output:
cluster subset $c_S$
{$ContextSelectorThread$}
while $last\_time\_stampcurrent\_time < 30min$
 do_nothing( )
$last\_time\_stamp = current\_time$
$total\_clusters = \boldsymbol{length}(c_s)$
$vote\_vector[1, \ldots, total\_clusters] = 0$
for all $x_{text}$
  for all $k$ in $c_B$
    $clust\_cent =$ cluster_centroid($k$);
      if dist($x_{text}, clust\_cent) < thresh$
          $vote\_vector[k] = vote\_vector[k] + 1;$
 $c_s =$ return_topNvoted_clusters($vote\_vector, N$)
```

the second derivative of the WSS plot. The elbow point was determined to have the largest negative value of the second derivative. This search is performed periodically rather than for every inference. Depending on the context, a subclassifier is chosen, which establishes the node-level occupancy based on the new observation $x_{\text{new}}$ collected at the last timestamp. The occupancy output of the selected subclassifier then reaches a network-level BF occupancy detection and tracking algorithm, which is described in Section V.

### C. Context Selector

Among the subclassifier layer of ODLL KNN models, a suitable model must be selected to perform classification whenever a *context* is evaluated. This is achieved via a KNN-based Context Selector. It must be reiterated here that the evaluated *context* in Section IV-A is a cluster(s) based on similar occupancy observation. In other words, each cluster is formed due to the SLEEPIR sensor response to a particular occupancy scenario.

The only remaining task now is to be able to choose a relevant set of clusters dictated by the occupancy scenario. The distribution of observations in the selected set of clusters should match the distribution of observations in the incoming observations from the sensor node. For this purpose, algorithm 1 is proposed. Algorithm 1 receives a vector $x_{\text{test}}$ of features calculated over sensor observations accumulated over a tunable predefined period, for example, 30 min. For each observation in $x_{\text{test}}$ that falls within a threshold distance of a cluster contained in a list of clusters $c_B$, evaluated in Section IV-A, a vote is added to a vector whose elements correspond to each element of $c_B$. Top $N$ clusters regarding vote count are selected and returned as a list $c_S$, as an output of Algorithm 1. At this point, it takes a negligible amount of time to train a subclassifier with all the feature points that are members of clusters in list $c_S$. The subclassifier can be pretrained as well, but this would involve pretraining subclassifiers for all possible combinations of clusters identified in Section IV-A. Next, the trained subclassifier is used to establish occupancy for the next 30 min (or a predefined time interval).

This step concludes the last step of the top layer of the proposed HCS framework, as shown in Fig. 1. In short, the *context* determines a subdataset based upon which a trained ODLL KNN sub-classifier performs occupancy classification for the new observation and outputs the occupancy status for each node in binary.

## V. Bayes Filter-Based Occupancy Detection and Tracking

The BF occupancy detection and tracking algorithm determines the networkwide occupancy while tracking the occupancy state of the observed area, that is, the occupancy status for each sensor node in the system. After binarizing the node-level observations using the proposed HCS architecture in Section IV, these observations are utilized to update a BF [19], which generates an estimate of occupancy at the network level.

Based on the available information, this BF method provides a real-time posterior pdf of the state (occupancy belief). The BF method is considered "optimal" because it seeks the posterior distribution that integrates and incorporates all the available information expressed as probabilities [22]. The BF-based algorithm for detecting occupancy at the network level employs a dual-stage algorithm, details of which can be found in [19]. In the initial stage, the occupancy status of the area under surveillance is modeled as a Markov decision process (MDP). The MDP is a representation of real-world dynamics. The MDP proposes an optimal policy—a sequence of state transitions needed before reaching a goal state.

