


Research Article

Security Monitoring in Smart Homes Using IoT Data Analytics

Zahraa M. Ammar Smaysm^{1,*} 
Department of Media and Government
University of Information and Communications
Technology
Bagdad, Iraq
zahraa.smaysm@gmail.com

²Davood Akbari Bengar² 
Department of Computer Sciences,
Science and Research Branch,
Islamic Azad University
Khouzestan, Iran
Akbari.b1980@gmail.com

ARTICLE INFO

Article History

Received: 07/01/2024

Accepted: 31/03/2024

Published: 01/06/2024

This is an open-access article under the CC BY 4.0 license:

<http://creativecommons.org/licenses/by/4.0/>



ABSTRACT

With the rapid proliferation of Internet of Things devices in smart homes, the necessity of robust security measures has become clearer than ever. This study aims to apply data analytics to enhance security monitoring within smart home environments. By leveraging the wealth of data generated by IoT devices, the study also aims to create an intelligent system that is capable of proactively identifying and addressing potential security threats through Android mobile devices.

Therefore, privacy concerns were addressed through encryption and anonymization methods to protect sensitive information. The study evaluates the effectiveness of the developed security monitoring system through simulated and realistic scenarios, thereby highlighting its ability to detect and mitigate a wide range of security threats. Thus, the research contributes to the development of smart home security by providing a smart, data-driven approach to monitoring security incidents. In the evolving landscape of smart homes, the proposed framework forms a cornerstone for ensuring the safety and privacy of residents within this interconnected ecosystem.

Keywords: *Internet of Things (IoT), Smart home security, Data analysis, Android mobile application*

1. INTRODUCTION

This Smart homes have long had the potential to improve the safety and productivity of our daily surroundings. The majority of people's time is spent at home or at work [1] According to the Environmental Protection Agency, Americans spend up to 89% of their time indoors, 69% of which inside residences [2]; moreover, many people believe that these locations are their havens. [3]. Smart homes can use various technologies, such as integrated machine learning and sensors methods to identify, recognize, and react to possible hazards to preserve this sense of comfort [4]. In this study, we present a home security strategy that is used on smart homes. Our proposed method is based on the partnership between the Intelligent Home Security and Control System (iHOCS) and the Center for Advanced Studies in Adaptive Systems (CASAS). The partnership represents an innovative step toward achieving enhanced security and comfort in smart homes. This partnership also combines smart home control technology with comprehensive data analysis capabilities for lifestyles and overall health. iHOCS offers advanced systems for controlling lighting, heating, home security, and comfort. Meanwhile, CASAS leverages data analysis and artificial intelligence to gain a detailed understanding of the lifestyle and health patterns of individuals residing in the home. This integration allows individuals to enjoy a highly customizable and intelligent level of comfort and security within their homes.

Through this partnership, the home experience is greatly enhanced because our proposed method relies on integrating sensors into the surrounding area. Sensors collect data about residents and home conditions. Using these details, activity learning approaches may help to discover and explain typical or expected behavior in terms of known and expected activities. Threat detection is based on the perception of abnormal behavior on this detected behavior. As soon as strange conduct is recognized as dangerous, the decision regarding the course of action to be taken is made at home, and users can proactively address the threat through Android mobile devices [5]. Our approach was evaluated within the context of actual smart home tests based on reading data from CASAS. This center is the infrastructure of our advanced systems based on adaptive systems technologies. CASAS and iHOCS for smart homes, which are the

infrastructure installed in homes [5], is explained in the next section. We use real and synthetic data from independently evaluated smart homes. By identifying and examining routine or typical behavior, activity learning approaches use these data to evaluate our approach to identifying and responding to risks by recognizing and anticipating recognized behaviors. In addition, using data from Kyoto's multiresident smart housing on the test site, we are showcasing a smart home that is capable of autonomous response.

These integrated technologies can ultimately achieve the overarching goal of making smart homes more interactive and intelligent by integrating proactive and analytical capabilities, which can improve energy efficiency, safety, and the personalization of healthcare and provide a comfortable and safe environment for residents.

Finally, we discuss barriers to the use and commercialization of intelligence-based home security devices as well as possible solutions.

2. RESEARCH BACKGROUND

2.1. Intelligent Home Control and Security System (iHOCS)

First, as an Internet of Things (IoT) application, smart home automation allows for the remote control of mobile homes, thus enhancing security, safety, comfort, and energy efficiency. IoT technology empowers users to control their homes and devices from anywhere. Various commercial products, such as Amazon Echo and Google Nest Hub, have been developed for intelligent home control. Despite progress, challenges such as interoperability, data security, data analytics, communication security, and home automation persist [6, 7]. To address these issues, an intelligent home automation system called iHOCS is introduced. The system's general functionality is explained and shown in the next paragraph.

a. iHOCS System

iHOCS is an IoT-based system that is designed to control home appliances and monitor environmental factors, thus ensuring comfort, convenience, and safety. It employs the support vector machine algorithm to classify images that can distinguish among regular occupants, their pets, and intruders. An Android-based smartphone software allows users to monitor and manage their houses.

iHOCS consists of six modules: **the intelligent device module** (controlling smart appliances and sensors), **the communication and gateway module** (facilitating communication between devices), **the management and decision module** (enhancing security and reducing false alarms), **the cloud computing module** (for data storage and processing), **the presentation module**, and **the security module** [6, 7].

- i. **Intelligent Device Module:** This module describes smart home devices and sensors, including light bulbs, heaters, and cameras. Sensors are used for data gathering and monitoring.
- ii. **Communication and Gateway Module:** Interaction between devices and sensors is made possible by this module, which uses an ESP8266 Wi-Fi board for communication with the Internet.
- iii. **Management and Decision Module:** It employs the support vector machine to classify images and send alerts based on detected motion, thus being able to distinguish between intruders and regular occupants.
- iv. **Cloud Computing Module:** Data generated by the system, including environmental data, are stored in the cloud for real-time monitoring and analysis [8].
- v. **Presentation and Control Module:** This module manages the user interface for remote home control. It allows users to control and monitor home appliances through a mobile application. The module includes screens and tabs for various tasks and displays data from home sensors, such as temperature and humidity, with the mobile application's passive infrared (PIR) motion sensor.
- vi. **System Security Module:** This module ensures user security. It employs an authentication mode to verify the user's identity. During the mobile app setup, a unique authentication code is sent to the user's email address. This code is necessary to program the system to control and monitor devices. Without the correct authentication

b. Architectural Design

The user, sensors, Wi-Fi module, home appliances, and cloud platform constitute the iHOCS system's architectural design. The ESP8266 Wi-Fi module functions as a microprocessor and a communication device, as seen in Figure 1. It gathers information from sensors and sends it to the user over the Internet. Through communication with the ESP8266, the user engages with the system via an Android mobile application. Security, control, and home

