Anomaly Detection in Smart Homes Using Bayesian Networks

Sasan Saqaeeyan[1], Hamid Haj Seyyed javadi[1, 2*] and Hossein Amirkhani[1, 3]
[1]Department of Computer Engineering, Borujerd Branch, Islamic Azad University, Borujerd, Iran.
[sasan.sagha@gmail.com]
[2]Department of Mathematics and Computer Science, Shahed University, Tehran, Iran.
[h.s.javadi@shahed.ac.ir]
[3]Computer Engineering and Information Technology Department, University of Qom, Qom, Iran.
[amirkhani@qom.ac.ir]
*Corresponding author: Hamid Haj Seyyed javadi

Received October 23, 2018; revised July 1, 2019; accepted January 20, 2020; published April 30, 2020

Abstract

The health and safety of elderly and disabled patients who cannot live alone is an important issue. Timely detection of sudden events is necessary to protect these people, and anomaly detection in smart homes is an efficient approach to extracting such information. In the real world, there is a causal relationship between an occupant's behaviour and the order in which appliances are used in the home. Bayesian networks are appropriate tools for assessing the probability of an effect due to the occurrence of its causes, and vice versa. This paper defines different subsets of random variables on the basis of sensory data from a smart home, and it presents an anomaly detection system based on various models of Bayesian networks and drawing upon these variables. We examine different models to obtain the best network, one that has higher assessment scores and a smaller size. Experimental evaluations of real datasets show the effectiveness of the proposed method.

Keywords: Smart homes, Sensory data, Anomaly detection, Bayesian networks

1. Introduction

Nowadays, the tendency to live alone and independently is increasing. For patients and elderly people who have this type of lifestyle, there are concerns over depression and sudden incidents such as falling, heart attack, and unconsciousness [1]–[3]. The easiest solution to this problem is hiring a nurse, but the excessive cost of this method and invasion of patients' privacy do not allow using this method all the time. Rapid advances in technologies related to sensors and machine learning algorithms and using remote control systems for healthcare have provided another viable option to solve this problem. These systems are examples of smart homes. Today, because of the development of the Internet and the related technologies, the idea of smart home has moved from the design stage in laboratories to common people's lives [4],[5]. Smart homes have a multi-layered architecture as shown in **Fig. 1** [6]. It includes four layers: physical layer (environment, objects, and residents), communication layer (wired and wireless sensor network), data processing layer (data storage and machine learning algorithms), and interface layer (a software such as a mobile phone application). Sensors get the residents' activities and the environment state, and transfer sensory data to the data processing layer on a server, where the data is analyzed. The users get the results (such as alarms) and interact with the smart home via an interface software.

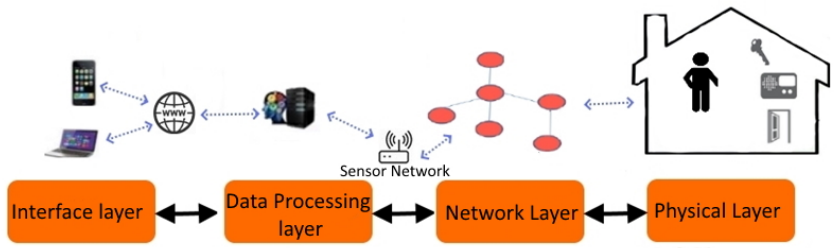

Fig. 1. Multi-layered architecture of a smart home [6]

Main applications of smart homes are automation tasks aimed at reducing energy consumption and increasing comfort at home, activity recognition for a variety of purposes such as activity reminders for Alzheimer's patients, remotely monitoring people's health by controlling the vital signs of an individual, and improving security to prevent adverse events [4], [7]. For security improvement at smart homes, different aspects and threats should be considered: detection of intruders (e.g, detection of unusual entry and prevention of entry when an anonymous person intends to enter the home), detection of health events (e.g, recognition of an elderly person who lies on the floor without any movement), and detection of physical changes in the environment (e.g, detection of high temperatures or specific gases in the air when part of a building block is burning, and warning other occupants to leave the complex) [5], [7]. Such systems use previously collected sensing data about an occupant's interaction with home furniture and appliances, assess these data using a danger detection algorithm, and take appropriate action based on the level of danger [8]. However, these homes, along with the mentioned benefits and applications, face challenges such as high installation and maintenance costs, invasion of occupants' privacy, unauthorized entry into the home software system and raising the temperature of the heater and causing of house fire, or failure in the smart home hardware, which can bring about unwanted results [4], [5], [9]–[11].

In recent years, many research projects on smart homes have focused on understanding occupants' behavior from sensory data or categorizing them [12]–[15]. Moreover, some studies have been conducted to detect anomalies and predict activities and events [16], [17]. In the present paper, we want to detect anomalies in the pattern of life of an individual in a smart home in order to detect health events.In this paper, anomaly or outlier is defined as a datapoint or an object that is different from the rest of datapoints or other objects. Along the same line, the term normal is used to describe an object or datapoint not significantly different from others [18]–[20]. Anomalies can be classified into three categories: point anomaly, contextual anomaly, and collective anomaly. Point anomaly is a set of datapoints with considerable deviation from the rest of the data, and its major challenge is establishing an appropriate criterion for calculating the deviation [21]. For example, staying in a bathroom for a longer-than-normal period of time can be classified as a point anomaly. Contextual anomaly is a set of datapoints or objects fundamentally different from others in a specific context [21]. This means that point anomaly is a subcategory of contextual anomaly whose implicit context is considered null. As a real-life example of contextual anomaly, sleeping in on an off day is considered normal, but sleeping in on a workday is regarded as an anomaly. Or eating food in the kitchen is normal, but eating food in the bathroom is considered as an anomaly. Collective anomaly is a subset of objects and datapoints that fundamentally deviates from the whole dataset [21]. For instance, if the sensors of the kitchen cabinet and the oven switch on and following that, the sensors of the bedroom and the workout room switch on, then this sequence of events is considered as an anomaly. However, each of these events can be regarded as normal events on its own.

Various studies have been performed on anomaly detection in smart homes through the use of different kinds of sensors and various algorithms. Most of these research attempts are concerned with the detection of anomalous activities [8]. However, these methods suffer from an extra error during activity recognition. Some studies also use probabilistic graphical models as a tool to model normal and abnormal behavior [16], [17]. A person performs his or her daily activities habitually and on a regular basis or based on a causal relationship, such as taking a bath after an exercise to cleanse the body. In addition, the behavior of human can include substantial amount of uncertainty. Bayesian networks are suitable tools to analyze causal relationship in the presence of uncertainty [22], [23]. Various studies with different backgrounds have used Bayesian networks for anomaly detection [24]–[27]. The aim of the present paper is to detect anomalies at the right time to avoid dangerous incidents when a person interacts with home objects. It sets out to improve anomaly detection in smart homes by extending functionalities to analyze raw sensory data and obtaining appropriate directed probabilistic graphical models (Bayesian networks). The idea is to calculate the probability of the current sensor that switches on so that the model can generate an alarm if the probability is lower than a certain level. For this purpose, we develop different models of Bayesian networks with different sizes and evaluate them to obtain the most appropriate network with suitable causal relationships between random variables. The present study is innovative in that it detects anomalies at smart homes by modeling and training sensory data through the use of Bayesian networks. Moreover, finding the optimal structure of Bayesian networks leads to higher assessment metrics and a smaller size by presenting an algorithm to remove unnecessary random variables. (We consider first-order Markov property and training and evaluate Bayesian networks using different subsets of random variables).The rest of the paper is organized as follows. Section 2 reviews related work on anomaly detection in smart homes. Section 3 briefly explains the steps involved in developing a

Bayesian network. The proposed method is described in Section 4 and is experimentally evaluated and discussed in Section 5. Finally, Section 6 concludes the paper.