The goal state is the occupancy state detected by the networked SLEEPIR sensor nodes, for example, $X^2$, $X^3$ which indicates that occupancy was detected at nodes $X^2$ and $X^3$. The MDP also needs a starting state, the previous occupancy state detected by the BF-based method. 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 $\pi$ that need to be navigated to reach goal state while beginning at the start state. This suggested sequence $\pi$ would also require an expected time to be navigated. Using fundamental Markov analysis (FMA) technique [43], the sequence of occupancy states and expected traversal time, given a starting and a goal occupancy state, can be evaluated. As this detail is not the primary focus of the research presented in this article, the evaluation process is skipped for the sake of conciseness. In the second stage, the BF consistently receives updates about occupancy status from individual SLEEPIR nodes, e.g., $X^2$, $X^3$. Based upon the suggested sequence of occupancy states $\pi$, 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 $\pi$. If an incoming occupancy observation aligns well with $\pi$ suggested by the MDP, it is assigned a higher likelihood than an observation that doesn't align well. The sensor model for the BF correlates the incoming observations to the likelihood values of the overall space being observed as occupied (or unoccupied). The posterior pdf, which represents the present probability for all possible occupancy states, is provided in Fig. 7. Fig. 7 shows how the occupancy belief of the

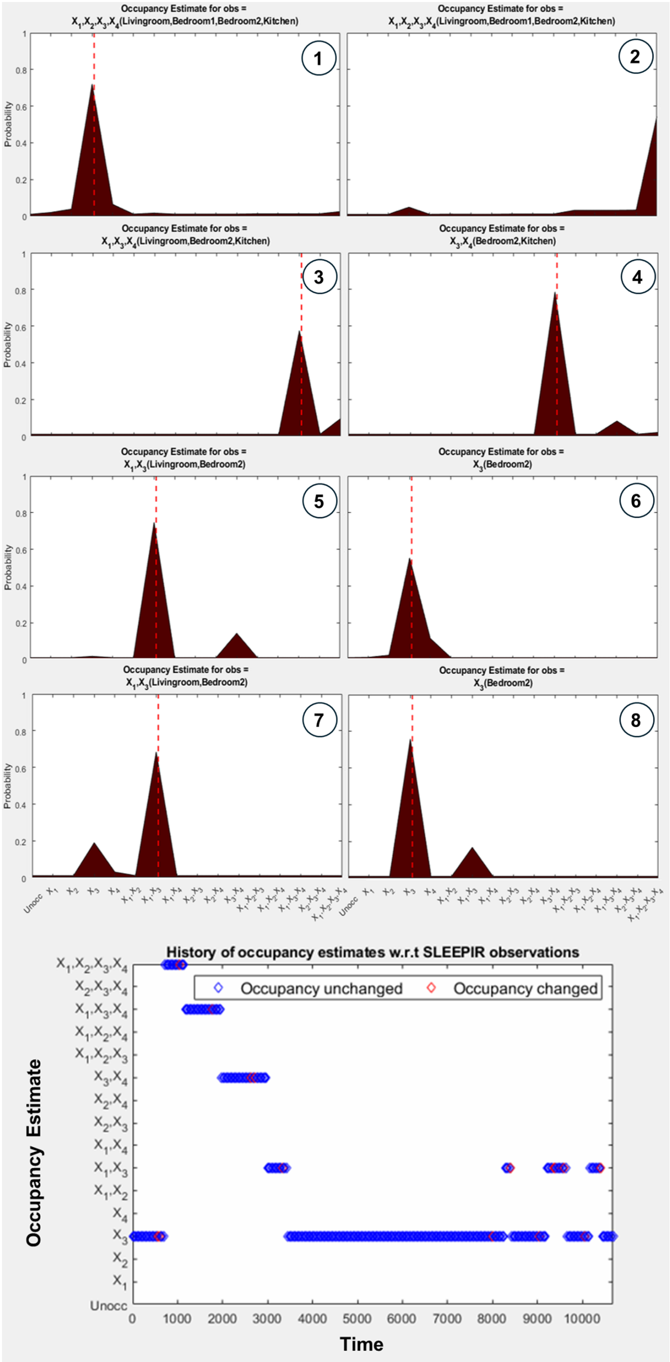


Fig. 7. Insets 1 to 6 show progression of BF belief about the occupancy via posterior probability distribution over all possible occupancy states (top). A time-chart that shows the history of BF belief over time (bottom). Inset 1 represents system belief at time = 0, while inset 8 represents system belief at time = 10 862.

BF-based occupancy-tracking algorithm evolved over time. Fig. 7 also shows the history of tracked beliefs about the occupancy state. The temporal progress chart comprising the system's occupancy belief over time is also shown in Fig. 7.

## VI. Results

### A. Datasets

To evaluate the adaptability of the proposed context-aided ODLL occupancy detection algorithm across different occupancy scenarios, such as residential settings and office spaces, two datasets were collected: one in a residential unit and the other in an office space.

*1) Residential Dataset:* Four SLEEPIR sensor nodes were deployed at a residential apartment, as shown in Fig. 8. The apartment unit has two bedrooms and two bathrooms. The apartment-covered area is 140 m$^2$. Each node is installed at the height of 2.8 m. Each node collects one observation every 30 s. The duration of data collection is 30 days. Webcams are used to collect ground truth data. The Kitchen sensor node ($X^4$) encounters frequent IR noise due to the presence of a stove and warm water tap within sensor FoV. The living room sensor node ($X^1$) and Bedroom 1 sensor node ($X^2$) also encounter IR noise due to sunlight encroaching from the windows.