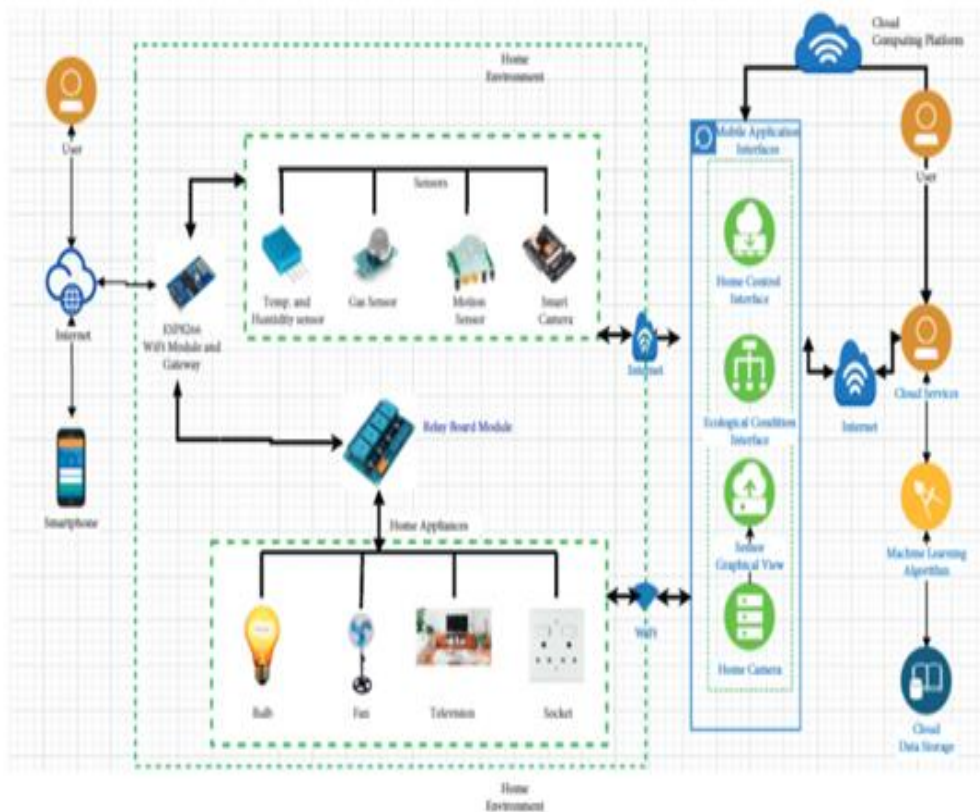


Fig. 1. . iHOCS Architecture

Controlling electrical outlets and basic household equipment, such lightbulbs, televisions, and environmental monitoring devices, is part of the iHOCS system setup process. It also detects movements and captures images when necessary. The relay board module connects sockets and appliances to the ESP8266 for command communication. Sensors, such as the DHT11, measure temperature and humidity. Security is enhanced using a PIR motion sensor, which sends alerts to the Android app. Data from all sensors are stored in the cloud, which is accessible through the mobile app. Algorithm 1 outlines the steps and services involved in the iHOCS system.

The system makes intelligent decisions using the SVM algorithm, thus ensuring efficient home monitoring and security. Abbreviations used in Algorithm 1:

EHA: Electrical Home Appliances, **SD:** Home sensors and detectors, **NC:** Network Connectivity, **HC:** Home control, **M:** Motion, **CS:** Cloud storage, **HO:** Home Occupant, **PE:** Pet, **HP:** Home premises, **D:** Darkness, **LDR:** Light Dependent Resistor, **IM:** Image.

This comprehensive system combines technology and intelligent algorithms to enhance home automation and security while addressing existing challenges. The architectural design is illustrated in Figure 2.

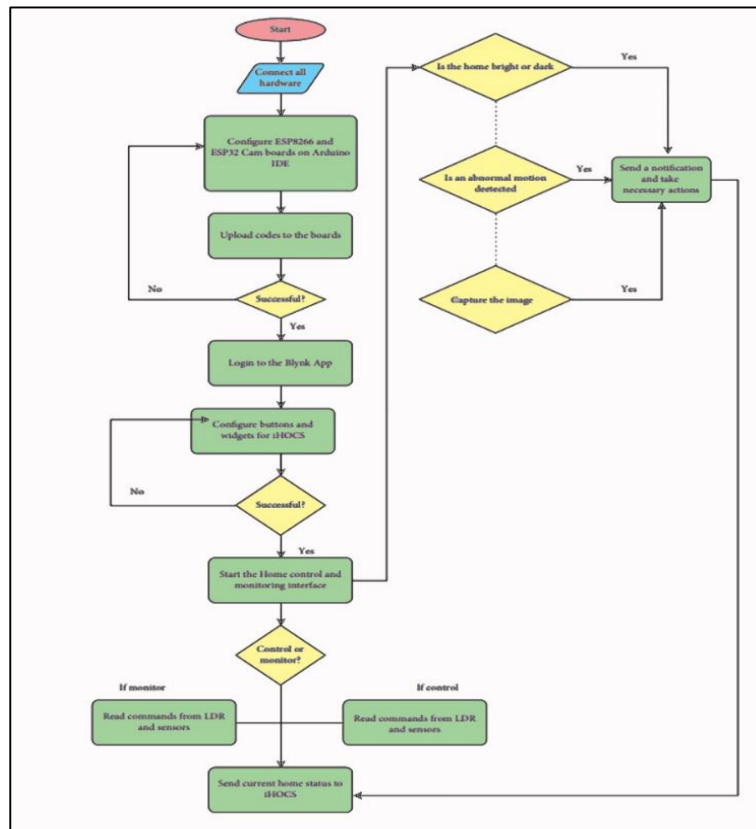


Fig. 2. Flowchart of iHOCS.

2.2. CASAS Smart Home

The infrastructure for smart homes created at CASAS serves as the basis for our suggested secure smart house. A continuous cycle is represented by the sense, identify, assess, and act functions of the smart home (Figure 3a). The house monitors the tenants' physical environment and their health using integrated sensors to analyze behavior and gauge residents' wellbeing (Figure 3b) The available software assures the occupants' comfort, safety, productivity, and security.

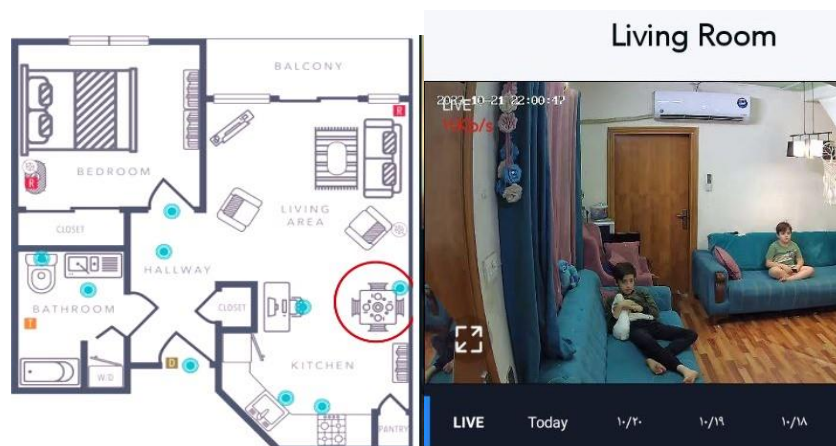


Fig. 3. Project objectives and the CASAS smart apartment testbed (a) smart home infrastructure; (b) home monitoring

The CASAS infrastructure presented in Figure 4 [4] has physical-layer components that can detect the environment and take appropriate action. The middleware layer's components handle component synchronization, identification, and communication. In addition, the application layer's components offer customized services such as activity learning, sensor fusion, and optimization for particular smart home objectives.

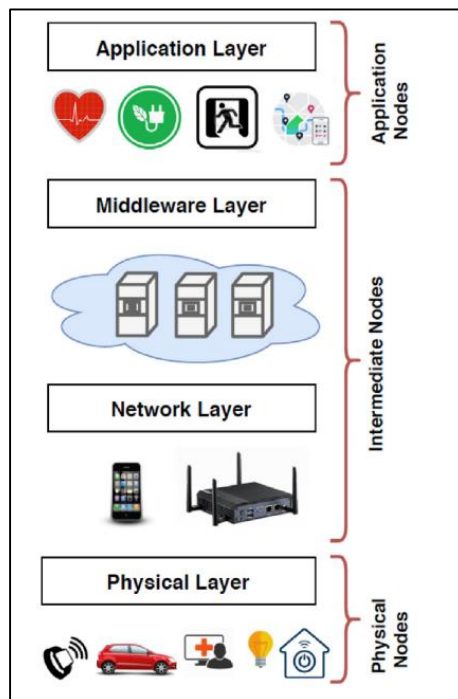


Fig. 4. IoT is used in smart home software infrastructure.

