A Novel Probabilistic Hybrid Model to Detect Anomaly in Smart Homes

Sasan Saqaeeyan[1], Hamid Haj Seyyed Javadi[1, 2, *] and Hossein Amirkhani[1, 3]

Abstract: Anomaly detection in smart homes provides support to enhance the health and safety of people who live alone. Compared to the previous studies done on this topic, less attention has been given to hybrid methods. This paper presents a two-steps hybrid probabilistic anomaly detection model in the smart home. First, it employs various algorithms with different characteristics to detect anomalies from sensory data. Then, it aggregates their results using a Bayesian network. In this Bayesian network, abnormal events are detected through calculating the probability of abnormality given anomaly detection results of base methods. Experimental evaluation of a real dataset indicates the effectiveness of the proposed method by reducing false positives and increasing true positives.

Keywords: Smart homes, sensory data, anomaly detection, Bayesian networks, ensemble method.

1 Introduction

Today, the elderly population are increasing quickly. They tend to live independently more often. However, this trend has negative consequences, and there are significant concerns about health and safety of this group of people [Eyal, Hurst, Norheim et al. (2013); Häfner, Baumert, Emeny et al. (2012); Risteska Stojkoska, Trivodaliev and Davcev (2017)]. As such, a system to detect dangerous incidents and take appropriate timely action is necessary to protect the residents' health. Remote healthcare monitoring systems have provided a viable option to solve this problem [Caroux, Consel, Dupuy et al. (2018); Gomes, Muniz, da Silva e Silva et al. (2017)]. By rapid advances in technologies related to sensors and machine learning algorithms, a number of smart home's commercial applications have been building and have been using in common daily life [Dahmen, Cook, Wang et al. (2017); Gomez, Chessa, Fleury et al. (2019); Suryadevara and Mukhopadhyay (2015)]. The applications are used for automating works at home, optimizing energy consumption, activity recognition, and dangerous event increasing security such as health monitoring [Dahmen, Cook, Wang et al. (2017); Stojkoska and Trivodaliev (2017)].

[1] Department of Computer Engineering, Borujerd Branch, Islamic Azad University, Borujerd, Iran.

[2] Department of Mathematics and Computer Science, Shahed University, Tehran, Iran.

[3] Computer Engineering and Information Technology Department, University of Qom, Qom, Iran.

[*] Corresponding Author: Hamid Haj Seyyed Javadi. Email: h.s.javadi@shahed.ac.ir.

CMES. doi: 10.32604/cmes.2019.07848

Smart homes system consists of hardware and software [Amiribesheli, Benmansour and Bouchachia (2015); Chan, Estève, Escriba et al. (2008); Ni, García Hernando and Pau de la Cruz (2016)]. The hardware includes sensors, communication network, server, and a caregiver interface device. Sensors are installed in different places at home, objects, and resident's body. In smart homes, the events are sensed using the embedded sensors which are attached to the residents' body, objects, and places at home. The collected data is then processed and analyzed using machine learning algorithms on the server. In situations that could be dangerous for residents, they send an alarm to caregiver to perform appropriate reactions. Caregivers can control the various parts of the smart home via interface software [Bakar, Ghayvat, Hasanm et al. (2016)]. Fig. 1 shows the smart home components and its data cycle.

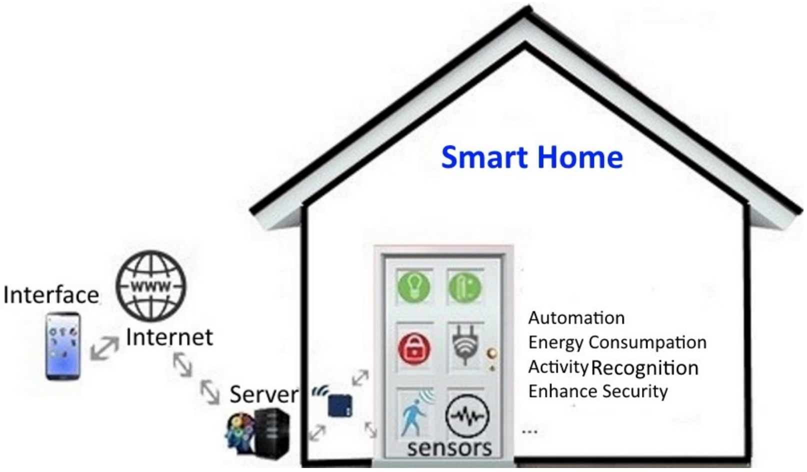

Figure 1: Smart home's data cycle

In recent years, many research studies are conducted on smart homes to detect residents [Lesani, Ghazvini and Amirkhani (2019)], recognize residents' activities [Alemdar, van Kasteren and Ersoy (2017); Lara and Labrador (2013); Ni, García Hernando and Pau de la Cruz (2016); Yang, Jafari, Sastry et al. (2009)], predict activities and events [Nazerfard and Cook (2015)], and detect abnormal activities [Bakar (2016); Song, Wen, Lin et al. (2013); Zhu, Sheng and Liu (2015)]. However, these homes are immature in some aspects, such as the anomaly detection, security of software system, self-healing hardware failure, and the cost of installation and maintenance [Dahmen, Cook, Wang et al. (2017); Suryadevara and Mukhopadhyay (2015); Theoharidou, Tsalis and Gritzalis (2017)]. This paper presents a method for detecting anomalies in the pattern of residents' life from sensory data in a smart home.