*2) Office Dataset:* Three SLEEPIR sensor nodes were deployed in an office space, as shown in Fig. 9. Certain areas are inaccessible, so these are shown to be crossed out in the figure. This space has three rooms and two bathrooms. The office-covered area is 103.3 m$^2$. Each node is installed at the height of 2.8 m. In this dataset as well, each node collects one observation every 30 s. The duration of data collection is 15 days. Again, webcams are used to collect ground truth data. There are no windows in this office space, so IR noise due to sunlight does not impact accuracy. However, the environment

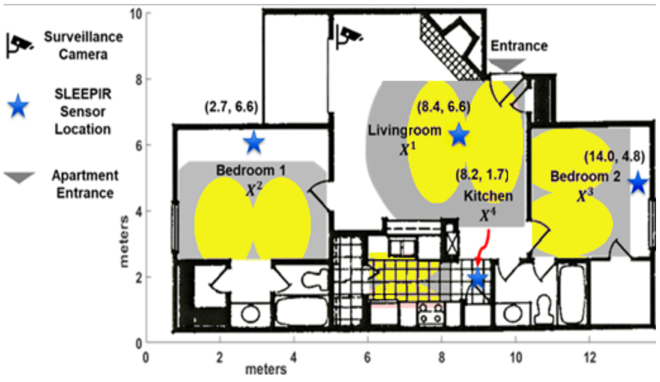


Fig. 8. Dataset was collected at a two bed, two bath apartment. Four SLEEPIR sensor nodes were for dataset collection. Sensor node locations are shown in the figure as well. Yellow shade shows SLEEPIR detection footprint while gray shade represents traditional PIR detection footprint.

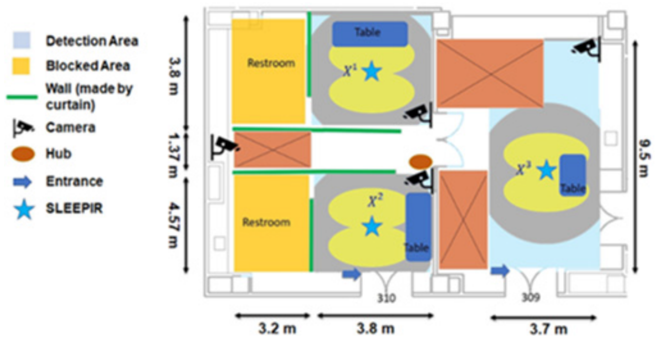


Fig. 9. Dataset was collected at an office space consisting of three rooms. Three SLEEPIR sensor nodes were deployed for dataset collection. Sensor node locations are shown in the figure. Yellow shade shows SLEEPIR detection footprint, while gray shade represents traditional PIR detection footprint.

is filled with electronic office equipment, including LED/liquid crystal display (LCD) screens, PCs, and laptops, etc. We used 7 out of 15 days to extract probabilities of human traffic between various office rooms. For both datasets, 80% of the data for node-level detection was used for training and validation, while the remaining 20% was used for testing. The accuracy analysis presented in Section VI-B is based on sensors placed in an uncontrolled environment where IR noise is encountered frequently, thus representing the method's performance in challenging environments.

### B. Accuracy Results and Analysis

This study chooses state-of-the-art EKF and PF as baseline comparison methods. The proposed BF method outperformed these in terms of occupancy detection and tracking using networked sensor nodes.

It is essential to state that the proposed and baseline algorithms are compared for accuracy at the two levels described below.

1) Comparison between sensor fusion algorithms, that is, EKF, PF, and proposed BF algorithms. The results of this comparative experiment for the residential scenario are listed in Tables III and IV. The corresponding results for the office scenario are presented in Tables V and VI.
2) Comparison between the LSTM [44] and KNN models trained on static, nonevolving occupancy dataset versus the context-aided KNN model trained on bounded evolving occupancy dataset.

Among the fusion baseline algorithms, the EKF implementation [45] was selected for comparison due to its ability to approximate a Gaussian distribution and its resemblance to BF, which incorporates linear, quadratic, and Gaussian assumptions. In previous literature, EKF has frequently been compared to BF in terms of algorithmic efficiency [46], [47], [48]. The rationale behind choosing PF [1] as a baseline method was simple as it has been used previously with networked SLEEPIR sensor data to report detection accuracy. Although inherently robust to non-Gaussian noise, PF is a computationally expensive choice. Moreover, the PF implementation [1] uses historical sensor data to model occupancy behavior. Inhomogeneous hidden Markov models (IHMM) [49], although a good candidate for a baseline algorithm, were not chosen as a baseline algorithm because the method fused sensor modality by integrating data from infrared (IR) sensors to other environmental parameters such as temperature, humidity, and carbon dioxide levels. This modality-based sensor fusion approach differs from the proposed BF approach that integrates a single modality data from multiple nodes to construct a comprehensive occupancy hypothesis. A comprehensive review study [20] encompasses the state-of-the-art in in terms of fusion algorithms for occupancy detection. It comments on the input modalities, datasets, subjects, algorithm details, and intended purpose of such algorithms. Since the proposed method deals with a unique sensor signal and claims success within a specific category of occupancy, that is, stationary occupancy, the review study suggests EKF, BF, and PF-based techniques to be the most suitable for performance comparison.