The home is outfitted with infrared motion/ambient light sensors, which give readouts for opening or shutting doors or windows and controlling equipment that change temperature, such as stoves and showers (Figure 5). They also have combined shutdown/temperature sensors. Drawing from prior research, we have distinguished four kinds of smart home metrics that are employed to extract and correlate with variables [9]. Using four different categories of smart home characteristics, we were able to extract and correlate chemical variables. They include temperature, the entire area of open doors and windows, the length of each action with an automatic label, and the overall activity level (based on felt movement). To determine the activity level, the number of motion sensor “trigger” events in each room of the house is counted. This feature can be compared with chemical sensors. Over the period of continuous data collection, we recorded this information hourly.

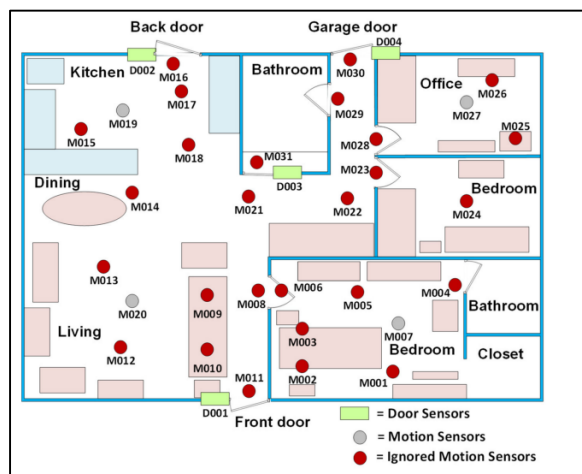


Fig. 5. CASAS smart apartment testbeds: Kyoto

We tracked household activities and recorded their durations during the corresponding hour of data collection because of the availability of activity recognition software. We tagged the motion, door, light, and temperature sensor data from the smart home with the appropriate activity labels using machine learning algorithms. Then, the time span of the sensor events during the activity-designated hour was used to compute the activity duration. Based on threefold cross-validation, our machine-learning approaches produced activity labeling with an average accuracy of 95% [10].

The behaviors we tracked for this study included resting in bed, using the restroom, relaxing, going outside the house, cooking, eating, maintaining personal hygiene, taking a bath, coming inside the house, taking medication, doing the dishes, and working. Through software “bridges” or communication links, the smart home’s components talk to one another. The Zigbee bridge, the Scribe bridge, which saves messages and sensor data in a relational database, and the bridges for individual applications are examples of these bridges that facilitate network connections. The smart home testbeds we utilized to evaluate our proposed secure smart house use the simplified CASAS “smart home in a box” (SHiB), as presented in Figure 6. Each SHiB sensor has a predetermined placement that corresponds to a functional region of the house, thus enabling the development of generalizable activity models. An average of fewer than two hours may be spent placing each CASAS SHiB [4] and less than 20 hours to remove them. The inability of many smart homes and activity learning initiatives to evaluate innovative software techniques, such as those described in this article, outside laboratory or simulation settings is one of their main problems. In addition, the design is lightweight [5].



Fig. 6. Smart Home in a Box (SHiB) is a product of the Center for Advanced Studies in Adaptive Systems (CASAS).

The smart houses may safely upload events stored in a relational database on the cloud even if each smart home site functions independently. Table 1 displays an example of data collected from the smart home together with the corresponding, automatically produced activity labels. For instance, this series of sensor events is associated with the label for sleep activity:

TABLE I. TABLE 1 AN AUTOMATICALLY GENERATED ACTIVITY LABEL, A TIMESTAMP, A SENSOR NUMBER, AND A SENSOR MESSAGE ARE ALL INCLUDED IN RAW SENSOR DATA.

date	time	Sensor	status	annotation
2022-03-10	06:48:24.855293	bedroomed Motion	ON	Sleep
2022-03-10	06:48:29.727262	bedroomed Motion	OFF	Sleep
2022-03-10	06:48:30.479044	bedroomed Motion	ON	Bed-Toilet
2022-03-10	06:48:33.102565	bedroomed Motion	OFF	Bed-Toilet
2022-03-10	06:48:35.102567	bedroomed Motion	ON	Sleep

2.3. IoT Software and Hardware Tools

We used a variety of IoT hardware and software technologies inside the framework of the iHOCS system's architectural design to develop an intelligent home control, monitoring, and security system. This development required the usage of a number of hardware parts, such as the DHT11 sensor, the HC-SR501 PIR motion sensor, the ESP8266 Wi-Fi board, and the relay module. The iHOCS Android mobile application enables users to perform functions such as controlling home appliances, monitoring environmental conditions, automating lighting based on home conditions, detecting motion, receiving home notifications, and viewing sensor data graphically [30].

a. DHT11 Sensor

The DHT11 sensor is an essential component for measuring temperature and humidity in the home environment. It provides real-time data, which are displayed on the user's smartphone via the Android application, thus allowing for adjustments to ensure comfort. Its simplicity, low cost, and high sampling rate make it an effective choice [6].

b. PIR Motion Sensor HC-SR501

The HC-SR501 is a reliable and sensitive motion sensor used for detecting unauthorized movement in the home. This IoT device is applicable in security systems and other automated applications, thereby providing a signal to the ESP32-CAM module for image capture. The SVM algorithm classifies movement features to distinguish intruders [6].

c. Wi-Fi Development Board ESP8266

The ESP8266 serves as the Wi-Fi module and development board in the system. It enables communication with the Wi-Fi network and facilitates the connection with sensors and the 5 V four-channel relay module; hence, electrical appliances can be controlled remotely [6].

d. Four-channel Relay Module with 5 V

This module is used to control high voltage and current loads in the home. It enables the remote activation of various electrical appliances, such as fans, bulbs, sockets, and additional units. LED indicators indicate the status of the relays [6]. These hardware components collectively contribute to the functionality of the iHOCS system, thus enhancing home control, monitoring, and security through IoT technology.

e. ESP32-CAM

A low-cost development board for IoT prototyping that has a Wi-Fi camera for taking pictures and streaming videos is called the ESP32-Cam. The ESP32-CAM is a system-on-a-chip module that features an integrated PCB antenna, a camera, a micro-SD card interface, and Wi-Fi and Bluetooth capabilities. The security of a smart home automation prototype is guaranteed by the ESP32. Given the lack of a USB port on the ESP32-Cam, it is utilized in this instance using an FTDI programmer. Consequently, the ESP32-CAM module receives instructions for taking pictures via the FTDI, which is used to transfer code created in the Arduino IDE to it.[11]

Algorithm 1: Intelligent Home Control, Monitoring, And Security Algorithm [11].

```
(1) Begin
(2) Define  $N_c$  parameters
(3) Initialize  $EHA$  and  $HSD$ 
(4) Establish and confirm the status of  $N_c$ 
(5) If  $N_c = 1$ 
(6) Evaluate the initial state of  $EHA$ ;  $\forall EHA \in N_c$ 
(7) while  $EHA = n$  (where  $n$  = number of configured home appliances)
(8) Start  $HC$ 
(9)   Else, go to step 4
(10) End if
(11) While  $N_c \&\&HC = 1$ ; continue action till the desired state is reached
(12) Evaluate the initial state of  $HSD$ ;  $\forall HSD \in N_c$ 
(13) If  $HSD = n$  (where  $n$  = number of home sensors and detectors)
(14) Connect  $iHOCS$  to the Internet
(15) Acquire sensor data to  $iHOCS$  via the  $ESP8266$ 
(16) Else, go to step 4
(17) For each round do
(18) Get the values for  $T, H, M$  and  $IM$ 
(19) Upload data to  $CS$  via  $iHOCS$ 
(20) Update status of  $HSDs$  in  $iHOCS$ 
(21) Display graphical status of  $HSDs$  in  $iHOCS$ 
(22) Synchronize data to  $CS$ 
(23) Else, go to step 12
(24) Case 1: (LDR)
(25) if  $(D = 1)$  then
(26) Notify the user "It's DARK, Turn on the LIGHTS"
(27) Else
(28) Notify the user "Its BRIGHT, Turn off the LIGHTS"
(29) break;
(30) Case 2: (PIR Sensor)
(31)   If  $M$  is detected, then
(32) Notify user via e-mail "TOSIN: Motion detected"
(33) break;
(34) Case 3: (ESP Cam)
(35) If  $M$  is detected, capture  $IM$ 
(36) Notify via  $iHOCS$  and apply  $SVM$ 
(37) If  $IM \in (PE_1, PE_2, PE_3, HO_1, HO_2, \dots, HO_n)$ 
(38) Mute alarm
(39) Else,
(40) Raise alarm and send picture to e-mail
(41) end if
(42) User monitors  $EHA$  and  $HSD$  via  $iHOCS$  app
(43) Remotely control the home
(44) End
```