Anomaly or outlier is defined as an object or data point that differs from other objects or the rest of data points [Chandola, Banerjee and Kumar (2009); Steen, Frenken, Eichelberg et al. (2013); Tonejc, Güttes, Kobekova et al. (2016)]. Accordingly, abnormal events or activities in the smart home are patterns in sensory data that do not conform subjects' past behavior patterns. Because abnormal events such as falling on the floor, and heart attack occur very rarely in the real life, it is an imbalanced problem. Moreover, we encounter a one-class problem, and our purpose is to detect novelties in the sensory data. The novelty is new or unknown objects which differ from historical data. Because of the similar solutions for anomaly detection and novelty detection, these terms are used interchangeably in the literature [Pimentel, Clifton, Clifton et al. (2014)]. Novelty detection methods are used to detect unknown data points when the number of abnormal data is not sufficient for learning their pattern [Ding, Li, Belatreche et al. (2014); Pimentel, Clifton, Clifton et al. (2014)].

Various studies have been performed to detect one or more aspects of the anomaly (time, duration, location, sequence, etc...) through the use of different algorithms in smart homes [Bakar (2016); Dahmen, Cook, Wang et al. (2017)]. Each of these researches has its advantages and disadvantages, but they all suffer from a lack of an appropriate model to detect anomaly in different conditions and aspects. To alleviate this problem, we present a two-step ensemble method to aggregate the results of different methods. In the first step, it detects anomalies from sensory data using different novelty detection methods. Then, it builds a directed probabilistic graphical model (Bayesian network) which is used to calculate the probability of the current event given the results of other anomaly detection methods. If the probability is lower than a certain threshold, the model considers this event as an anomalous one.

The main contributions of this work are as follows:

- Presenting a probabilistic hybrid method based on Bayesian networks.
- Determining conditional independence relationships between different algorithms using the Bayesian network's d-separation algorithm.

The rest of the paper is organized as follows: Section 2 reviews the related work on novelty detection in smart homes. Section 3 briefly describes the Bayesian network and conditional independence. The proposed method is presented in Section 4 and it experimentally evaluated and discussed in Section 5. Finally, Section 6 concludes our work.

2 Related work

Various methods have been used for anomaly detection in smart homes. These approaches can be categorized according to the type of anomaly, type of used sensor, and level of data analysis. Generally, there is three type of anomaly: point anomaly, contextual anomaly, and collective anomaly [Han, Pei and Kamber (2011); Anderson, Ros, Keller et al. (2012); Hoque, Dickerson, Preum et al. (2015)]. Furthermore, anomaly detection methods in smart homes are categorized into three classes based on sensor type: visual-based sensors, wearable sensors, and distributed sensors [Zhu, Sheng and Liu (2015); Lara and Labrador (2013); Yin, Yang and Pan (2008)]. Also, they divided into two class depending on the level of data analysis [Bakar, Ghayvat, Hasanm et al. (2016)]: detection of anomaly in activities and detection of anomalies in sensory data. In this paper, anomaly detection methods in smart homes are categorized into five categories according to the used algorithm [Pimentel, Clifton, Clifton et al. (2014)], as is reviewed in the following subsections.

2.1 Statistical methods

Statistical methods detect an object or data as an anomaly if the object is in the low-density areas of the training set. These areas have a high probability of containing abnormal objects.

These methods are categorized into parametric and nonparametric methods. The parametric methods assume a parametric distribution to generate the normal data. They define a probability function f (x, ϴ) to calculate the probability of object x with the parameters ϴ. The nonparametric methods do not assume a hypothesis in advance but try to determine the distribution from the input data [Ding, Li, Belatreche et al. (2014); Pimentel, Clifton, Clifton et al. (2014)].

Nonparametric statistical methods: Histogram is a nonparametric statistical method. Song et al. [Song, Wen, Lin et al. (2013)] calculated the number of times that an activity is repeated in a place and used that to detect anomalous behaviors according to the daily histogram changes. However, methods based on the histogram do not consider dependency between the features.

Parametric statistical methods: Three main parametric statistical methods used for anomaly detection are Dynamic Bayesian Network, Gaussian Mixture Model (GMM), and Hidden Markov Model (HMM).