TABLE III
RESIDENTIAL SPACE DETECTION ACCURACY COMPARISON TO BASELINE MODELS

| Date | EKF + Static LSTM Accuracy | PF + Static LSTM Accuracy | BF + Static LSTM Accuracy | BF + Static KNN Accuracy | Proposed Context-aided KNN + BF Accuracy |
|---|---|---|---|---|---|
| 15 April | 77.2% | 81.9% | 83.1% | 89.3% | 95.9% |
| 16 April | 82.1% | 89.2% | 91.7% | 94.0% | 95.3% |
| 17 April | 70.5% | 77.0% | 81.4% | 90.7% | 97.6% |
| 18 April | 61.2% | 88.5% | 82.5% | 86.8% | 91.0% |
| 19 April | 59.0% | 88.3% | 86.8% | 87.1% | 95.1% |
| 20 April | 71.6% | 82.5% | 89.5% | 91.2% | 97.5% |
| 21 April | 80.2% | 85.2% | 91.6% | 93.9% | 97.8% |

TABLE IV
RESIDENTIAL SPACE TRACKING ACCURACY COMPARISON TO BASELINE MODELS

| Date | EKF + Static LSTM Accuracy | PF + Static LSTM Accuracy | BF + Static LSTM Accuracy | BF + Static KNN Accuracy | Proposed Context-aided KNN + BF Accuracy |
|---|---|---|---|---|---|
| 15 April | 57.9% | 70.5% | 77.5% | 86.9% | 93.2% |
| 16 April | 63.5% | 77.9% | 87.1% | 91.7% | 92.4% |
| 17 April | 52.1% | 69.4% | 75.8% | 88.0% | 95.4% |
| 18 April | 48.5% | 77.7% | 78.5% | 84.8% | 85.1% |
| 19 April | 44.0% | 79.7% | 81.7% | 86.7% | 91.0% |
| 20 April | 53.2% | 75.2% | 84.5% | 90.5% | 93.9% |
| 21 April | 62.9% | 78.2% | 88.4% | 92.2% | 94.0% |

TABLE V
OFFICE SPACE DETECTION ACCURACY COMPARISON TO BASELINE MODELS

| Date | EKF + Static LSTM Accuracy | PF + Static LSTM Accuracy | BF + Static LSTM Accuracy | BF + Static KNN Accuracy | Proposed Context-aided KNN + BF Accuracy |
|---|---|---|---|---|---|
| 6 Mar | 86.15% | 94.22% | 93.71% | 90.10% | 98.22% |
| 20 Oct | 84.33% | 92.36% | 94.82% | 93.39% | 96.50% |
| 3 Nov | 89.70% | 91.72% | 93.29% | 91.04% | 92.90% |

This study aims to demonstrate that the proposed context-aided BF method surpasses existing implementations in accuracy and efficiency despite its similarities with EKF and PF. This evaluation also focuses on establishing the effectiveness of the proposed method in accurately tracking occupancy at the room or zone level within a building. First, the accuracy results that compare the occupancy detection performance using the networked SLEEPIR nodes are presented. The results of this comparative experiment are listed in Tables III and V for both office and residential scenarios.

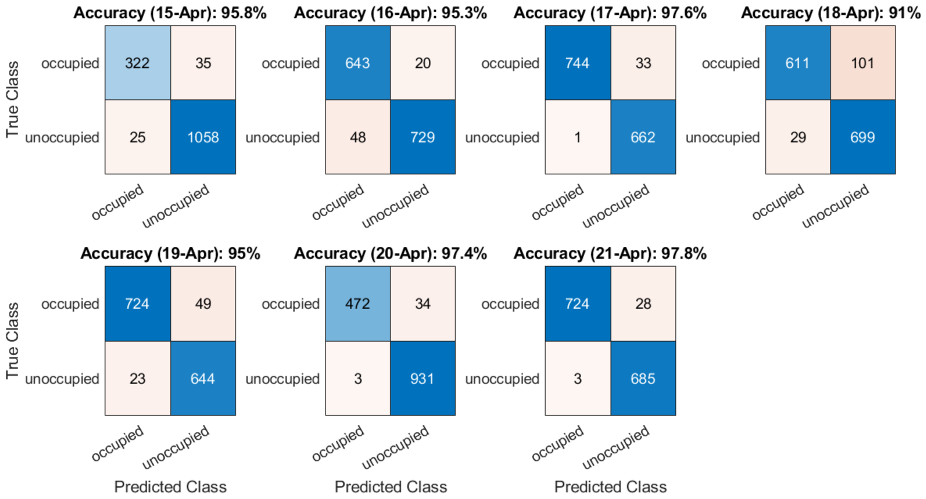


Fig. 10. FPs and FNs for 7-day test data collected in residential space showing the performance for the proposed context-aided occupancy detection framework.