2.4. Software Components

Only a predetermined set of preconfigured instructions, methods, and programs may be used to operate and monitor the house, its gadgets, appliances, and sensors. Specific software programs are required for the correct operation and communication of our $iHOCS$ system. The Arduino IDE and Blynk software are the main software packages used for configuring, programming, designing, and developing the $iHOCS$ system.

a. Arduino IDE

For use with Windows, Linux, and MAC operating systems, the Arduino Integrated Development Environment is a cross-platform application based on the C and C++ programming languages. To create code that can be uploaded to

boards, the Arduino IDE is utilized. In IoT prototyping [32], it is used to setup microcontroller boards. The Arduino IDE is used to write the code for iHOCS, which is then uploaded to the ESP8266 and ESP32-Cam boards to operate household appliances and sensors.

b. Blynk Application

The Blynk app, which is compatible with iOS and Android, is a cross-platform tool for IoT project design. It facilitates the creation of mobile applications to control hardware and display sensor-generated data. Blynk consists of three key elements: the app for designing project interfaces, a server for hardware–smartphone communication, and libraries for interconnecting hardware, server, and commands. In our iHOCS system [31], we employ Blynk to craft graphical user interfaces and reading interfaces, thus utilizing its cloud services for real-time sensor data storage.

3. ACTIVITY LEARNING

Our method of danger detection in smart homes is distinctive because it uses information about the present and activities to spot changes from typical behavioral patterns. These variances signify possible hazards that need additional attention and a reaction. The present context is often identified using a defined set of characteristics, such as time and location. Many context-aware services, including security services, are built upon this context. This parameter set is quite limited, and the parameters are normally taken into account independently. We hypothesize that learning activities give smart houses a better supply of knowledge and may hence increase home security. Furthermore, given that it can spot deviations from ingrained complicated activity patterns, we argue that a house that is aware of activities will provide even more robust security services than one that only tracks the whereabouts and movements of people. Secure smart homes must be able to learn from and comprehend observed actions to be responsive to the demands of the people who live in them. Our smart home uses activity learning to transform the system into an activity aware one. We further examine the algorithms for activity detection and discovery, which are the foundation of activity awareness in the secure smart home.

3.1 Activity Recognition

Based on the information gathered from environmental sensors, activity identification algorithms provide labels to various types of activities. When the data are accessible, we can determine what situations—such as sleeping, coming and departing from the house, cooking, and other activities involving valuables—affect home security. An activity recognition algorithm's objective is to map a series of sensor events or readings, $x = \langle e_1 e_2 \dots e_n \rangle$, onto one of a number of prespecified activity labels, $a \in A$ [5]. One kind of supervised machine learning task is activity recognition. [5]. We also presume that a feature function is available, Φ .

This function is able to take a series of sensor events and calculate a d-dimensional feature vectors [5]. To transfer a feature vector, $X \in R^d$, which represents a specific sensor event sequence onto an activity label, $h: X \rightarrow A$, our activity recognition system, CASAS-AR, trains a function h . The learned function may be used by CASAS-AR to identify and label instances of the learned activity [5]. Among machine learning issues, activity recognition presents some difficulties. The input data are occasionally multilabel, frequently sequential and noisy, and not always cleanly divided into activity segments. Further data processing can address some of these problems. For example, the steps shown in Figure 7 comprise gathering and cleaning up sensor data, thus breaking them down into smaller, more manageable segments.

Then, the properties of these segments are extracted. The final feature vectors are either put into a trained model to obtain the appropriate activity label or labeled by an expert to be used as training data. Table 2 summarizes the features we utilize to transform the raw data from smart homes into activity models. All smart home sensors detect discrete events, and sampling-based sensors give a continuous stream of data. As such, they only send texts containing sensor data or events while the sensor is inside.

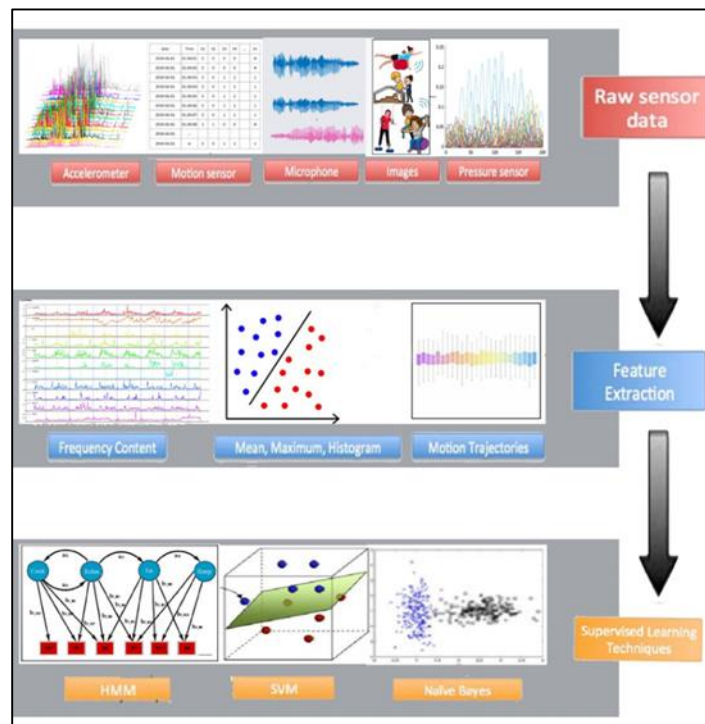


Fig. 7. Phases of gathering raw sensor data, segmenting and preparing the data, extracting features, and supervising machine learning are all included in activity recognition.

TABLE II. ACTIVITY CLASS DESCRIPTIONS, CHARACTERISTICS, AND RAW SMARTPHONE DATA

Domain	Number	Types of Features
Raw Sensor data	4 types	infrared motion (ON/OFF); magnetic door (OPEN/CLOSE); ambient light (continuous); ambient temperature (continuous)
Timing features	3 features	day of week; hour of day; seconds past midnight
Window features	9 features	most recent sensor in window; first sensor in window; window duration; most frequent sensors from previous two windows; last sensor location in window; last motion sensor location in window; entropy-based data complexity of window; time elapsed since last sensor event in window
Sensor features (n sensors in home)	2n features	count of events for each sensor in window; elapsed time for each sensor since last event

Our safe smart home’s activity identification method expands on earlier research that used machine learning approaches to generate activity models from sensor data [12, 13]. Various groups have explored supervised activity recognition across different sensor types, including environment, wearables, objects, smartphones, and video [14, 15],[16, 17],[18]. These approaches encompass template matching, generative, discriminative, and ensemble methods, each offering unique advantages. Nevertheless, current activity detection algorithms are frequently restricted to presegmented data, single users, and no activity pauses, and they are often created for particular circumstances. Recent efforts have extended these methods to generalize activity models across multiple users and recognize activities in new spaces without prior training data. Real-time data labeling eliminates the need for offline data segmentation; a sliding window is applied to sensor data for feature extraction and activity label assignment, thus achieving >95% recognition accuracy for numerous activities in various smart homes [19, 20]. This approach allows for a robust and generalized activity recognition system for secure smart homes.