2. Related work

Various methods have been used for anomaly detection in smart homes. These approaches can be categorized according to different criteria such as sensor types, the algorithms used, and the level of data analysis.

2.1 Categorizing anomaly detection methods based on sensor types

Zhu et al. [16] categorized anomaly detection methods in smart homes into three classes depending on sensor types: visual-based sensors, wearable sensors, and distributed sensors.

Visual-based methods: These can extract useful information such as location and the state of the face from the images taken to recognize patterns and detect anomalies. The main problem with this approach is the invasion of individual's privacy [16], [28].

Methods using wearable sensors: Wearable sensors are widely used in health systems and sports. Depending on their applications, they are attached to a specific part of the body to monitor health and location and to detect motion. Unlike visual-based methods, they do not have the blind spot problem, but wearing the sensor for a long time is difficult for occupants [16], [29].

Methods based on distributed sensors: With these methods, sensors such as pressure and motion sensors are attached to different parts of the home to monitor the environment and the activities of occupants. These sensors are the most used sensors in smart homes because of their reliability, low costs, and ease of installation [6], [8], [16], [29], [30].

2.2 Categorizing anomaly detection methods based on the algorithms used

Anomaly detection can also be divided up into several categories according to the algorithms employed [8].

Statistical methods such as histogram: Song et al. [31] consider places where an occupant has performed a certain activity a lot of times. Behavioral changes and anomalies are detected according to daily histogram changes. This method shows the frequency at which an activity is repeated. However, it cannot show the interdependence between the features.

Probabilistic models: Cardinaux et al. [32] extracted the characteristics of activity level, activity duration, the number of sensors participating in the activity, and the starting time of an activity and trained a Gaussian mixture model (GMM) based on these characteristics. The GMM can takes account of all the features for modeling, but it does not fit when the work comes with a large number of dimensions or features.

Neural networks: Novák et al. [33] used self-organizing map (SOM) for clustering and modeling the activities. After the training process, activities that deviate from the cluster group are detected as an anomaly. This study only considers activities that extend beyond 15 minutes, while there are many activities, such as brushing, that may take less than 15 minutes. Moreover, it should be noted that in a neural network, users cannot add new rules to the hidden layer.

Semantic and rule based methods: Hoque et al. [34] pointed out that failure to use semantic rules is one of the main reasons for false positive diagnosis. In this approach, specific features are defined for activities on different days of the week. Then, these features

are adjusted according to the day of activity and the order in which activities are performed, and also based on some health rules such as brushing before going to bed. These methods allow human rules to be added and cause a reduction in the rate of false alarms. However, they do not manipulate noisy data appropriately.

Hela et al. [35] tried to detect the early risk of occurring an anomaly in the environment using residents' activities. For this purpose, they extracted reasons for the occurrence of abnormal activities using causal association rules. The extracted reasons are used in Markov logical networks to detect the risk of an anomaly in real time. However, they got different answers in different time windows.

Using a combination of algorithms: Yin et al. [36] used a two-phase anomaly detection model. In the first phase, the model detects normal activities using a one-class support vector machine (OCSVM) so that most of the normal activities would be filtered. The remaining suspicious activities are sent to another module, i.e, kernel nonlinear regression (KNLR) that uses the hidden Markov model (HMM), for more investigation. In the second phase, KNLR calculates the occurrence probability of the sequence of activities via the HMM. If the probability is lower than the threshold, this probability is detected as an anomaly. This study reduces the rate of false alarms but does not focus on the false negative rate. In the hybrid approach proposed by Forkan et al. [37], anomalies in the sequence of daily activities are detected using the HMM, behavioral changes in an individual's daily activities pattern (changes in the time, time duration or number of times of routine activity in a day) are detected using normal distribution, physiological changes in the vital signs are detected via statistical analysis, and then, the output of these methods is sent to a fuzzy module so that the final decision about anomalies can be made. This study detects various aspects of anomaly in an individual's life and uses a fuzzy system to combine different anomalies. It reacts proportionately to the level of anomalies and reduces false alarms. However, in hybrid methods, the error from each of the methods employed will contribute to the final error in anomaly detection.

2.3 Categorizing anomaly detection methods based on the level of data analysis

The present study considers a new classification for anomaly detection in smart homes. Here, methods of anomaly detection are categorized into two classes: detection of anomaly in activities and detection of anomaly in sensory data.

Detection of anomaly in activities: These methods are commonly performed in three steps: low-level analysis, recognition of activity, and detection of anomalous activity[8].

Zhu et al.[16] proposed a method to detect anomalies in four contexts of location, time, order, and duration of activities. They used the first-level dynamic Bayesian network with a known structure. In these methods, the final error increases due to the combination of activity recognition error and anomaly detection error.

Detection of anomaly in sensory data: These methods involve two steps: analysis of sensory data and detection of anomaly in the prepared sensory data. Recognition or discovery of activity is omitted in these approaches. Park et al. [38] defined episodes consisting of several events and including information about occupants' location and time with the purpose of detecting abnormal episodes.

Shin et al. [39] used infrared motion sensors and obtained three features: activity level, which calculates the instances of motion sensed by sensors for each location in the home; motion level; and threshold of non-response interval, which shows the interval between

occupants' motions. They used support vector data description, a type of support vector machine (SVM), with a Gaussian kernel to detect anomaly and normal pattern.

Ordóñez et al. [24] recognized the behavior patterns of occupants by means of the Bayesian statistic. They defined three probabilistic features: sensor activation likelihood (to detect individual health), sensor sequence likelihood (to recognize consciousness), and sensor duration likelihood (to determine the physical condition of an individual). The probability of each feature was calculated using Bernoulli, multinominal, and Gaussian distributions, respectively. The main advantage of this method is that it uses prior knowledge and efficiently and quickly combines this knowledge with the new sensory data via Bayesian theory. This study only detects specific aspects of anomaly.

The aforementioned methods have their advantages and disadvantages. At the level of activity, anomaly detection has activity recognition error and anomaly detection error. Probabilistic graphical models have been used in a few studies [16], [17], [32], but, to the best of our knowledge, no attempt has been made to obtain the best or most appropriate graph for the models. The present paper focuses on these cases and aims to determine the best structure of probabilistic graphical models by training different Bayesian networks through the use of sensory data.

3. Bayesian networks

Bayesian networks or belief networks can efficiently solve the problem of uncertainty in artificial intelligence. These networks are directed acyclic graphs whose nodes are random variables, each with a conditional probability distribution based on its parent. Graph edges represent a kind of causal relationship between parents and their children. In such networks, the probability distribution of n variables is calculated using Eq. (1) [40], [41]:

$$P(X_1, X_1, \dots, X_n) = \prod_i P(X_i | Parent(X_i)) \quad (1)$$

The construction of a Bayesian network depends on factors such as complete or incomplete data and known or unknown network structure. Building a Bayesian network with complete data and unknown structure entails three steps: structure learning, parameter learning, and inference [41].

3.1 Structure learning

Network structure indicates how nodes interact with each other. Structure learning is performed in score-based or constraint-based ways [42].