Zhu et al. [Zhu, Sheng and Liu (2015)] use enhanced first-level Dynamic Bayesian network and added time as a new node. They proposed a coherent anomaly detection model to detect different types of anomalies in four contexts: spacing, timing, sequence, and duration of activities. They proposed to first recognize activities and then detect abnormal activities. This may propagate the errors from the activity recognition phase to anomaly detection.

Cardinaux et al. [Cardinaux, Brownsell, Hawley et al. (2008)] trained a GMM using normal data. They used rule-based algorithms to recognize activities and defined a set of characteristics consist of the start time, duration, weekday, and activity level to detect anomalies. GMM was trained for each type of pre-defined activities. Activities which get a probability lower than a threshold are considered as an anomaly. The GMM can consider dependencies between features for modelling, but it is not appropriate when the work comes with high dimensions or features. Rashidi et al. [Rashidi, Cook, Holder et al. (2011)] proposed to detect and track frequent resident's lifestyle activities. They defined irregular activities as an abnormality. In the first phase, a varied-order sequence miner discovers frequent patterns of sensor events. These patterns are grouped into clusters of frequent activities. In the second phase, a voting Multi-HMM system tracks frequent activities. Sensor events represent observable states and activity labels are hidden states. Ghayvat et al. [Ghayvat, Mukhopadhyay, Shenjie, et al. (2018)] proposed to perform real-time anomaly detection. They created an anomaly prediction model based on the active periods and inactive periods of objects. They defined two parameters for assessing the health of residents. The first parameter indicates that the resident does not use home's objects temporary or ever, and second parameter indicates if the resident uses objects dynamically or temporary. They used a time series to define trend of activities. If the current activity is outside the range of duration of routine activity, it is considered as an anomaly.

Generally, the small size of training dataset in statistical methods decreases performance [Ding, Li, Belatreche et al. (2014); Han, Pei and Kamber (2011)].

2.2 Distance-based methods

Distance-based methods assume an object as an anomaly if the proximity of the object with its neighbours has a considerably different from the proximity of other objects with their neighbours in the same data set [Shams Shirazi (2017)].

Liao et al. [Liao, Kong and Wang et al. (2017)] used the local outlier factor (LOF) model to detect anomalies and defined three features including start time, times, and duration. They calculated behavior anomaly degree using k-nearest neighbours, the density of the feature vector and reachability distance between two feature vector. Also, they performed a deeper analysis of anomalies. They built a visual system. It visualized residents' daily activities from different views such as reasons of the anomaly, activities which have more impact on the anomaly, date of the anomaly. Parvin et al. [Parvin, Chessa, Manca et al. (2018)] presented an architecture that has two main parts. The first part analyzes and models resident's behaviors, and the second part performs real-time analysis, to identify the deviations, and propose appropriate activities to prevent anomalies based on the degree of anomaly. Their approach detected anomalies in the sequence of activities at the end of the day and detected online anomalies in the substring of activities throughout the day. Online detection of anomaly and determine the degree of anomaly are advantages of their method. However, they used pre-defined activities.

Distance-based methods require definite appropriate distance measure for the given data. Thus, when there is data with high-dimensional, they cannot accurately discriminate between normal and abnormal data points [Ding, Li, Belatreche et al. (2014); Shams Shirazi (2017)].

2.3 Domain-based methods

These methods draw a boundary around the normal data points out of this boundary are considered as anomaly [Pimentel, Clifton, Clifton et al. (2014)].

Shin et al. [Shin, Lee and Park (2011)] defined three features: activity level to model an individual's behavioral pattern: motion level, and non-response interval. Activity level was used to detect abnormal physical conditions such as weaknesses. Motion level was related to the resident's general health and detected diseases such as altered mental status. Non-response interval was for unresponsive statues such as when a person is found dead. They combined these three features and used Support Vector Data Description (SVDD) to detect abnormal behavioral situations. Yin et al. [Yin, Yang and Pan (2008)] focused on abnormal activities to reduce the false positive rate. They applied a One-Class Support Vector Machine (OCSVM) to detect abnormal activities. Then, they used a kernel nonlinear regression (KNLR) to filter the suspicious activities for more investigation. This study does not focus on the false negative rate. Domain-based methods require to choose an appropriate scaling method for their features. Moreover, because of locating the boundary just using the training data, there are outliers in training data that can influence the model [Ding, Li, Belatreche et al. (2014); Pimentel, Clifton, Clifton et al. (2014)].