3.2 Activity Discovery

We must combine activity identification and discovery using a creative approach to progress the field of activity recognition research and build an activity-aware smart home. Finding activities that are interesting to track and modeling those activities form the most popular method for inferring human activities from sensor data. However, a significant amount of sensor data that have already been tagged with ground truth information must be accessible to model and recognize actions. Such prelabeled data are not easily accessible for the majority of operations that are conducted in real-world settings. As a result, the collection of routinely monitored actions only constitutes a small

portion of a person's overall behavioral pattern [21, 22]. Given that the remaining data offer crucial context and lumping unlabeled actions into the "Other activity" category results in a large class imbalance, tracking so few activities can have an impact on learning performance. More than half of the sensor data are accounted for by the "Other activity" category when considering the datasets. This situation is difficult to depict effectively because the "Other activity" category represents a possibly infinite combination of different activities. Although activity detection can be used in several other ways for smart home security, we specifically use it in this instance to address the problem of class imbalance. In particular, we provide intraclass clustering as a stage in the preprocessing pipeline for activity detection [1]. In this case, major classifications, such as "Other activity", are subdivided into smaller divisions. This process causes the activity class a_i to be split up into subclasses: $S_{ai} = \{c_{i1}, \dots, c_{ik}\}$. To create the new activity models, training instances are given new class names that correspond to the associated subclasses. Using labels from specified and undiscovered subclasses, the classifier will forecast activity labels. The parent class labels (in this case, "Other activity") are mapped back onto the subclass labels when the learned models are used to classify fresh data. This process may be done using a variety of clustering techniques. Others, such as cascade simple k-means [23], divide data based on natural groupings. Certain algorithms, such as k-means++ [24], need a certain number of clusters; in our scenario, to maintain balance, we have selected the number of clusters so that the resultant subclasses are fairly near to the mean activity size. According to our earlier study [25], for unbalanced class distributions, intraclass cluster typically performs better than sampling-based techniques. In our safe smart home, we use intraclass clustering using k-means++ to improve the CASAS-AR activity identification.

3.3 Anomaly Detection

To improve security in smart homes, the system can be trained to identify particular incidents, including resident falls. Although this technique can work well with a large amount of realistically labeled data, such data can be difficult to gather in actual houses. Furthermore, limiting the system's emphasis to predetermined target scenarios may prevent it from identifying a greater variety of security concerns.

To address this limitation, we can turn to anomaly detection. Anomaly detection involves identifying patterns in data that deviate from expected behavior, as defined by Chandola [26]. Smart homes can be helpful in automatically spotting abnormalities that may endanger someone's safety or health [27]. Various established techniques can detect anomalies or outliers. One common method involves clustering data points according to how far away they are from the cluster center. After clustering, data points that are significantly distant from all cluster centers can be identified as outliers or anomalies. In cases where data follow a normal distribution, z-scores can be calculated. A data point's z-score, which is normalized by the sample standard deviation, indicates how much it deviates from the sample mean. Z-scores greater than 3.5 are usually regarded as outliers. However, the use of clustering techniques presents a problem because they rely on determining the distances between data points, which cannot adequately represent the complexity of multivariable data produced by smart home sensors. We use an isolation forest to overcome this issue and find anomalies in the sensor data from smart homes [28] to produce scores for anomalies; a larger score indicates a more notable deviation from the usual data patterns. To identify data points that fall into recognized classifications, an isolation forest creates a decision tree. Decision tree techniques aim to construct the smallest decision tree that is consistent with the training set of data. A data point, x , that has an abnormally long route from the decision tree's root to the data point located at the leaf node, or $h(x)$, may be regarded as an outlier. Given that each branch along the path requests the value of a particular characteristic, this anomaly identification approach is sensitive to activities occurring within the home and employs the same feature vector as activity recognition. The same set of data is used to train a collection of decision trees, which generates an anomaly score. By randomly selecting a feature to query and a split point for the feature values, each tree in the collection generates its offspring. A high average $h(x)$ value, or $E(h(x))$, indicates that a point is more likely to reflect a real anomaly because several trees lessen the propensity to overfit. Figure 8 presents the decision tree and the $h(x)$ value. [1] The average height is divided by $c(n) = 2H(n-1) - (2(n-2)/n)$ to standardize the value across trees. In a binary search tree, $H(i)$ is the harmonic number, n is the number of leaves nodes, and $c(n)$ indicates the average distance travelled during a fruitless search [1, 5]. The definition of the anomalous score s of a data point x is given by Equation (1):

$$s(x, n) = 2^{-\frac{E(h(x))}{c(n)}} \quad (1)$$

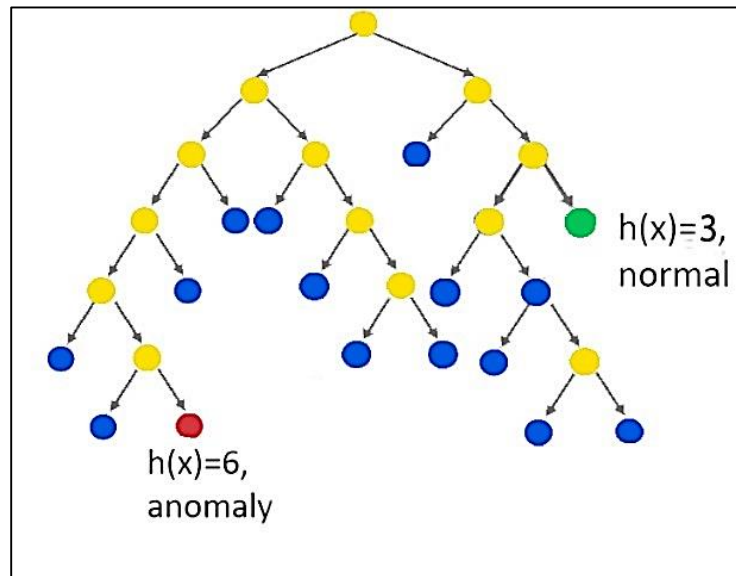


Fig. 8. Path lengths are averaged over a collection of decision trees to determine the anomaly score for a smart home sensor data sequence.

Even on streaming data, this approach has proven successful in identifying abnormalities for a wide range of applications [29]. We identify anomalies as the data points that have the highest anomaly scores. Prior to determining the score threshold that should be reported as abnormal, the application must be selected. The number of true positive anomalies that are discovered may decrease if the threshold is set too high because some abnormalities can go undiscovered. False positives will increase if the threshold is set too low because many normal events will be recorded as anomalies. We assessed our method using multiple outlier reporting fractions in a real-world scenario. We discovered that reporting 0.1% efficiently identifies the majority of known anomalies without producing an excessive number of false positives.

3.4 Hybrid Real–Synthetic Data Generator

In our evaluation, we employ a hybrid real–synthetic data approach. We utilize real smart home data to test our security detection system under normal daily conditions. However, given that anomalies in real-world situations are infrequent and often undocumented, we create synthetic data resembling real-world scenarios. Then, the synthetic data are modified to include known anomalies. To generate this hybrid dataset, we design a data generator based on a Markov model. The generator yields data that, depending on the detection techniques used, may be classified as anomalies. However, it also generates data that are probabilistically comparable with genuine smart home information. A statistical model for characterizing dynamic systems is called a Markov model. It establishes a limited number of states that are each connected to a multivariate probability distribution across parameters. Transition probabilities control changes from one state to another. In this instance, the states stand for potential sensor event states found in a smart house. Assuming that the method follows a Markov process (the history of past states is limited, and it determines the current state.), we use a hidden Markov model (HMM), which is a stochastic process that is not immediately observable. According to the Markov property, the probability of a state depends only on the probability of the state that came before it. We employ a third-order Markov model in our model. As Equation (2) illustrates, the likelihood of the subsequent state, x_{i+1} , relies on the three states that came before it.

$$P(x_{i+1} | x_i, x_{i-1}, x_{i-2}, \dots, x_1) = P(x_{i+1} | x_i, x_{i-1}, x_{i-2}) \quad (2)$$

First, we use the real smart home data and extract the structure of our HMM to create synthetic data. [21]. This structure guides the synthesis of data that closely resemble the original dataset but may include slight variations. This process allows us to create a substantial volume of data while maintaining the same anomalies identified in the first datasets from the actual world. HMMs are widely used in a variety of fields, such as voice synthesis, gene sequencing, and cognitive processes, and are ideally suited for such data creation tasks [22, 23].