TABLE VI
OFFICE SPACE TRACKING ACCURACY COMPARISON TO BASELINE MODELS

| Date | EKF + Static LSTM Accuracy | PF + Static LSTM Accuracy | BF + Static LSTM Accuracy | BF + Static KNN Accuracy | Proposed Context-aided KNN + BF Accuracy |
|---|---|---|---|---|---|
| 6 Mar | 53.60% | 77.51% | 81.80% | 77.14% | 95.23% |
| 20 Oct | 46.33% | 78.20% | 78.41.% | 76.29% | 97.52% |
| 3 Nov | 60.20% | 79.54% | 85.20% | 79.11% | 94.01% |

The occupancy tracking performance of the abovementioned baseline algorithms with the proposed context-aided BF tracking algorithm is also compared in Tables IV and VI. We chose to evaluate the performance of algorithms over each day as the clusters associated with the context evolve due to the observance of new data (as explained in Section IV-B). This entails that the order of the incoming observations has an impact on how clusters evolved and thus influences accuracy. Thus, the progression of accuracy over time needs to be shown. The proposed model error is broken down into FPs and FNs, and Fig. 10 illustrates the FP and FN reported by the proposed and each baseline method. A total of 77% of the dataset was used for training, while the remaining 23% was used for testing. It was ensured that a continuous whole week of data was tested for performance evaluation.

The following points highlight the most notable observations about the reported results.

1) On the 18 April, it was observed that there was a high FN rate across all models, with the lowest accuracy being 61.2% for the EKF + Static LSTM model and the highest being 91.0% for the proposed context aided KNN + BF model. This unusually high FN rate could be attributed to the subject spending most of the day in bed. This nontypical behavior might have misled the models, causing them to incorrectly predict the absence of the subject and hence increase the FN rate. Additionally, as subjects are susceptible to covering themselves with blankets or comforters when in bed [50], this behavior can result in significant IR shielding. Such IR shielding has significantly contributed to the anomalously low model accuracy on this day through an increase in FNs.
2) On the 17, 20, and 21 April, it is noted that there were fewer FPs than usual across all models. This means that the models performed better in accurately predicting occupancy. A critical factor that helped in this improved performance is the capability of the proposed context-aided KNN + BF model to adapt well to the IR noise caused by the solar IR in the living room and bedroom 1. The model minimized FPs, thereby increasing its accuracy, which ranged from 97.6% to 97.8% on these dates.
3) It is noteworthy that on 3 November, the detection accuracy for the static LSTM model was higher than that for the proposed context-aided ODLL KNN model. Static models, when tested on data that follow a similar distribution to the distribution of their static training dataset, can produce better results than an ODLL model. While such circumstances can occur, the chances of occurrence are very low.
4) It needs to be mentioned that the presented accuracies for the proposed KNN + BF model are based on the fact that the context was evaluated using 30-min observation windows, as detailed in Section IV-A. The 30-min window is a tunable parameter. It was found experimentally that considering context over less than 30 min resulted in nonconsistent context, while notching up the value to over 30 min led to missing certain short-term contexts like having lunch or cooking a small meal, etc.
5) The office and residential space differ in terms of human occupancy behavior. For example, in the office space, the sensor is exposed to frequent and large human motion, for example, employees walking across the FoV of the sensor. In the residential environment, the subjects were found to be static for long periods of time, for example, during sleep. Subjects were also found to be covered mostly during sleep (IR-shielding)

that introduces FN in terms of occupancy detection. We find that average-tracking accuracy for office space is 95.59% which is higher than average-tracking accuracy for residential setting, that is, 92.14%. This is a direct consequence of losing track of subjects due to IR-shielding.