2.4 Reconstruction-based method

These methods build a regression model to compare the target (reconstruction data) with the actual observed data. An object is detected as an anomaly if the reconstruction error is l significant [Pimentel, Clifton, Clifton et al. (2014)]. Neural networks are a type of used reconstruction-based methods. Novák et al. [Novák, Biňas and Jakab (2012)] trained self-organizing map (SOM) based on the normal data and clustered activities. They defined a 2-dimensional data input. The first dimension represented the time of entry to a location and the second dimension was the duration of staying there. Activities that differed from the cluster group were detected as anomalies. They used the first-order Markov model to calculate the probability of transition between two activities. Reconstruction-based methods depend on a number of hyper-parameters to build the model structure [Ding, Li, Belatreche et al. (2014); Pimentel, Clifton, Clifton et al. (2014)].

2.5 Hybrid methods

A combination of algorithms can be used to alleviate the shortcoming of different methods. They are known as ensemble or hybrid methods. The results of based methods are combined in a final module or the results of one part is sent to another part as input.

Forkan et al. [Forkan, Khalil, Tari et al. (2015)] used different algorithms to detect anomalies for various contexts of individual life. An HMM detected changes in the location and sequence of activities. Gaussian distribution detected changes in routine behaviors. Also, they analyzed vital signs using statistical methods. Then, a fuzzy model fused their outputs to make the final decision. This study defined different levels for anomalies to perform proportional reactions for each level. Their study reduced false alarms rate. Ordóñez et al. [Ordóñez, Toledo and Sanchis (2015)] recognized the behavior patterns of occupants using 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, multinomial, 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 anomalies.

The aforementioned methods have their advantages and disadvantages. However, to the best of our knowledge, no attempt has been made to obtain the most appropriate ensemble models. The present paper focuses on these cases and aims to determine the structure of the Bayesian network to fuse the results of other anomaly detection methods from different categories.

Fig. 2 shows the classification of smart homes' applications and components. As is clear in this chart, anomaly detection is a subgroup of the health of residents.

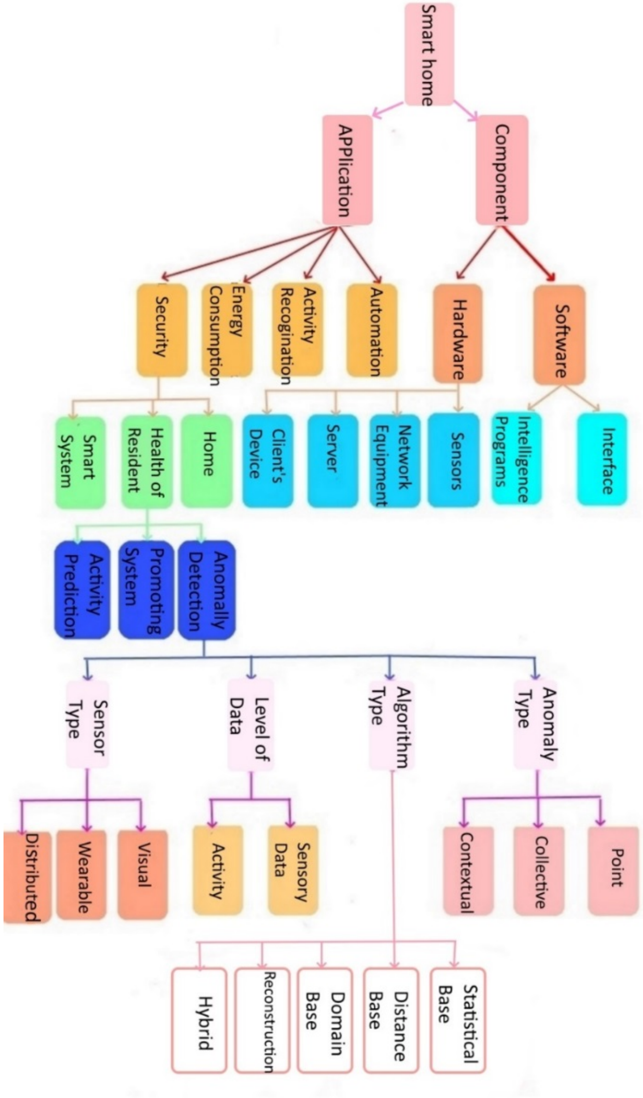

Figure 2: Smart homes' applications and components

3 Bayesian networks

Bayesian network is a probabilistic directed acyclic graphical model. Its nodes correspond to random variables and its edges represent statistical conditional dependency relationships between nodes. These networks calculate the joint probability of distribution of *n* variables using Eq. (1) [Heckerman, Geiger and Chickering (1995); Koller and Friedman (2009)]:

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

where X_i indicates a random variable and $Pa(X_i)$ is its parents.