3.5 Validation of Anomaly Detection

Our review has two main phases. First, we confirm that our anomaly-based method for detecting threats to smart home security is accurate. Second, we demonstrate its usefulness in a smart home security situation. For the initial step of

validation, we employ a hybrid real–synthetic data approach by simulating one week of data for the smart home. In these datasets, we deliberately introduce 20 random anomalies by modifying the generated data. Each anomaly consists of 30 events, which match our activity recognition sliding window’s size. These alterations are made at random while preserving the original timestamps. We experiment with various feature vectors to investigate the effect of activity awareness on anomaly identification. These are the feature vectors that have been tested.

- *f.standard*: It includes the Table 1 timing, window, and sensor characteristics.
- *f.activity*: Timing information and the activity label should be included in the output of the discovery-enhanced CASAS-AR activity recognition algorithm for the corresponding window of sensor data.
- *f.all*: It includes the activity label and all the regular features.

As part of our performance review, we calculate true positive and false positive rates to gauge how well our anomaly detection system performs in terms of identifying known abnormalities. By changing the percentage of recorded anomalies from 0.1% to 0.5%, these rates are computed. A receiver operating characteristic (ROC) curve is made. Then, the AUC, or the area under the ROC curve, is calculated using the data. This assessment technique not only assesses the overall effectiveness of the algorithm but also enables us to evaluate various feature vectors to comprehend how activity awareness affects our anomaly detection methodology.

Table 3 provides a summary of our experiment’s findings. We observe that most embedded anomalies are successfully detected across all datasets. The B1 dataset, which is characterized by a highly structured daily routine, presents an easier case for detecting anomalies. However, the anomaly detector identifies several anomalies in the normal data even in this dataset, which is expected given the inherent unpredictability of human behavior.

Conversely, the B3 dataset proves to be the most challenging for anomaly detection. This difficulty can be attributed in part to the existence of preexisting anomalies in the normal data, thus making the embedded anomalies more difficult to discern.

TABLE III. AREA UNDER CURVE (AUC), AVERAGE FALSE POSITIVE RATE (FPR), AND AVERAGE TRUE POSITIVE RATE (TPR) OF THE B1, B2, AND B3 DATASETS USING DIFFERENT FEATURE VECTORS.

Dataset	Features	Average TPR	Average FPR	AUC
B1	<i>f.standard</i>	1.0000	0.0008	0.73
	<i>f.activity</i>	1.0000	0.0021	0.68
	<i>f.all</i>	1.0000	0.0008	0.87
B2	<i>f.standard</i>	0.8000	0.0021	0.49
	<i>f.activity</i>	0.5000	0.0032	0.42
	<i>f.all</i>	1.0000	0.0011	0.79
B3	<i>f.standard</i>	0.4500	0.0044	0.47
	<i>f.activity</i>	0.7000	0.0040	0.40
	<i>f.all</i>	0.8500	0.0038	0.47

Across all datasets, the inclusion of activity awareness enhances the anomaly detection capability. Specifically, in the B1 and B2 datasets, combining standard features with activity labels significantly improves the AUC values compared with using standard features alone ($p < 0.01$). Using only activity labels in the B2 dataset proves to be successful in recognizing almost all the embedded abnormalities.

4. TRAINING OF THE MACHINE LEARNING

An extensive synopsis of the machine learning image classification dataset’s training is given in this section. Lights, fans, outlets, and one additional electrical equipment in the home are all controlled by the iHOCS system. It also monitors the humidity and temperature in the house, senses movement, and takes pictures when it detects movement. These images are promptly transmitted to the user’s Android-based smartphone via the mobile application, which is accessible from anywhere. Coding in the Arduino IDE and uploading the code to the board using a USB connection comprise the setup phase. To guarantee smooth connectivity, some home network credentials—such as the Service Set Identifier (SSID) and password—are set during programming. Additionally, the Blynk mobile application setup created an authentication code. Commands to control the appliances’ state (turning them on or off) are issued via an Android-based smartphone. Sensor-generated data are gathered, saved in a cloud database, and shown on the mobile application’s screen, as explained in the system architecture. Compared with earlier approaches that depend on web-based apps and platforms, such as ThingSpeak, this method has a significant benefit. Without having to switch to or dismiss the home control program, users may simply follow data trends on the mobile application.

5. SECURITY APPLICATION

The second portion of our analysis focuses on the security case study scenario for a smart house. In this case, the resident's absence from the residence is monitored, activity is recognized using sensors, and facial recognition using a camera is enabled. When an anomaly is detected, the smart home activates the camera. We assess the secure smart home's anomaly detection system to show its capabilities in this situation. We use data from the Kyoto smart home testbed, which covers almost 10 years and is annotated with CASAS-AR software activities. Given that no security concerns were detected after data collection, we select instances of the "Enter home" action and create feature vectors for these occurrences, treating them as usual. Next, we evaluate the system's anomaly detection capability by introducing three sample cases into the data. These scenarios involve a person entering the home through different means. The smart home should recognize normal and anomalous behavior and potentially turn on the camera for the latter. The anomaly detection algorithm is applied, with a 0.1% threshold, and it detects known anomalies among the data points.

The findings demonstrate that the algorithm is cautious. It detects seven abnormalities, some of which were uncommon but did not pose a threat to security. Data marked as baseline (normal activity) are occasionally thought to be unusual, perhaps as a result of particular patterns such as maintenance tasks or packing tendencies at the conclusion of a semester. Table 4 presents the results if the implementation.

TABLE IV. IMPLEMENTATION OF ACTIVITY AWARE (ANOMALY DETECTION) IN SCENARIOS RELATED TO (SMART HOME SECURITY)

True Positives	False Positives	True Negatives	False Negatives
2	5	756	0

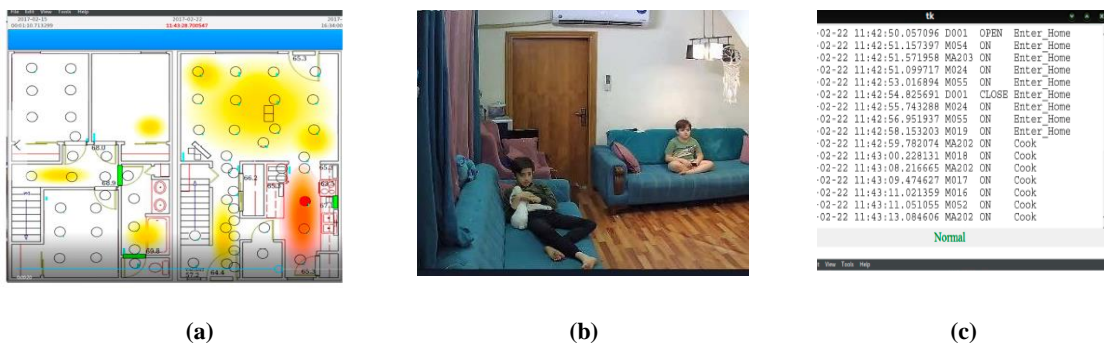


Fig. 9. Figure 8 Screenshots of Safe Smart Home Scenario 1. (a) The computer screen displays each sensor event. (b) The window's status bar at the bottom shows that it's typical for residents to move home during the hours that are recognized as such. (c) The sensor event visualizer in real-time displays the home's sensor status during the scenario. The kitchen (bottom right corner of the home) is where the occupant is seen.



Fig. 10. Screenshots of Safe Smart Home Scenario 1. (a) The computer screen displays each sensor event. (b) The window's status bar at the bottom shows that residents typically move around the home during the hours that are recognized as such. (c) The sensor event visualizer in real-time displays the home's sensor status during the scenario. The kitchen (bottom right corner of the home) is where the occupant is seen.