A score-based algorithm assigns scores to each candidate structure and attempts to obtain the maximum score structure using a heuristic search algorithm. There are different scoring functions such as Bayesian Dirichlet (BD), K2 Metric, and Bayesian Dirichlet with equivalent uniform prior (BDeu) [40]. Recently, it is suggested to use experts' knowledge besides data to obtain a more robust scoring of the structures. As the search strategy, we use the K2 algorithm, which is a greedy search exploiting the order of the nodes to make the search space smaller [22], [43]. In the well-known K2 algorithm, inputs are the following: a set of nodes, an arrangement of the nodes, an upper bound u on the number of parents that a node may have, and a database D that consists of m cases. A printout of the parents of each node constitutes the output for each node.

The scoring function of this algorithm is known as K2, which is defined as Eq. (2):

$$g(x_i, \pi i) = \prod_{j=1}^{q_i} \frac{r_i - 1}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}! \quad (2)$$

The above equation is defined for a Bayesian network that has n discrete random variables $x_1, x_2, \ldots x_n$. Furthermore, x_i has r_i values $v_{i1}, v_{i2}, \ldots v_{ri}$.

In Eq. (2), πi denotes the set of parents of node xi, qi is the list of all possible instantiations of the parents of xi in database D, ri indicates the list of all possible values of the attribute xi, N_{ijk} is the number of cases (i.e, instances) in D, where the attribute xi is instantiated with its kth value, the parents of xi in πi are instantiated with the jth instantiation in qi, and $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$, which is the number of instances in the database where the parents of xi in πi are instantiated with the jth instantiation in qi.

The constraint-based approach uses the independence test to define constraints on edges and tries to find the structure satisfying these constraints [44].

3.2 Parameter learning

After learning the Bayesian network structure, we should learn network parameters. Network parameters are the conditional probability distribution of the variables given their parents. Parameter learning is the process of estimating these distributions based on the training data. As in statistical parameter estimation, there are two main approaches to Bayesian network parameter learning, maximum likelihood estimation and Bayesian estimation. The main difference between these approaches is that in the Bayesian estimation, we use a probability distribution, $P(\theta)$, as a prior belief about the environment before observing the data [41].

3.3 Inference

Having the network structure and parameters, we can calculate the probability of each query given the state of a subset of network variables. There are two approaches to the inference: exact and approximate inferences. The exact inference uses both the conditional independence assertions the network makes and the associated factorization of the joint distribution to perform an effective inference [41]. The approximate inference attempts to obtain an approximation of the query probability through methods such as sampling from the network and approximating the original network by a simpler one [45], [46].

4. The proposed method

Based on the Bayesian network models, we will analyse sensory data to detect anomalies in smart homes to enhance the safety and health of the occupants. We propose a multi-phase architecture and define a set of random variables based on the prepared sensory data. The proposed method builds all subsets of the random variables with a size larger than one and ensures that the ID of the current activated sensor is a member of all these subsets. We want to calculate the probability of the current activated sensor and to estimate a threshold to detect anomalies. Different Bayesian network structures are trained on the basis of these subsets of random variables. The purpose is to obtain the best graphical model with the highest evaluation score.

The assumptions which are used in the proposed method are as follows:

- *Assumption 1*: There are no sensor failures.

As the purpose is to detect anomalies in the behavior of the occupant rather than the sensors themselves, it goes without saying that sensor failure can affect the results.

- *Assumption 2*: An event occurs when the occupant moves between or interacts with two sensors (the first sensor switches on and then the second one switches on). We did not consider more than two sensors because we considered the first-order Markov condition for solving the problem.
- *Assumption 3:* The data gathered from the individual's daily life are normal. We made this assumption because we have a one-class problem as the collected data are related to the everyday life of the individual.

The proposed method works in four main phases (**Fig. 2**): pre-processing, model learning, model evaluation, and anomaly detection.

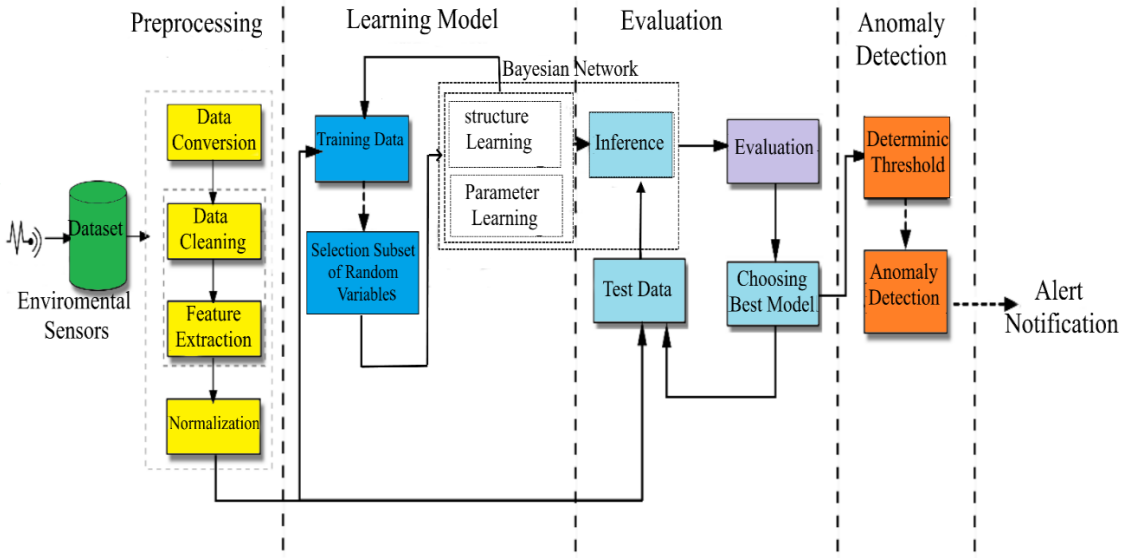

Fig. 2. The proposed architecture for anomaly detection in smart homes

4.1 Pre-processing

This phase consists of reading sensory data, data cleaning, defining random variables, and normalization. We read raw sensory data that consist of time and date attributes and the on/off states of the sensors. Since we cannot consider an exact time for most daily activities, we should consider a few minutes earlier or later as tolerance. Therefore, we map the minute part of the time feature into discrete values as $\lfloor m/k \rfloor + 1$, where m is the minute part of the time when a sensor changes its state, k is an integer in $\{15,20,30\}$, and $\lfloor . \rfloor$ shows the floor function. For example, if time is 11:31, m is 31. For x = 30, time will map to 112. Missing data and unused features are eliminated. In training the data, the record from a sensor would be deleted if the sensor turned on but did not turn off. We defined a variable called "Day." In the early experiments, at the time of building the Bayesian network structure, this variable was unconnected with the other nodes. Thus, we removed the feature because it did not have any influence on the results and inference. In addition, we divided the home's map into several parts and added sensor location as a feature. Random variables were defined based on the prepared data as in **Table 1**.

Table 1. Random variables used in our system

Variable	Description
Id1	Identification number for the current activated sensor
Stime	The switching on time of the current activated sensor
Etime	The switching off time of the current activated sensor
Location1	Current activated sensor's location
Id2	Identification number for the previous activated sensor
Stime2	The switching on time of the previous activated sensor
Etime2	The switching off time of the previous activated sensor
Location2	Previous activated sensor's location
Day	Day of the week

4.2 Model learning

This phase includes two parts: structure learning and parameter learning. For structure learning, we use the K2 algorithm [22]. To obtain the best structure from the data, we consider all subsets of the random variables with a size larger than 2 and consisting of the current activated sensor. Then, the network structures are built on each subset using the K2 algorithm. Parameter learning is performed on each of these structures based on the training data. After this phase, we have different Bayesian networks that should be evaluated to obtain the best structure.