## VII. Discussion

It must be highlighted that no prior assumptions are made about the clusters that are essentially the context classes, as evaluated in Section IV-A. Instead, a data-driven approach, that is, $K$-means clustering, is solely used to establish similar occupancy patterns. As per the conclusions gathered from the literature review [25], it was expected that the datasets gathered for this research would indicate that the context classes determined by the $K$-means clustering algorithm would loosely coincide with specified periods of a typical week, for example, weekday mornings/evenings where the occupant activity pattern was found to be similar. On the contrary, the clustering technique mentioned in Section IV-A did not produce clusters corresponding to a specific period within the week. Instead, it was found that each cluster predominantly contained observations from a frequent activity carried out in a typical week, such as sleeping, eating at the table, cooking, and exercising. Yet, some similar observations from multiple activities also crossed over into a cluster of other similar activities, for example, sleeping and watching TV in bed looks similar to SLEEPIR. This adapts well to node-level ODLL KNN subclassifier architecture, as these classifiers are blind to the underlying meaning of these clusters. The classification accuracy for these subclassifiers is agnostic to the fact that the clusters represent a time period or not. This is because the basis for classification is similarity in the occupancy patterns that may belong to any period of a typical week. Thus, similar-looking occupancy patterns within two different time frames within a week can be part of the same cluster depending upon their distance to another cluster within the feature space.

One of the most critical sets of clusters was found to be the clusters formed due to the weekday night activities. These clusters are unique in that the subject(s) are primarily stationary(sleeping), and during the months with frequent cold nights, most parts of the subject's body are covered with a blanket or fabric. This inhibits the emitted body IR radiation from reaching the sensor, resulting in IR shielding [50]. Thus, unique patterns form clusters in proximity to clusters representing unoccupancy within the feature space. It would have been a particularly difficult task to perform occupancy classification for the nonclustered data for the cold nights where subjects are mostly covered in blankets, yet due to the clustering algorithm, both cold night features and unoccupancy features lie within the same scenario, and a specialized subclassifier was trained to distinguish between these two occupancy phenomena.

## VIII. Future Research

One of the major strengths of the proposed context-aided ODLL occupancy detection algorithm is its ability to adapt to varying occupancy scenarios (e.g., changes in occupant behavior, occupant numbers, or observed space, i.e., residence or office). This ability is well investigated in [16] and is due to the inherent nature of ODLL, which periodically updates the model according to new occupancy scenarios. The proposed addition of context generation and BF fusion algorithm on top of the previously proposed ODLL model enables the algorithm to avoid the FP and FN caused by overlapping scenario clusters that existed in the previously proposed ODLL model [16].

The context also resolves the issue of FP caused by heat sources in the residence, such as heating ducts, kettle, warm water from tap, stove, and oven. The main takeaway of the proposed approach is that a human presence without any heat source will be plotted at a different location in the feature space shown in Fig. 6, compared to when the sensor observes the heat sources alongside the human subject. It can be observed in Fig. 6 that the blue feature points representing the occupancy cluster are divided into two different scenarios based upon the generated context boundaries, that is, IR-shielding (blue boundary) and high-noise scenario (green boundary). Since a distinct model handles each scenario, the classification is much cleaner and with less FP and FN.

Previously published studies [1], [16], [19], [20] have thoroughly assessed the impact of increasing and decreasing the number of the SLEEPIR sensor nodes on occupancy detection accuracy. Thus, this question was not addressed in this study. These studies determined that a node density of 364 ft$^2$ is sufficient to achieve reliable detection accuracy for the residential indoor spaces used in the variety of datasets used in these studies. It is helpful to highlight here that each SLEEPER node can observe an area of 364 ft$^2$ when installed at the ceiling height of 2.8 m, directly observing the area under it. It is acknowledged the results may not generalize to all deployment sizes; however, given the prior work establishing node count effects, the choice of a four-node setup for residential and a three-node setup for office space was based on experimental data provided by previous studies. It is also acknowledged that since the periodic updates to the model happen to cater to the new occupancy scenarios, the frequency for these updates, which is set at every 30 min in algorithm 1, and the extent for these updates is determined by the variable *thresh* in algorithm 1 are good avenues for future research. Presently, 96 MHz CPU frequency of the IoT platform places an upper limit on both these parameters. It is expected that with the maturity of future IoT platforms, this upper limit can be enhanced. Power consumption limitations also play a critical role in determining the upper limit under question. For example, adding trivial components, such as logic-level shifters that enable intermicrocontroller communication in certain scenarios, can significantly add to the overall power draw. Thus, future efforts can attempt to minimize the impact of hardware power on the expected performance of the proposed algorithm.

## IX. Conclusion

This work presented a context-aided occupancy detection and tracking framework for networked sensor nodes. The article addresses the inherent complexity of occupancy detection due to evolving environmental and occupancy scenarios. It emphasized the challenges of collecting comprehensive training datasets encompassing all anticipated occupancy patterns and the need for model updates in dynamic scenarios.