6. PROTOTYPE IMPLEMENTATION

As is common with smart home automation systems, the technology offers ease by permitting remote home management from any place. Consequently, the house may be remotely seen and managed from anywhere. Different LEDs indicate electrical appliances and sensors, and these LEDs can be controlled through a smartphone. Figures 11(a) and (b) illustrate the switching ON and OFF of appliances and show the control of multiple appliances. Figure 11(c) shows the user's notification that motion has been detected on the property and that an image had been taken. Our method is improved by allowing the user to take an instant picture of the culprit. This feature enables the user to proactive action while it automatically records all photographs of the individual causing the movement. If the user is outside, the user can immediately assess the situation at home. Alternatively, they can choose to disregard the message if they are inside the premises. The system shows the picture that is taken of the occupant of the residence when the motion sensor detects movement. An added advantage of iHOCS is the security of the home and the proposed SVM algorithm for user notifications. With the aid of an SVM algorithm, false alarms and unnecessary notifications can be averted. Although all images sensed by the system will still be captured for future reference, the user will only be notified when the system classifies the image as that of an intruder.

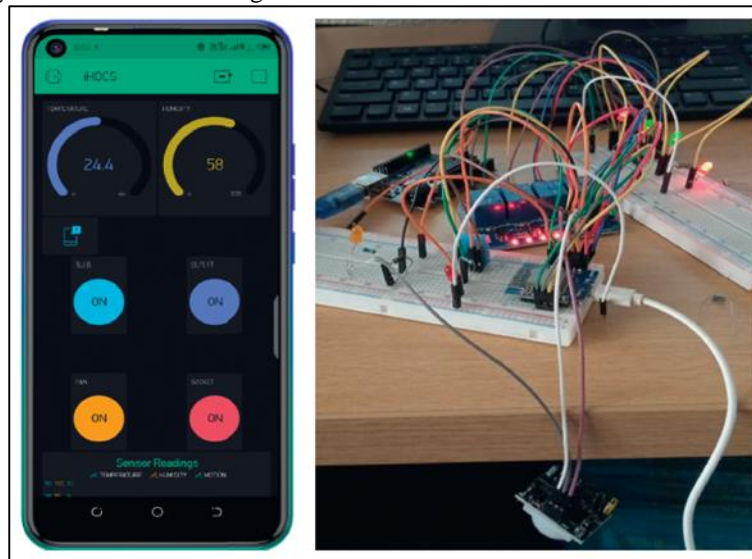


Fig. 11. (a) All appliances ON. (b) Corresponding light indicators.

7. ANALYSIS OF RESULTS AND DATA

This research was conducted with the aim of investigating the factors affecting the acceptance of the proposed new technology, security monitoring in the smart home using IoT data analysis in the smart city of Baghdad, using the UTAUT model.

7.1. IoT Data Analysis

- a. The study utilized quantitative research methods and structural equation modeling to validate the proposed hypotheses and confirm a conceptual framework.
- b. Data collection involved an online questionnaire converted to Google Form format, which was distributed to the target population. The process resulted in 130 valid surveys for analysis.
- c. The questionnaire comprised two parts: one related to demographic and behavioral information and another related to measurement information based on the proposed model, with responses measured on a five-point Likert scale.
 - a. The proposed model included four independent constructs: performance expectancy (PE), effort expectancy (EE), social influence (SI), and facilitating conditions (FC). The dependent variable was behavioral intention (BI). The model had 20 measurement items.
 - b. The conceptual framework was based on the UTAUT model, which includes the four main constructs of performance expectancy, expected effort, social influence, and facilitating conditions.
 - c. Three moderating variables were included in the research context: gender, age, and time spent reviewing the proposed new technology.

d. Four hypotheses were proposed:

- H1: Performance expectancy significantly affects behavioral intention to use the proposed new technology.
- H2: Effort expectancy significantly affects behavioral intention to use the proposed new technology.
- H3: Social influence significantly affects behavioral intention to use the proposed new technology.
- H4: Facilitating conditions significantly affect behavioral intention for the proposed new technology.

e. The measurement items for each construct are detailed as follows:

- Performance Expectation (PE)
 - Increases its protection.
 - Increases detection of security threats in smart homes.
 - Effectively enhances security and integration between homes and buildings.
 - Meets the security needs of citizens.
 - Reduces the time to detect security breaches.
- Expected Effort (EE)
 - Easy to apply.
 - Clear and understandable.
 - Easy to find the skills to use.
 - Saves money and time to identify all types of security threats.
- Social Influence (SI)
 - Supports security-related infrastructure, such as police department, fire department, etc.
 - The material value of homes and buildings that is used is higher than those that are not used.
 - Important people in the security field of Baghdad Smart City endorse its use.
- Facilitating Conditions (FC)
 - These conditions are necessary to operate the model.
 - Necessary knowledge.
 - Specific individuals or groups who can help with issues and problems are available.
 - Necessary resources.
 - Enough experience.
- Behavioral Intention (BI)
 - Construction industries are planning to use it in the near future.
 - The construction industry is determined to use it in the near future.
 - The construction industry intends to use it.
 - It will be used when an urgent need to improve security in Iraq's smart city emerges.

7.2. Descriptive Statistics and Demographic Variables

- a. Gender: Out of 130 individuals, 58 (44.6%) were women, and 72 (55.4%) were men.
- b. Age: Among the 130 individuals, 24 (18.5%) were under 30 years old, 54 (41.5%) were between 31 and 40 years old, 23 (17.7%) were between 41 and 50 years old, and 29 (22.3%) were over 50 years old.
- c. Education level: Of the 130 individuals, 47 (36%) were experts, 67 (51.5%) had a master's degree, 13 (10%) held a doctorate, and 3 (2.3%) had postdoctoral education.
- d. Type of work: Among the 130 individuals, 48 (37.36%) were in private partnerships, 67 (51.5%) worked in public institutions, and 15 (11.5%) were in mutual partnerships (private and public).

- e. Practical experience: Out of the 130 individuals, 32 (24.6%) had less than 5 years of experience, 23 (17.7%) had 5 to 10 years of experience, 50 (38.5%) had 10 to 15 years of experience, and 25 (19.2%) had more than 15 years of work experience.

7.3. Descriptive Statistics for Research Questions

- a. The result of the number of completely opposite and opposite answers compared with the other answers for each question depends on the number of different survey participants' responses to each question.
- b. This preliminary result is the first step to using IOT data analysis for security monitoring in homes and buildings in the smart city of Baghdad.
- c. Other descriptive indicators, such as mean and standard deviation, were determined for all research questions.
- d. Figure 12 shows that the highest standard deviation relates to question 8, which means that respondents are finding skills using the proposed new technology for security monitoring in homes and buildings in the smart city of Baghdad.
- e. They have more differences of opinion. The lowest standard deviation is associated with question 12, which means that respondents' answers to this question were more consistent than to the other questions.

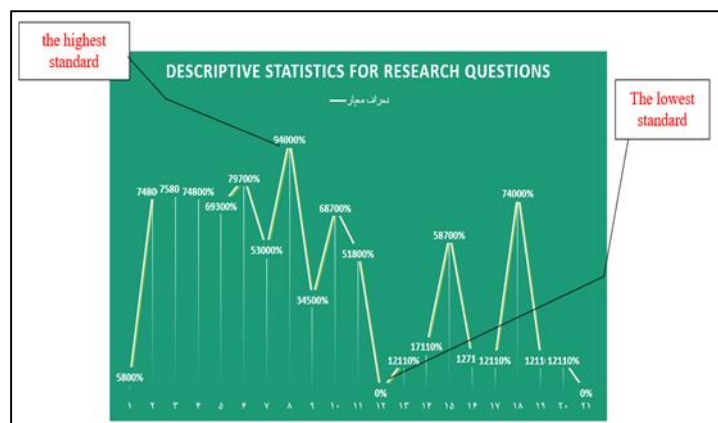


Fig. 12. Descriptive statistics for research questions

7.4. Results

- a. Experimental results show that the smart home automation method improves the control status, equipment management, and communication performance.
- b. The control accuracy of smart home always remains above 93%.
- c. The scheduling time for smart home control tasks is always less than 59 ms.