4.3 Model evaluation

Model evaluation is performed in this phase to obtain the best model. We use the Bayesian networks in an inference task and select the network with the best performance in this task.

- *Inference*: Each test sample is given to the trained models as a query such as

  p (Id1 | evidence), where evidence constitutes the values of the parents of the current activated sensor with identifier Id1. In other words, the probability that the current activated sensor switches on according to the given evidence is calculated based on the trained models.

- *Evaluation*: This is to obtain the best model and the suitable threshold. For this purpose, we use the area under curve (AUC) criterion, which presents the average conditions of the model at various thresholds, and a graph weight criterion (the graph size that is the sum of the nodes and the graph edge).

4.4 Anomaly detection

The chosen model can be simply used for anomaly detection. When an event occurs, its probability is calculated based on this model, and if the probability is lower than the determined threshold, it is detected as an anomaly, and an alarm notification is generated. To determine the threshold, we define 100 values (i.e, 100 thresholds) between 0 and 1, and calculate the F score (evaluation criteria in Section 5.1 below) to compare results of tests using these thresholds. The threshold with the highest value for the F score will be selected as the final threshold for the model.

4.5 Updating the database

Upon detection of normal or abnormal events, if the event is normal, it will be stored in a dataset. However, if the event is abnormal and has regularly repeated itself on the previous days and the occupant confirms this, it is added to the normal dataset (due to the change in the person's lifestyle). Otherwise, it is stored in the abnormal dataset. The database is updated monthly.

4.6 The pseudo-code of the proposed method

Finally, we present two pseudo-codes to show the important parts of the proposed method. Algorithm 1 is related to training the probabilistic models and obtaining the best model, while Algorithm 2 presents the anomaly detection part.

```
Algorithm 1 Model Tuning Algorithm
Input: Training Dataset(D_train), Testing Dataset(D_test)
Output: Model, Threshold
1:FOR each subset in RandomVariableSet
2: FOR each CV_train, CV_test in ThisRoundDatasets[subset]
3:       Train_subset = Dtrain(Cv_train)
4:       Test_subset = Dtest(Cv_test)
5:       //Structur learning using K2 meter and hill climbing search
6:       model = StructureLearning(Train)
7:       //Parameter learning using bayesian estimation algorithm
8:       normalModel = ParameterLearning(model, Train)
9:       //Inference for all events in test dataset
10:      probabilities = Inference(normalModel, Test)
11:            //Compute AUC_s, F1-scores, Graphs wight
12:      AUC_i, F_i, W_i = evaluate(model, probabilities)
13: //Find maximum AUC, F, Graph-wight
14: model, threshold = Find-best(AUC, F, W)
15: Return model, threshold
```

Algorithm 1 receives the training and testing data and additional knowledge about the variables as input. Each step of the outer loop selects a subset of random variables. Structure and parameter learning for each subset are performed in the inner loop. The probability of each query of test data is calculated by the inference module. The AUC, F-measure, and graph weight criteria are calculated to obtain the best model. In the end, the *Find-best* function returns the best model and the best threshold.

Algorithm 2 receives new events as input and reports anomalies using the selected model and threshold.

```
Algorithm 2 Anomaly Detection
Input: Model, Threshold, Event
Output: Notification
1: IF GetProbability(Model, Event) < Threshold THEN
2:  Notification = True
3: ELSE
4:  Notification = False
5: END IF
6: Return Notification
```

5. Experimental evaluation

This section presents the experimental results to evaluate the proposed method and other alternative approaches using an actual dataset [47]. The Markov property will be verified for the problem, and the proposed model will be tested using another validated dataset [48].

5.1 Experimental setup

We use the data from a public accessible dataset (Kasteren dataset) [47] that has been used and referred in many studies [14], [15], [49]. It is real data gathered from a single occupant in a three-bedroom smart home apartment. They have built a network of distributed sensors to monitor all parts of the home. Sensors have been installed in different parts of the home, such as the kitchen cabinet, bathroom, entrance door, and beds, as depicted in **Fig. 3**. The red signs mark the locations of the 14 installed sensors. All are digital sensors and use the RFM DM 1810 Kit.

Fig. 3. The location of the sensors in the home [47]

There are 1,318 records in this dataset. Sensory data are collected from the daily life of an occupant having lived alone for 28 days. Every record of the raw dataset has three fields: start time (i.e, when the sensor switches on), end time (when the sensor switches off), and ID (sensor id), which is compatible with real smart homes including distributed sensors. A segment of the records of the dataset is shown in **Fig. 4**.

Start time	End time	ID
25-Feb-2008 09:37:17	25-Feb-2008 09:38:02	4
25-Feb-2008 09:49:23	25-Feb-2008 09:53:28	13
25-Feb-2008 10:02:28	25-Feb-2008 10:12:42	5
25-Feb-2008 10:19:06	25-Feb-2008 16:55:38	1
25-Feb-2008 17:00:31	25-Feb-2008 17:01:34	4

Fig. 4. Kasteren dataset records [47]

We read sensory data in a one-minute time slot. If a sensor switches on and off in sequence in this time slot, we combine these records together and convert them into one record. For this record, the time when the current sensor first switches on in the time slot is

considered its switching on time, and the time when it last switches off in the time slot is regarded as its switching off time. The sensory data gathered from daily life are considered normal data. We manually generated 100 abnormal data samples for the Kasteren dataset. After the dataset was checked, we generated the abnormal records based on unusual behavior and statistical information of the dataset that was different from the rest of the dataset and had anomaly in time, location, sequence, or interval time between the switching on of the sensors; for example, records that had not previously occurred in the locations at a specific time, or records from an unusual or rare sequence of the switching on of the sensors. Because of complexity of generating manually and identifying the abnormal records that may exist in the real data, we used algorithm proposed by Shin et al. [39] to generate other synthetic abnormal samples randomly .We split up the dataset into a training set and a test set. We used ten-fold cross-validation to train models on the training set, and to tune its parameters (thresholds) and evaluate the performance of the proposed model using the test set.

***Evaluation criteria*:**

We use the standard confusion matrix to measure classification performance. The confusion matrix consists of the following four cells:

True Positive (TP): The number of correctly detected abnormal records

True Negative (TN): The number of normal records correctly considered as normal

False Positive (FP): The number of normal records incorrectly detected as abnormal

False Negative (FN): The number of abnormal records incorrectly considered as normal

The used classification evaluation criteria are as follows [21], [50]:

- Recall = TP / (TP + FN)
- Precision = TP / (TP + FP)
- F1 = (2 × Precision × Recall) / (Precision + Recall)
- AUC: The area under the Receiver Operating Characteristics (ROC) curve, where ROC is a two-dimensional curve that shows the TP rate against the FP rate.

5.2 Experimental results

The proposed system is implemented in python. There are a number of open-source python libraries to use with probabilistic graphical models and Bayesian networks. We use Pgmpy which has many algorithms for structure learning and inference [45]. Four other approaches are compared with the proposed model: two manual models with known structures (Manual-1 [16], Manual-2), OCSVM [36], and naive Bayes.

5.2.1 The proposed method

As explained in section 4, we trained all structures based on the subsets of random variables and chose the best model based on the results of the evaluations. Bayesian networks have limitation to deal with continuous data and we discretized continuous variables (the time of switching on/off the sensors). The cost of finding the appropriate structure is high, but it does not make a problem during the usage of the system since the model is trained just once. **Fig. 5** shows a number of such learned structures in the Kasteren dataset.