The proposed framework leveraged the bounded size of the training dataset and utilized contextual information to enhance occupancy detection accuracy. The method effectively limited the classification space through data clustering and HCS framework, improving accuracy and execution time compared to baseline algorithms. The BF-based occupancy detection and tracking algorithm provided a robust occupancy estimation at both network and node levels, incorporating the occupancy state's real-time posterior pdf. The performance was evaluated by collecting datasets in both residential and office space settings.

The article demonstrated the effectiveness of the proposed method through a comprehensive evaluation using a dataset collected from a residential apartment. The results showcased superior performance compared to state-of-the-art methods regarding occupancy detection accuracy and tracking precision. The lowest detection accuracy was recorded at 61.2% for the EKF + Static LSTM model, while the proposed Context-aided KNN + BF model achieved the highest accuracy of 98.22%

The contributions of this research include the novel context-aided hierarchical classification approach, the BF-based occupancy detection and tracking algorithm, and the elimination of offline training and over-the-cloud model updates. The proposed framework offers potential applications in building automation, energy management, and occupancy-based services.

## References

[1] 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.

[2] 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.

[3] B. Feagin, M. E. Poplawski, and J. T. Day. (2020). *A Review of Existing Test Methods for Occupancy Sensors*. [Online]. Available: https://www.osti.gov/biblio/1668746

[4] 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.

[5] G. Alfian, M. Syafrudin, M. F. Ijaz, M. A. Syaekhoni, N. L. Fitriyani, and J. Rhee, "A personalized healthcare monitoring system for diabetic patients by utilizing BLE-based sensors and real-time data processing," *Sensors*, vol. 18, no. 7, p. 2183, Jul. 2018.

[6] S. Azimi and W. O'Brien, "Fit-for-purpose: Measuring occupancy to support commercial building operations: A review," *Building Environ.*, vol. 212, Mar. 2022, Art. no. 108767.

[7] B. Gunay, Z. Nagy, C. Miller, M. Ouf, and B. Dong, "Using occupant-centric control for commercial HVAC systems," *ASHRAE J. Article*, vol. 63, p. 30, Jun. 2021.

[8] Z. D. Tekler, R. Low, C. Yuen, and L. Blessing, "Plug-mate: An IoT-based occupancy-driven plug load management system in smart buildings," *Building Environ.*, vol. 223, Sep. 2022, Art. no. 109472.

[9] *Building Energy Efficiency Standards for Residential and Nonresidential Buildings: For the 2022 Building Energy Efficiency Standards Title 24, Part 6, and Associated Administrative Regulations in Part 1*, Standard CEC-400-2022-010, 2022.

[10] *Standard for Health Care Facilities*, document NFPA 99, 2021.

[11] Z. D. Tekler, R. Low, and L. Blessing, "An alternative approach to monitor occupancy using Bluetooth low energy technology in an office environment," *J. Phys. Conf. Ser.*, vol. 1343, no. 1, Nov. 2019, Art. no. 012116, doi: 10.1088/1742-6596/1343/1/012116.

[12] C. Int. Code. (2021). *International Building Code*. [Online]. Available: https://search.library.wisc.edu/catalog/999907647602121

[13] I. Rodriguez, M. Lauridsen, G. Vasluianu, A. N. Poulsen, and P. Mogensen, "The gigantium smart city living lab: A multi-arena LoRa-based testbed," in *Proc. 15th Int. Symp. Wireless Commun. Syst. (ISWCS)*, Aug. 2018, pp. 1–6, doi: 10.1109/ISWCS.2018.8491077.

[14] A. Filippoupolitis, W. Oliff, and G. Loukas, "Bluetooth low energy based occupancy detection for emergency management," in *Proc. 15th Int. Conf. Ubiquitous Comput. Commun. Int. Symp. Cyberspace Secur. (IUCC-CSS)*, Dec. 2016, pp. 31–38, doi: 10.1109/IUCC-CSS.2016.013.

[15] 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.

[16] M. Emad-Ud-Din and Y. Wang, "Promoting occupancy detection accuracy using on-device lifelong learning," *IEEE Sensors J.*, vol. 23, no. 9, pp. 9595–9606, May 2023, doi: 10.1109/JSEN.2023.3260062.

[17] A. N. Sayed, F. Bensaali, Y. Himeur, and M. Houchati, "Edge-based real-time occupancy detection system through a non-intrusive sensing system," *Energies*, vol. 16, no. 5, p. 2388, Mar. 2023.

[18] H. Wecker, A. Friedrich, and H. Adel, "ClusterDataSplit: Exploring challenging clustering-based data splits for model performance evaluation," in *Proc. 1st Workshop Eval. Comparison NLP Syst.*, 2020, pp. 155–163.