3.1 Building and using a Bayesian network

The process of building and using a Bayesian network with complete data consists of three steps of structure learning, parameter learning, and inference [Heckerman, Geiger and Chickering (1995); Koller and Friedman (2009)]. Network structure represents relationships between nodes with each other. An unknown structure can be learned using two major approaches: score-based and constraint-based methods. In addition to the training data, experts' knowledge can be exploited to obtain a more accurate network structure [Amirkhani, Rahmati, Lucas et al. (2016)]. Network parameters are the conditional probability distribution of random variables given their parents. Parameter learning estimates conditional probability tables (CPTs) based on the training data and network structure [Heckerman, Geiger and Chickering (1995); Koller and Friedman (2009)]. After learning the structure and parameter, the model can be used to answer probabilistic queries about random variables. This step is called inference which can be exact or approximate [Heckerman, Geiger and Chickering (1995); Koller and Friedman (2009)].

3.2 Conditional independence relationships

The Bayesian network structure can be investigated to extract the conditional independence relationships. It indicates whether nodes X are conditionally independent of Y, given nodes Z, denoted by (X⊥Y|Z) [Cooper and Herskovits (1992)]. A node can have a relationship with another node either directly or indirectly.

If they are directly connected, they are correlated regardless of any other variables as evidence. Fig. 3 shows all types of indirect connections between X and Y via Z.

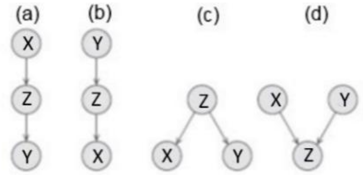

Figure 3: Relations between node X and Y via Z

A path between two nodes (or two sets of nodes) without considering the directions is called a trail. A trail between nodes X to Y via Z is called active if influence can flow from X to Y via Z; otherwise, it is called blocked. In Fig. 3(a) (indirect causal effect), Fig. 3(b) (indirect evidential effect) and Fig. 3(c) (common cause), node X can influence node Y if and only if Z is unobserved and is denoted by X⊥Y|Z. It means the trail between X and Y via Z is active if and only if Z is unobserved. In Fig. 3(d) (common effect or v-structure), influence can flow on the trail between X and Y via Z if Z is observed.

X and Y are called d-separated given Z if there is no active trail between any node x∈X and y∈Y, given Z (it is denoted by d-sep(X,Y|Z)). The concept of d-separation can be used to determine conditional independence relationships between different nodes of a Bayesian network [Cooper and Herskovits (1992)].

4 The proposed method

This section explains the details of the proposed hybrid method based on the Bayesian network to detect anomalies from collected sensory data in smart homes. The proposed model has two phases for detecting anomalies. In the first step of our method, it detects anomalies using various base methods with different types explained in Section 2. Subsequently, all methods are employed on the validation set and (obtained by cross-validation) and build a new dataset containing the prediction of these methods and actual labels. This set is used for training the Bayesian network in the next phase. There is one node for each base anomaly detection method in this Bayesian network. Finally, learned Bayesian network is used to calculate the probability of abnormality for the test samples given the prediction of the base methods. A sample is detected as an anomaly if its probability of abnormality is greater than its normal of being normal. As is depicted in Fig 4, the proposed model works in five steps: pre-processing, base anomaly detection, building the Bayesian network, final anomaly detection and model evaluation.