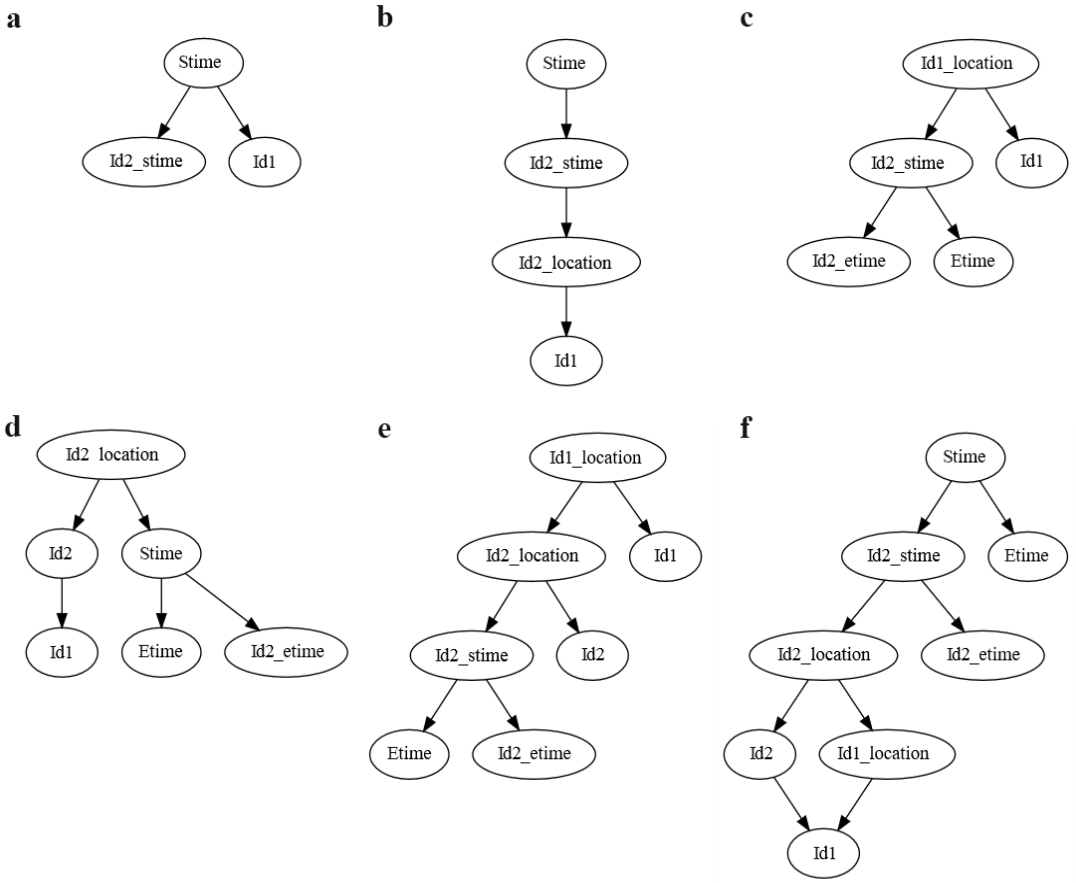

Fig. 5. Examples of the trained structures in the Kasteren dataset

Table 2 summarizes the experimental results from the models shown in **Fig. 5**. The results indicate that the location has a strong influence on the current activated sensor. Therefore, for the final ordering of variables in K2 algorithm, we put the sensor location variable after the switching on/off time of sensors. In addition, we put the Id1 and Id2 at the end of the variable order list. According to **Table 2**, the structure of **Fig. 5-f**, which has 8 nodes, gets the best result.

Table 2. Experimental results for the trained structures

Model	Recall	Precision	F1	AUC	Threshold
3-node (a)	0.970	0.527	0.682	0.708	0.31
4-node (b)	0.962	0.536	0.688	0.759	0.22
5-node (c)	0.920	0.747	0.824	0.932	0.27
6-node (d)	0.952	0.629	0.757	0.851	0.33
7-node (e)	0.932	0.715	0.809	0.932	0.41
8-node (f)	0.938	0.968	0.953	0.987	0.10

5.2.2 Two manual models with known structures

We use two Bayesian network models whose structures are known. The first model (Manual-1) is a first-order dynamic structure based on [16] (**Fig. 6(a)**). In this structure, the current activated sensor depends on the previous sensor (Id2). The structure of Manual-2 is shown in **Fig. 6(b)**. In this structure, the current activated sensor relies on the previous activated sensor (Id2), its location (Id2_location), and its activation time (Id2_stime).

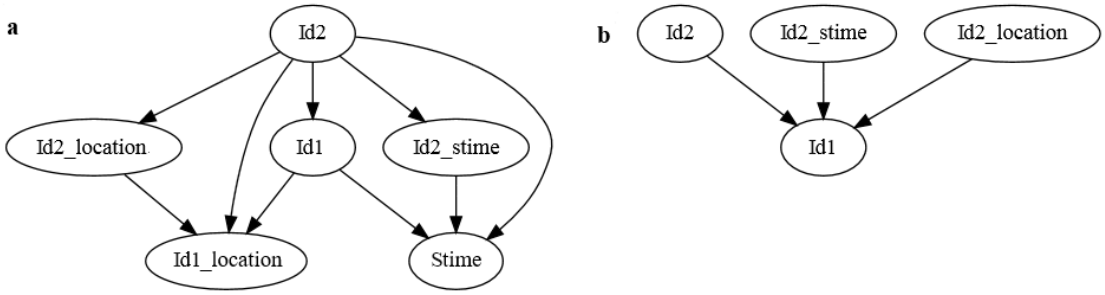

Fig. 6. The structures of Manual-1 (a) and Manual-2 (b)

5.2.3 OCSVM

OCSVM is a kind of SVM that uses one class to train data. It learns the classification boundary for the normal class, and any data outside this class is detected as an anomaly. For OCSVM, we defined the same features as those used in the propounded method and set the values of nu and gamma to 0.14 and 0.01, respectively.

5.2.4 Naive Bayes

Naive Bayes is a Bayesian network with a strong independence assumption: all the features are independent given the class variable (Id1 in our work).

5.2.5 Comparison of different methods

Table 3 shows the experimental results comparing the proposed method with the competing approaches, and **Fig. 7** shows the ROC curves of different methods. According to these results, the idea of using Bayesian networks to detect anomalies based on sensory data in smart homes is a promising idea and can outperform the competing approaches. This is because the network, employed in the proposed approach, is suitable for uncertainties and to analyze human behavior and determine the best network.