[19] M. Emad-Ud-Din, Z. Chen, Q. Shen, L. Wu, and Y. Wang, "Bayes filter-based occupancy detection using networked SLEEPIR sensors," *IEEE Sensors J.*, vol. 23, no. 19, pp. 22832–22844, Oct. 2023, doi: 10.1109/JSEN.2023.3304372.

[20] 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.

[21] (2022). *Occupancy Sensor Market: Global Industry Trends, Share, Size, Growth, Opportunity and Forecast 2022–2027*. [Online]. Available: https://www.imarcgroup.com/occupancy-sensor-market

[22] 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 *Proc. 19th Int. Conf. Ad-Hoc*, 2020, pp. 125–130.

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

[24] *Data Split Example*. Accessed: Aug. 30, 2024. [Online]. Available: https://developers.google.com/machine-learning/data-prep/construct/sampling-splitting/example

[25] 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 Build.*, vol. 112, pp. 28–39, Jan. 2016.

[26] A. Khan et al., "Occupancy monitoring using environmental & context sensors and a hierarchical analysis framework," presented at the 1st ACM Conf. Embedded Syst. Energy-Efficient Buildings, Memphis, TN, USA, 2014.

[27] A. A. Adamopoulou, A. M. Tryferidis, and D. K. Tzovaras, "A context-aware method for building occupancy prediction," *Energy Buildings*, vol. 110, pp. 229–244, Jan. 2016.

[28] Y. N. Malek et al., "On the use of IoT and big data technologies for real-time monitoring and data processing," *Proc. Comput. Sci.*, vol. 113, pp. 429–434, Jun. 2017.

[29] K. Akkaya, I. Guvenc, R. Aygun, N. Pala, and A. Kadri, "IoT-based occupancy monitoring techniques for energy-efficient smart buildings," in *Proc. IEEE Wireless Commun. Netw. Conf. Workshops (WCNCW)*, Mar. 2015, pp. 58–63.

[30] F. Lachhab, M. Essaaidi, M. Bakhouya, and R. Ouladsine, "Towards a context-aware platform for complex and stream event processing," in *Proc. Int. Conf. High Perform. Comput. Simul. (HPCS)*, Jul. 2016, pp. 961–966.

[31] F. Lachhab, M. Bakhouya, R. Ouladsine, and M. Essaaidi, "Performance evaluation of linked stream data processing engines for situational awareness applications," *Concurrency Comput. Pract. Exper.*, vol. 30, no. 12, Jun. 2018, Art. no. e4380.

[32] 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.

[33] L. Wu, F. Gou, S. 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.

[34] *Panasonic PaPIRs*. Accessed: Aug. 30, 2024. [Online]. Available: https://industrial.panasonic.com/ww/products/pt/papirs

[35] *CMR Series—Ceiling Mount Line Voltage Sensors*. Accessed: Aug. 30, 2024. [Online]. Available: https://www.acuitybrands.com/products/detail/757191/sensorswitch/

[36] 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.

[37] J. Suto, S. Oniga, and P. Pop Sitar, "Feature analysis to human activity recognition," *Int. J. Comput. Commun. Control*, vol. 12, no. 1, p. 116, Dec. 2016.

[38] 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.

[39] M. Piernik and T. Morzy, "A study on using data clustering for feature extraction to improve the quality of classification," *Knowl. Inf. Syst.*, vol. 63, no. 7, pp. 1771–1805, Jul. 2021.

[40] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vols. PAMI-1, no. 2, pp. 224–227, Apr. 1979.

[41] 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.

[42] 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.*, Semarang, Indonesia, 2018, pp. 533–538, doi: 10.1109/ISEMANTIC.2018.8549751.

[43] 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.

[44] 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 Arbo, MI, USA, 2021.

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

[46] 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.

[47] 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.

[48] 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.

[49] Y. Yuan, X. Li, Z. Liu, and X. Guan, "Occupancy estimation in buildings based on infrared array sensors detection," *IEEE Sensors J.*, vol. 20, no. 2, pp. 1043–1053, Jan. 2020.

[50] 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.

**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 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 an Associate Professor of Mechanical Engineering, Electrical and Computer Engineering, and Biomedical Engineering with Texas A&M University, College Station, TX, USA. Her recent research interests include infrared sensing, signal processing, and machine learning, with applications in occupancy detection, human identification, and activity tracking. She has authored one- book chapter, 74 journal papers, and 40 conference proceeding papers and filed four U.S. utility patents and two provisional patents.