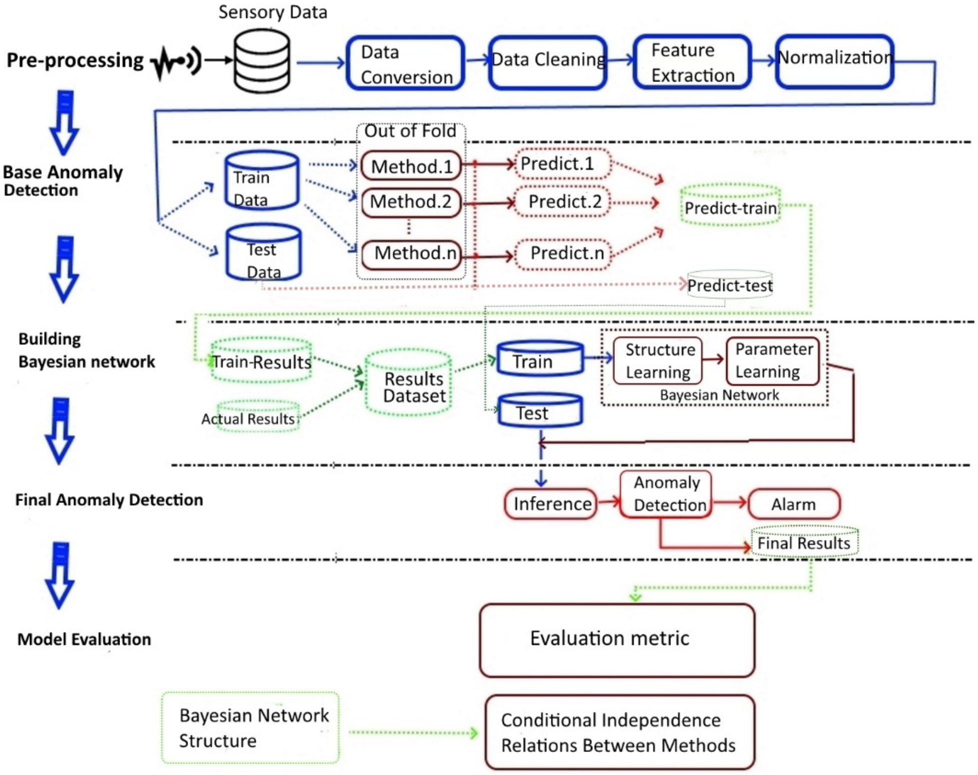

Figure 4: The proposed architecture for anomaly detection in smart homes

4.1 Pre-processing

This phase consists of data cleaning, data conversion, features extraction, data normalization, and dataset splitting. Raw sensory data are collected in a dataset. Each record of the dataset includes three features: time, date and the on/off states of different sensors. Time attributes are discretized based on $\lfloor a/b \rfloor + 1$ where a is the minute part of the time when a sensor's state changes, b is an integer in {15, 30}, and $\lfloor . \rfloor$ shows the floor function. For example, if time is 10:21, a is 21. For b=15, time will map to 103. Missing data and unused features are eliminated, for example, some records have their time of switching on, but their time of switching "off" missed. Table 1 shows the defined features based on the prepared data. The prepared data is then split up into two subsets: train and test sets.

Table 1: Features used in the proposed system

Features	Description
Sid 1	Identification number of the current activated sensor
Time1 on	The switching on time of the current activated sensor
Time1 off	The switching off time of the current activated sensor
Duration	The elapsed time between switching on and off
Sid 2	Identification number for the previous activated sensor
Time2 on	The switching on time of the previous activated sensor
Time2 off	The switching off time of the previous activated sensor

4.2 Base anomaly detection methods

In this phase, we choose multiple novelty detection algorithms from different categories including statistics, distance based, domain based, reconstruction, and ensemble methods. Cross-validation strategy is employed to split the training data to training and validation sets. Various models are learned using the training subset. Then, the trained models are employed on the validation subset to form the training dataset for the next step. Tab. 2 illustrates the selected base anomaly detection methods. They are chosen according to the defined categories in Section 2 since the diversity of the methods is an important factor for the success of ensemble methods.

Table 2: The selected base anomaly detection methods

Method	Description	Domain of novelty detection
Principal Component Analysis (PCA)	Calculate sum of weighted projected distances to the eigenvector hyper plane	Reconstruction
One-Class Support Vector Machine (OCSVM)	Kind of SVM that learns a boundary for the normal class, and any data outside this boundary is detected as an anomaly.	Domain-based
Local Outlier Factor (LOF)	Find k nearest-neighbours, calculates local reachability density (LRD), compare the LRD of a record with the LRDs of its k neighbours	Distance-based
k Nearest Neighbors (kNN)	Calculates distance to the kth nearest neighbour	Distance-based
Clustering-Based Local Outlier Factor (CBLOF)	Determine dense areas by clustering and calculate a density for each cluster	Distance-based

Minimum Covariance Determinant (MCD)	Calculates Mahalanobis distances	Distance-based
Histogram-based Outlier Score (HBOS)	Builds a histogram for each feature of dataset and assumes independence between features	Statistical-based
Naïve Bayes	All the features are independent given the class variable.	Statistical-based
Isolation Forest	An ensemble method which creates a bunch of decision trees	ensemble
Feature Bagging	An ensemble method which selects subsets of features randomly for training model.	ensemble

4.3 Building the Bayesian network

This phase consists of four main parts: composing a results dataset, split up results dataset, structure learning, and parameter learning.

First, prediction results of train data for anomalies detection with the actual results, which are correct classification results of data, to build new results dataset. Each column of the dataset includes prediction results of one model, and the last column is actual results. Then, random variables are defined corresponding to each column of result dataset, a Bayesian network is trained based on the predicted training data. Finally, predication of test data and trained model will be sent to the next phase for the second step of anomaly detection. Tab. 3 shows samples of results dataset. If value of cell$_{m,n}$ is 1, it represents that method$_n$ has detected record$_m$ of the first dataset as an anomaly.

Table 3: Results dataset includes anomaly detection results and actual results

Method $_1$	Method$_2$	...	Method$_n$	Actual results
0	1	…	1	1
1	0	…	0	1
0	0	…	1	0
1	1	…	1	1

4.4 Final anomaly detection

The trained model is used for final anomaly detection in three steps: inference, detect anomaly, and alarm to a caregiver.

• Inference: Each record of the test dataset is given to the trained Bayesian network and it calculates p (r | evidence), where r is the final label of the current record and evidence constitutes the predictions of the base anomaly detection methods.

• Anomaly detection and alarm: For the calculated probabilities, if p (normal | evidence)<p (anomaly | evidence), then the record is detected as an anomaly, and an alarm notification is generated in the next step.