Table 3. Comparison of different methods

Method	Recall	Precision	F1	AUC	Threshold
Proposed Model	0.93	0.96	0.95	0.98	0.10
OCSVM	0.96	0.77	0.85	0.88	NA*
Naive Bayes	0.84	0.91	0.88	0.91	0.02
Manual-1	0.92	0.85	0.88	0.92	0.27
Manual-2	1	0.82	0.90	0.81	0.09

* NA: Not Applicable

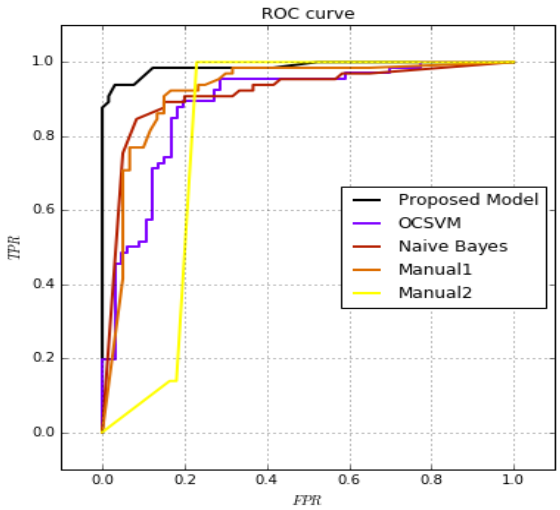

Fig. 7. The ROC curves of the models under investigation

Table 4 shows the calculated confusion matrix for the proposed model. A few normal records, which rarely occurred in the dataset, detected false positive. And a number of abnormal records, which have abnormality in their time, detected false negative.

Table 4. The confusion matrix of the proposed model

	Predicted class		
Actual class		Positive	Negative
	Positive	93	7
	Negative	3	97

Fig. 8 below illustrates the degree of similarity between the proposed method and the alternative approaches in the detection of abnormal and normal events. The values indicate that the proposed method is most similar to the Manual-1 method in anomaly detection, and it bears the strongest similarity to the Manual-2 method in the detection of normal events.

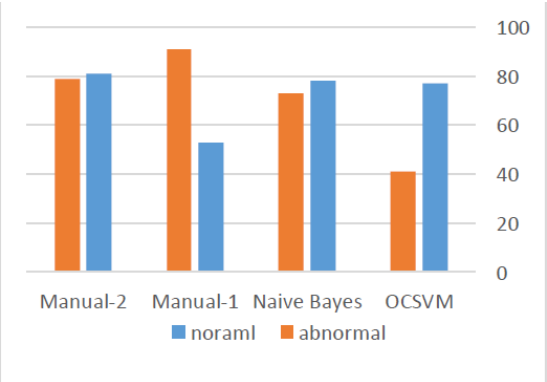

Fig. 8. The percentage of similarity between the proposed method and the alternative methods in the detection of normal and abnormal events

The proposed method detects anomalies in the events, and not in the activities, so it can detect different types of anomalies in the pattern of life. For example, changes in the basic pattern of life, such as late waking-up, can be detected because of the unusual start and end times of the events (the switching on and off of the sensors) and get a low probability. As another example, in the case of lack of mobility, the switching on time of the sensors, and also the time interval between the switching on and switching off events become unusual after a certain time slot, thereby giving them a low probability and detecting them as an anomaly. Experiments show that the proposed model makes it possible to detect all anomalies that are related to time and location and also the irregularity in the order and duration of the event.

Note that the symbolic approaches can also be used for anomaly detection in smart homes [33], [34]. They encode knowledge by adding rules, but finding the rules is expensive and time-consuming. Moreover, to use this approaches for a new setting, it is necessary to find new set of rules. Also, they cannot adapt to changes in the daily lifestyle. But the proposed model is a machine learning-based method. Thus, for applying it to a new smart home, we just need to train the model on the new data without any change to the proposed method.

Finally, our method may have two limitations. First, our work does not detect anomalies related to physiological states and disease symptoms, such as changing in the body temperature or unusually increase of heart rate (HR). The second limitation of this approach compared to the methods presented in [24], [37] is that it is unable to detect the type of anomaly and cannot classify anomalies, so it cannot define different reactions to different anomalies (i.e, normal, warning, and alert emergency).

5.3 Markov property test

To verify the Markov property of this problem, we implemented the proposed model in 100 random variable sets of the Kasteren dataset. The sets included random variables that have information from 2, 3, or 4 previous sensors. **Fig. 9** portrays some of the Bayesian networks from these sets after network training.

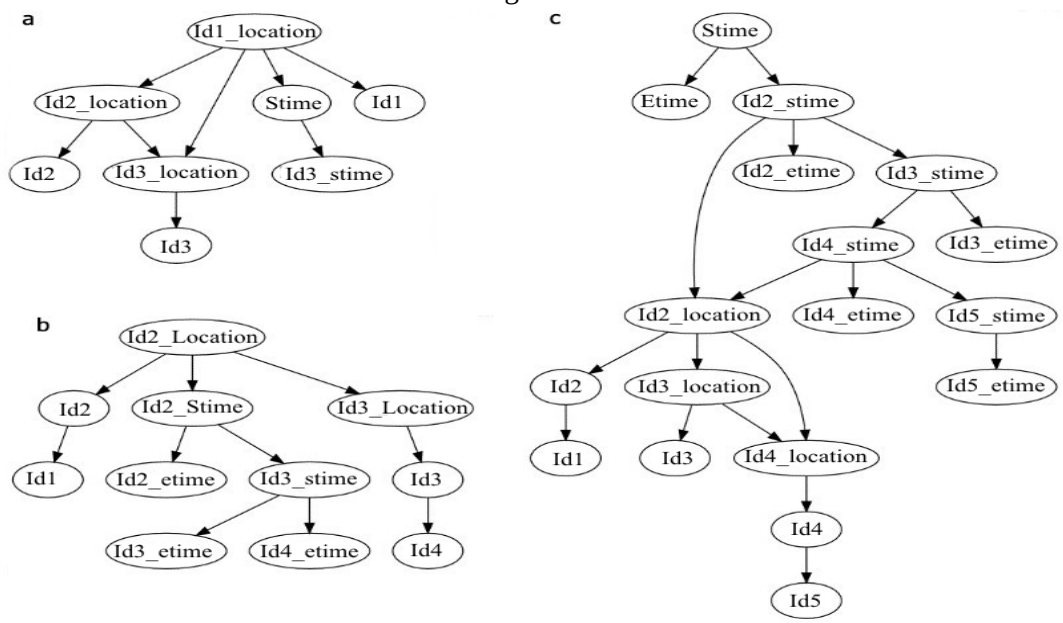

Fig. 9. Best structure for a network with (a) 3 previous sensors, (b) 4 previous sensors, and (c) 5 previous sensors.

Table 5 presents the evaluation metric of the networks shown in **Fig. 9** above. The results from these networks were worse than the evaluation results from the best structure (**Table 3**, Proposed Model).

Table 5. The Result of models that have information about previous sensors

Model	Recall	Precision	F1	AUC	Threshold
3 sensors (a)	0.84	0.98	0.90	0.94	0.05
4 sensors (b)	0.89	0.72	0.83	0.79	0.17
5 sensors (c)	0.85	0.76	0.80	0.77	0.13

It can be seen in **Table 5** that our model follows the Markov property, indicating that the proposed model only requires information of one previous sensor. Adding information from more previous sensors (more than one previous sensor) does not increase the evaluation metrics and the precision of the model. In the propounded method, all structures were created using subsets from a set of random variables, including information from two sensors (i.e, the current sensor and the previous sensor). The learning structure of these subsets are part of the search tree for situations where we want to determine the best structure using information from more than two sensors. Thus, following the Markov property prunes the search tree and causes the size of the best structure for the propounded network to be smaller.

5.4 Testing the proposed model on another dataset

In this section, we implemented the proposed model in another validated dataset called Aruba [48]. This dataset contains raw sensory data collected from the home of a volunteer adult over a period of 41 days. The sensor events are generated by 26 sensors placed at different locations in the home. Every record of the raw dataset has four fields: date and time of the sensor switches on or off, sensor id, and status of the sensor (on or off). We chose data for one month from the dataset. Random variables and the conditions of the implementation are the same as in the proposed method (Section 5.1). **Fig. 10** depicts the best structure after evaluation criteria are calculated and compared.