8. Conclusions

- a. Utilizing Real Datasets for Machine Learning: Research is conducted to gather and use real datasets for training machine learning algorithms in smart homes, especially for image classification tasks. This examination can improve the accuracy and effectiveness of smart home systems.
- b. Enhancing PIR Sensor Performance with Machine Learning: The use of machine learning is investigated to enhance the performance of PIR sensors in smart homes. It should focus on distinguishing among animals, occupants, and genuine security threats, which can significantly improve the reliability of these sensors.
- c. Developing Security Systems for Large Communities: The feasibility and implementation of security systems should be explored based on smart home automation concepts for large communities, such as smart cities, office areas, hotels, shopping malls, and academic environments. This research can propose scalable solutions tailored to each setting.

- d. Integration with Internet of Behavior (IOB) Devices: The potential integration of smart home automation models with IOB devices should be explored to enhance user comfort and security. IOB devices can be assessed to determine how they can augment the user experience in a smart home environment.

9. RECOMMENDATIONS FOR FURTHER STUDIES

- a. *Activity-aware Smart Homes*: Activity-aware approaches should be further investigated and developed for secure smart homes. Methods to leverage sensor data and activity recognition algorithms for real-time security threat detection should be explored.
- b. *Anomaly Detection and Response*: Research on anomaly detection in smart homes should be enhanced. The challenge of distinguishing between genuine security threats and benign anomalies in human behavior data should be addressed. Residents should be involved in training the system for improved accuracy.
- c. *Mobile Application Enhancements*: Further investigation should be conducted on how to incorporate machine learning into mobile apps for categorizing the system's recorded photographs. Real-time image classification for intruder detection should be implemented, and notifications to users should be optimized.
- d. *Cross-platform Compatibility*: The compatibility of mobile applications should be expanded to multiple platforms, such as iOS. Further examination should explore how cross-platform compatibility can broaden the user base and provide a seamless experience.
- e. *Predictive Analytics for Smart Home Environments*: The integration of machine learning for predictive analytics in smart home automation should be explored. Research should improve predicting weather conditions and home environmental factors to enhance user comfort and energy efficiency.
- f. *Application to Large Communities*: The applicability of the this study's smart home automation model to security systems in diverse large community environments should be determined. Moreover, the scalability, adaptability, and potential benefits in settings beyond individual residences should be assessed.

References

- [1] Sekar, A., E. Williams, and R.J.J. Chen, Changes in time use and their effect on energy consumption in the United States. 2018. 2(3): p. 521-536.
- [2] Hu, Y., et al., Smart home in a box: usability study for a large scale self-installation of smart home technologies. 2016. 2: p. 93-106.
- [3] Field, R.W.J.C.r.p.f.U.E.P.A., Office of Radiation and I. Air, Climate change and indoor air quality. 2010: p. 1-15.
- [4] Lin, B., et al., Analyzing the relationship between human behavior and indoor air quality. 2017. 6(3): p. 13.
- [5] Dahmen, J., et al., Activity learning as a foundation for security monitoring in smart homes. 2017. 17(4): p. 737.
- [6] Ali, W., et al. IoT based smart home: Security challenges, security requirements and solutions. in 2017 23rd International Conference on Automation and Computing (ICAC). 2017. IEEE.
- [7] Hall, F., et al., Smart homes: security challenges and privacy concerns. 2020.
- [8] Sharma, P., P.J.I.R.J.o.E. Kantha, and Technology, Blynk'cloud server based monitoring and control using 'NodeMCU. 2020. 7(10): p. 1362-1366.
- [9] Fadiga, K., et al. To do or not to do: finding causal relations in smart homes. in 2021 IEEE International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS). 2021. IEEE.
- [10] Philipose, M., et al., Inferring activities from interactions with objects. 2004. 3(4): p. 50-57.
- [11] Taiwo, O., A.E.J.S. Ezugwu, and C. Networks, Internet of things-based intelligent smart home control system. 2021. 2021: p. 1-17.
- [12] Krishnan, N.C., D.J.J.P. Cook, and m. computing, Activity recognition on streaming sensor data. 2014. 10: p. 138-154.



-
- [13] Crandall, A.S., Behaviometrics for multiple residents in a smart environment. 2011: Washington State University.
- [14] Cook, D.J.J.I.i.s., Learning setting-generalized activity models for smart spaces. 2010. 2010(99): p. 1.
- [15] Candamo, J., et al., Understanding transit scenes: A survey on human behavior-recognition algorithms. 2009. 11(1): p. 206-224.
- [16] Tapia, E.M., S.S. Intille, and K. Larson. Activity recognition in the home using simple and ubiquitous sensors. in International conference on pervasive computing. 2004. Springer.
- [17] Dawadi, P.N., et al., Automated cognitive health assessment from smart home-based behavior data. 2015. 20(4): p. 1188-1194.
- [18] Yang, A.Y., et al., Distributed recognition of human actions using wearable motion sensor networks. 2009. 1(2): p. 103-115.
- [19] Singla, G., et al., Recognizing independent and joint activities among multiple residents in smart environments. 2010. 1: p. 57-63.
- [20] Crandall, A.S., D.J.J.J.o.A.I. Cook, and S. Environments, Coping with multiple residents in a smart environment. 2009. 1(4): p. 323-334.
- [21] Wang, H.-C. Modeling idea generation sequences using hidden Markov models. in Proceedings of the annual meeting of the cognitive science society. 2008.
- [22] Degottex, G., P. Lanchantin, and M. Gales, A pulse model in log-domain for a uniform synthesizer. 2016.
- [23] Thomas, B.L. and A.S. Crandall. A demonstration of PyViz, a flexible smart home visualization tool. in 2011 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops). 2011. IEEE.
- [24] Du, G., et al., Novel automated K-means++ algorithm for financial data sets. 2021. 2021: p. 1-12.
- [25] Nath, A. and K.J.N. Subbiah, The role of pertinently diversified and balanced training as well as testing data sets in achieving the true performance of classifiers in predicting the antifreeze proteins. 2018. 272: p. 294-305.
- [26] Guggilam, S., V. Chandola, and A.J.a.p.a. Patra, Anomaly detection for high-dimensional data using large deviations principle. 2021.
- [27] Erhan, L., et al., Smart anomaly detection in sensor systems: A multi-perspective review. 2021. 67: p. 64-79.
- [28] Dahmen, J., D.J.J.A.T.o.I.S. Cook, and Technology, Indirectly supervised anomaly detection of clinically meaningful health events from smart home data. 2021. 12(2): p. 1-18.
- [29] Bridges, L.J.I., Communication and Society, Infrastructural obfuscation: unpacking the carceral logics of the Ring surveillant assemblage. 2021. 24(6): p. 830-849.
- [30] Raja Waseem Anwar, Kashif Naseer Qureshi, Wamda Nagmeldin, Abdelzahir Abdelmaboud, Kayhan Zrar Ghafoor, Ibrahim Tariq Javed, Noel Crespi" Data Analytics, Self-Organization, and Security Provisioning for Smart Monitoring Systems" Sensors (Basel). 2022 Oct; 22(19): 7201.
- [31] Y. Xiao, Y. Shen, J. Liu, L. Xiong, H. Jin, and X. Xu, "Dphmm: Customizable data release with differential privacy via hidden markov model," arXiv , 2023.
- [32] Nazar Waheed, , Fazlullah Khan, , Spyridon Mastorakis, , Mian Ahmad Jan, , Abeer Z. Alalmaie, Priyadarsi Nanda, "Privacy-Enhanced Living: A Local Differential Privacy Approach to Secure Smart Home Data" Computer Science and Engineering Department, University of Notre Dame, USA, 2023
-