4.5 Model evaluation

Model evaluation is performed to study the performance of the proposed model. For this purpose, F-score and area under the curve (AUC), explained in Section 5.1, are calculated. Then, the d-separation algorithm is used to determine the conditional independence relations between nodes of the Bayesian network that demonstrate the relations between the used methods in the trained Bayesian network.

4.6 Pseudo-code of the proposed method

A pseudo-code of the proposed method is presented in Algorithm 1. It receives the training and test data. In each step of the outer loop, one model is selected. The inner loop is used for training and testing models. Model_Set is an array containing the base anomaly detection methods. The cross-validation process is repeated k times and ThisRoundDatasets indicates the current iteration. After building the Bayesian network, the trained model performs inference for each record in the test data to estimate its probability of being anomalous.

```
Algorithm 1 Proposed Algorithm
         Input: Training Dataset ($D_{train}$),
                   Testing Dataset ($D_{test}$)
         Output: $Evalution_{metric}$,
                   Conditional Independence relationships
  (Model_Set is an array containing base anomaly detection methods,
  and CV refers to cross validation )

         BN_dataset = an empty dataset with models in Model_Set as features
  FOR each subModel in Model_Set do
           FOR each $CV_{train}$, $CV_{test}$ in CV[ThisRoundDatasets] do
         tempModel = Train a subModel on $CV_{train}$
         tempPred = Apply the tempModel on $CV_{test}$
                  Add tempPred to BN_dataset[subModel]
  BN_structure = HillClimbSearch(BN_dataset,scoring_method=K2Score)
         BN_parameters = ParameterEstimation(BN_dataset,BN_structure)
         Model = BayesianNetwork(BN_strucutre, BN_parameters)
  For each record in $D_{test}$ do
          Probability = Inference(Model, record)
          IF Probability < 0.5 THEN
         Mark this record as an anomaly
```

5 Experimental evaluation

In this section, we evaluate the proposed method compared to alternative approaches.

5.1 Experimental setup

The Kasteren dataset [van Kasteren, Englebienne and Kröse (2010)] is used for the evaluation of our work. This dataset consists of real-data records gathered from the daily life of a single-resident smart home apartment for 28 days. They used RFM DM 1810 kits to build a wireless network of nodes. Binary distributed sensors including Contact Switches sensors and pressure sensors are installed in different parts of the home, such as entrance door, beds, and kitchen, including freezer, microwave, and cabinets. Fig. 5 shows the smart home with the sensors marked by red signs. Records of the raw dataset have three fields: ID (sensor id), Start time (when the sensor switches on), and End time (when the sensor switches off). The sensory data gathered from daily life are considered as normal data. We manually generated a number of abnormal samples for the Kasteren dataset based on unusual behavior and statistical information of the dataset. They differ from the rest of the dataset in the time, location, sequence, or interval time between the switching on/off the sensors. Other abnormal samples were generated by the algorithm used in Novák et al. [Novák, Biňas and Jakab (2012)].

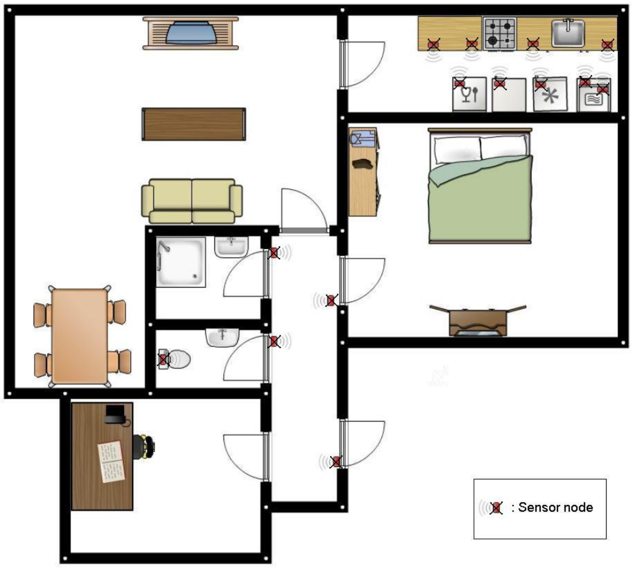

Figure 5: The location of the sensors in the smart home used in the experiments [van Kasteren, Englebienne, and Kröse (2010)]

Evaluation criteria

We calculated the confusion matrix that consists of the following four cells:

- TP: Total number of abnormal records that are correctly detected.
- FP: Total number of normal records that are incorrectly detected as abnormal.
- TN: Total number of normal records that are correctly detected as normal.
- FN: Total number of abnormal records that are incorrectly detected as normal.