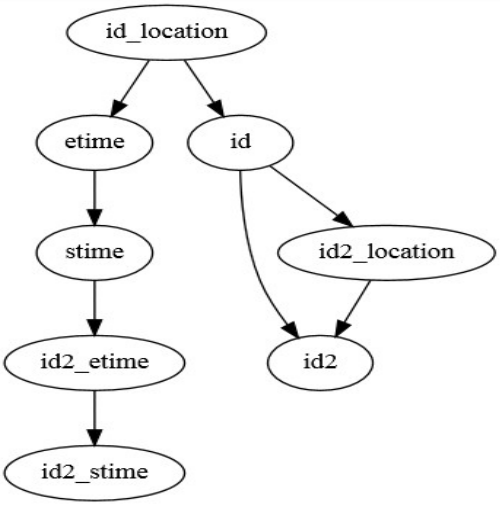

Fig. 10. Best structure for the Aruba dataset

The results of the model evaluation (**Fig. 10**) are summarized in **Table 6**.

Table 6. Experimental results for the best structure for the Aruba dataset

Recall	Precision	F1	AUC	Threshold
0.88	0.97	0.92	0.96	0.02

The best structure for the proposed method in the Aruba dataset is different from the best structure created in the Kasteren dataset due to differences in human behavior and the pattern of life. This difference indicates that we cannot always use a well-known structure [16], [17] or use best structure of one resident for another resident in predicting and analyzing human behavior in smart homes.

6. Conclusion

In this paper, we proposed a multi-phase architecture for anomaly detection using Bayesian networks to enhance safety in smart homes. We focused on finding the best network among different models with different sizes which obtained the best scores during evaluation. Experimental results using real datasets revealed the effectiveness of the proposed method. The results also indicated that using a known structure based on prior knowledge is not always the best approach to anomaly detection in smart homes. In our experiments, some methods were successful in detecting the positive labels and some in detecting the negative labels. In our future studies, we intend to improve the model through the use of ensemble learning, combining different Bayesian networks instead of selecting one. Moreover, we will investigate wearable sensors to provide the model with more useful information.

References

[1] S. Häfner et al., “To live alone and to be depressed, an alarming combination for the renin–angiotensin–aldosterone-system (RAAS),” *Psychoneuroendocrinology*, vol. 37, no. 2, pp. 230–237, 2012. Article (CrossRef Link)

[2] N. Eyal, S. A. Hurst, O. F. Norheim, and D. Wikler, *Inequalities in health: concepts, measures, and ethics*, Oxford University Press, 2013. Article (CrossRef Link)

[3] B. Risteska Stojkoska, K. Trivodaliev, and D. Davcev, “Internet of things framework for home care systems,” *Wireless Communications and Mobile Computing*, vol. 2017, Article ID 8323646, 2017. Article (CrossRef Link)

[4] B. L. R. Stojkoska and K. V. Trivodaliev, “A review of Internet of Things for smart home: Challenges and solutions," *Journal of Cleaner Production*, vol. 140, pp. 1454–1464, 2017. Article (CrossRef Link)

[5] N. K. Suryadevara and S. C. Mukhopadhyay, "Smart homes: design, implementation and issues," *Springer*, vol. 14, 2015. Article (CrossRef Link)

[6] M. Amiribesheli, A. Benmansour, and A. Bouchachia, “A review of smart homes in healthcare,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 6, no. 4, pp. 495–517, 2015. Article (CrossRef Link)

[7] J. Dahmen, D. J. Cook, X. Wang, and W. Honglei, “Smart secure homes: a survey of smart home technologies that sense, assess, and respond to security threats,” *Journal of Reliable Intelligent Environments*, vol. 3, pp. 83-98, 2017. Article (CrossRef Link)

[8] U. Bakar, H. Ghayvat, S. Hasanm, and S. Mukhopadhyay, "Activity and anomaly detection in smart home: A survey," *Next Generation Sensors and Systems, Springer*, pp. 191–220, 2015. Article (CrossRef Link)
[9] M. Theoharidou, N. Tsalis, and D. Gritzalis, "Smart Home Solutions: Privacy Issues," *Handbook of Smart Homes, Health Care and Well-Being*, pp. 67–81, 2017. Article (CrossRef Link)
[10]M. Khan, B. N. Silva, C. Jung, and K. Han, "A context-Aware smart home control system based on ZigBee sensor network," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 11, no. 2, pp. 1057–1069, 2017. Article (CrossRef Link)
[11]G.-J. Ra and I.-Y. Lee, "A Study on KSI-based Authentication Management and Communication for Secure Smart Home Environments," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 12, no. 2, pp. 892-905, 2018. Article (CrossRef Link)
[12]O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 3, pp. 1192–1209, 2013. Article (CrossRef Link)
[13]Q. Ni, A. B. García Hernando, and I. Pau de la Cruz, "A context-aware system infrastructure for monitoring activities of daily living in smart home," *Journal of Sensors*, vol. 2016, 2016. Article (CrossRef Link)
[14]J. Dahmen, B. L. Thomas, D. J. Cook, and X. Wang, "Activity Learning as a Foundation for Security Monitoring in Smart Homes," *Sensors*, vol. 17, no. 4, p. 737, 2017. Article (CrossRef Link)
[15]P. Rashidi, D. J. Cook, L. B. Holder, and M. Schmitter-Edgecombe, "Discovering activities to recognize and track in a smart environment," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 4, pp. 527–539, 2011. Article (CrossRef Link)
[16]C. Zhu, W. Sheng, and M. Liu, "Wearable sensor-based behavioral anomaly detection in smart assisted living systems," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 4, pp. 1225–1234, 2015. Article (CrossRef Link)
[17]E. Nazerfard and D. J. Cook, "CRAFFT: An Activity Prediction Model based on Bayesian Networks," *Journal of Ambient Intelligence and Humanized Computing*, vol. 6, no. 2, pp. 193–205, 2015. Article (CrossRef Link)
[18]J. Tonejc, S. Güttes, A. Kobekova, "Machine Learning Methods for Anomaly Detection in BACnet Networks," *Journal of Universal Computer Science*, vol. 22, no. 9, pp. 1203–1224, 2016.
[19]V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009. Article (CrossRef Link)
[20]S. Enno-Edzard, F. Thomas, E. Marco, F. Melina, and H. Andreas, "Modeling individual healthy behavior using home automation sensor data: Results from a field trial," *Journal of Ambient Intelligence and Smart Environments*, vol. 5, no. 5, pp. 503–523, 2013. Article (CrossRef Link)
[21]J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*, Elsevier, 2011. Article (CrossRef Link)
[22]G. F. Cooper and E. Herskovits, "A Bayesian method for the induction of probabilistic networks from data," *Machine Learning*, vol. 9, no. 4, pp. 309–347, 1992. Article (CrossRef Link)
[23]A. Abu-Samah, N. N. A. Razak, F. M. Suhaimi, U. K. Jamaludin, and J. G. Chase, "Towards Personalized Intensive Care Decision Support Using a Bayesian Network: A Multicenter Glycemic Control Study," *IEIE Transactions on Smart Signal*, vol. 8, no. 3, pp. 202–209, 2019. Article (CrossRef Link)
[24]F. J. Ordóñez, P. de Toledo, and A. Sanchis, "Sensor-based Bayesian detection of anomalous living patterns in a home setting," *Personal and Ubiquitous Computing*, vol. 19, no. 2, pp. 259–270, 2015. Article (CrossRef Link)
[25]S. Mascaro, A. E. Nicholso, and K. B. Korb, "Anomaly detection in vessel tracks using Bayesian networks," *International Journal of Approximate Reasoning*, vol. 55, no. 1, pp. 84–98, 2014. Article (CrossRef Link)
[26]L. Xiao, Y. Chen, and C. K. Chang, "Bayesian model averaging of bayesian network classifiers for intrusion detection," in *Proc. of Computer Software and Applications Conference Workshops (COMPSACW), 2014 IEEE 38th International*, pp. 128–133, 2014. Article (CrossRef Link)

[27]A. Gruber and I. Ben-Gal, "Using targeted Bayesian network learning for suspect identification in communication networks," *International Journal of Information Security*, vol. 17, no. 2, pp. 169–181, 2018. Article (CrossRef Link)
[28]D. Gutchess, N. Checka, and M. S. Snorrason, "Learning patterns of human activity for anomaly detection," *Intelligent Computing: Theory and Applications. V SPIE Orlando FL USA*, vol. 6560, pp. 65600Y–12, 2007. Article (CrossRef Link)
[29]J. Loane, B. O'Mullane, B. Bortz, and R. B. Knapp, "Interpreting presence sensor data and looking for similarities between homes using cluster analysis," in *Proc. of Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2011 5th International Conference on*, pp. 438–445, 2011. Article (CrossRef Link)
[30]P. Daneshjoo, H. H. S. Javadi, and H. R. Sharifi, "Sink Location Service Based on Fano Plane in Wireless Sensor Networks," *Journal of Communication Engineering (JCE)*, vol. 5, no. 10, 2016. Article (CrossRef Link)
[31]Y. Song, Z. Wen, C.-Y. Lin, and R. Davis, "One-Class Conditional Random Fields for Sequential Anomaly Detection," in *Proc. of International Joint Conferences on Artificial Intelligence*, pp. 1685–1691, 2013. Article (CrossRef Link)
[32]F. Cardinaux, S. Brownsell, M. Hawley, and D. Bradley, "Modelling of behavioural patterns for abnormality detection in the context of lifestyle reassurance," *Iberoamerican Congress on Pattern Recognition*, pp. 243–251, 2008. Article (CrossRef Link)
[33]M. Novák, M. Biňas, and F. Jakab, "Unobtrusive anomaly detection in presence of elderly in a smart-home environment," in *Proc. of 2012 ELEKTRO*, pp. 341–344, 2012. Article (CrossRef Link)
[34]E. Hoque and J. Stankovic, "Semantic anomaly detection in daily activities," in *Proc. of the 2012 ACM Conference on Ubiquitous Computing*, pp. 633–634, 2012. Article (CrossRef Link)
[35]S. Hela, B. Amel, and R. Badran, "Early anomaly detection in smart home: A causal association rule-based approach," *Artificial Intelligence in Medicine*, vol. 91, pp. 57–71, 2018. Article (CrossRef Link)
[36]J. Yin, Q. Yang, and J. J. Pan, "Sensor-based abnormal human-activity detection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 8, pp. 1082–1090, 2008. Article (CrossRef Link)
[37]A. R. M. Forkan, I. Khalil, Z. Tari, S. Foufou, and A. Bouras, "A context-aware approach for long-term behavioural change detection and abnormality prediction in ambient assisted living," *Pattern Recognition*, vol. 48, no. 3, pp. 628–641, 2015. Article (CrossRef Link)
[38]K. Park, Y. Lin, V. Metsis, Z. Le, and F. Makedon, "Abnormal human behavioral pattern detection in assisted living environments," in *Proc. of the 3rd International Conference on PErvasive Technologies Related to Assistive Environments*, p. 1-8, 2010. Article (CrossRef Link)
[39]J. H. Shin, B. Lee, and K. S. Park, "Detection of abnormal living patterns for elderly living alone using support vector data description," *IEEE Transactions on Information Technology in Biomedicine*, vol. 15, no. 3, pp. 438–448, 2011. Article (CrossRef Link)
[40]D. Heckerman, D. Geiger, and D. M. Chickering, "Learning Bayesian networks: The combination of knowledge and statistical data," *Machine Learning*, vol. 20, no. 3, pp. 197–243, 1995. Article (CrossRef Link)
[41]Koller D., Friedman N., *Probabilistic Graphical Model*, MIT Press, Massachusetts, 2009. Article (CrossRef Link)
[42]H. Amirkhani, M. Rahmati, P. J. Lucas, and A. Hommersom, "Exploiting Experts' Knowledge for Structure Learning of Bayesian Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 11, pp. 2154-2170, 2017. Article (CrossRef Link)
[43]H. Amirkhani and M. Rahmati, "Expectation maximization based ordering aggregation for improving the K2 structure learning algorithm," *Intelligent Data Analysis*, vol. 19, pp. 1003–1018, Sep. 2015. Article (CrossRef Link)
[44]P. Spirtes and C. Glymour, "An algorithm for fast recovery of sparse causal graphs," *Social Science Computer Review*, vol. 9, no. 1, pp. 62–72, 1991. Article (CrossRef Link)

[45]A. Ankan and A. Panda, "Mastering Probabilistic Graphical Models Using Python," in *Proc. of THE 14th PYTHON IN SCIENCE CONF (SCIPY 2015)*, 2015. Article (CrossRef Link)
[46]R. Daly, Q. Shen, and S. Aitken, "Learning Bayesian networks: approaches and issues," *The Knowledge Engineering Review*, vol. 26, no. 2, pp. 99–157, 2011. Article (CrossRef Link)
[47]T. Van Kasteren, A. Noulas, G. Englebienne, and B. Kröse, "Accurate activity recognition in a home setting," in *Proc. of the 10th international conference on Ubiquitous computing*, pp. 1–9, 2008. Article (CrossRef Link)
[48]D. J. Cook, "Learning setting-generalized activity models for smart spaces," *IEEE Intelligent Systems*, vol. 27, no. 1, pp. 32–38, 2012. Article (CrossRef Link)
[49]Y. Han, M. Han, S. Lee, A. Sarkar, and Y.-K. Lee, "A framework for supervising lifestyle diseases using long-term activity monitoring," *Sensors*, vol. 12, no. 5, pp. 5363–5379, 2012. Article (CrossRef Link)
[50]K. P. Murphy, "Performance evaluation of binary classifiers," *Technical Report, University of British Columbia, Report*, 2007.

Sasan Saqaeeyan received the B.S. and M.S degrees in Computer Software Engineering from Isfahan University,Iran, in 2007 and Islamic Azad University Science and Research Khuzestan Branch,Iran,in 2009 respectively. He is currently a P.H.D. candidate in Department of Computer from Islamic Azad University Borujerd Branch, Iran. His research interests include data mining.

Hamid Haj Seyyed javadi is corresponding author.He received the B.S.,M.S. and Ph.D degrees in Amirkabir University. He has been working as a full-time faculty member and Associate Professor in the Department of Mathematics and Computer Science of Shahed University. His research interests are ad-hoc network technologies, sensor network technology.

Hossein Amirkhani received the PhD degree in artificial intelligence from the Computer Engineering Department, Amirkabir University of Technology (Tehran Polytechnic), Iran in 2015 for his work on expert-based structure learning of Bayesian networks. He is currently an assistant professor at the Technology and Engineering Department, University of Qom, Iran. His research interests include machine learning, pattern recognition, and data mining.