The used classification evaluation criteria are as follow [Han, Pei and Kamber (2011); Murphy (2007)]:

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

5.2 Experimental results

The proposed system is implemented in python. For the first anomaly detection step, we used Python Outlier Detection (PyOD) which is an effective anomaly detection toolkit [Zhao, Nasrullah and Li (2019)]. Pgmpy is used to train the Bayesian network for final anomaly detection. It is an open-source python library for building probabilistic graphical models [Ankan and Panda (2015)]. Two other known ensemble approaches are compared with the proposed model: simple stacking and bagging. Simple stacking is similar to our proposed method but its final classifier is a simple logistic regression instead of Bayesian network. Bagging is a technique that bootstraps the training data into different sets, train a classifier on each set, and calculates the final result by majority voting [Han, Pei and Kamber (2011)]. Fig. 6 shows the structure of the trained Bayesian network (Y_true indicates final result). The K2 algorithm is used for structure–learning and Bayesian estimation for parameter learning [Amirkhani and Rahmati (2015)].

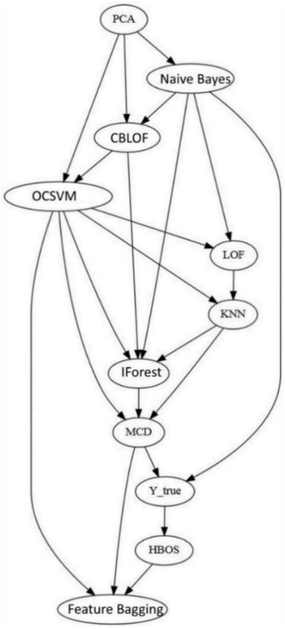

Figure 6: Trained Bayesian network based on the results of base methods

Evaluation

The evaluation results of the proposed method and base methods are presented in Tab. 4. It is clear that the proposed hybrid method obtains better F-score compared to the base anomaly detection methods. Especially, its precision is considerably higher than the base methods.

Table 4: Evaluation results

Method	Precision	Recall	F1
Proposed method	0.90	0.95	0.92
PCA	0.50	0.95	0.66
OCSVM	0.57	0.87	0.69
LOF	0.63	0.87	0.73
KNN	0.71	0.82	0.76
CBLOF	0.49	0.77	0.60
MCD	0.92	0.80	0.86
HBOS	1	0.67	0.80
Naïve Bayes	0.45	0.83	0.59
Feature bagging	0.56	0.67	0.61
Isolation Forest	0.65	0.82	0.73

Tab. 5 shows the experimental results comparing the proposed method with the competing ensemble approaches. Fig. 7 shows the ROC curves of the proposed model and two competing ensemble methods. The results presented in Tab. 6 suggests that the proposed hybrid model achieves a significantly higher AUC.

Table 5: Comparison of different ensemble methods

Method	Precision	Recall	F1
Proposed Method	0.90	0.95	0.92
Simple Stacking	0.85	0.90	0.87
Bagging	0.79	0.82	0.80

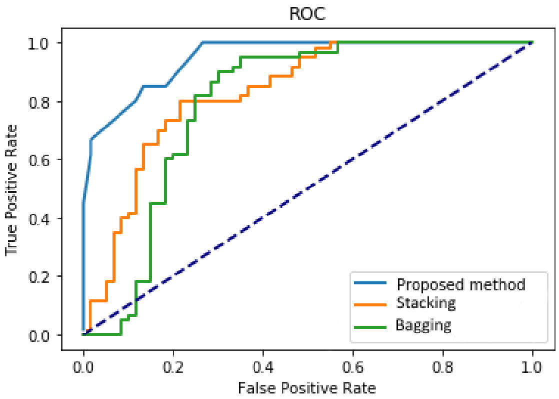

Figure 7: The ROC curves of the ensemble models under investigation

Table 6: AUC of ensemble methods

	Proposed Model	Simple Stacking	Bagging
AUC	0.93	0.84	0.81

Finally, Tab. 7 shows the calculated confusion matrix for the proposed method. Clearly, it is successful in detecting the anomalies while keeping the normal records correctly.

Table 7: The confusion matrix of the proposed method

	Predicted class		
		Positive	Negative
Actual class	Positive	228	12
	Negative	25	95

5.3 Conditional independence relationships

In this section, we determine conditional independence relationships between different nodes of Fig. 6 using the d-separation algorithm. Tab. 8 shows these relations. It indicates conditional independencies between different used anomaly detection methods. For example, the first row indicates that given the result of MCD and Naïve Bayes methods, all the methods in {KNN, CBLOF, LOF, IForest, OCSVM, PCA} are independent of Y_true.

Table 8: Conditional independence relationships between different algorithms

Conditional independencies
Y_true _|_ KNN, CBLOF, LOF, IForest, OCSVM, PCA | MCD, NaiveBayes
(KNN _|_ CBLOF, PCA, NaiveBayes | LOF, OCSVM)
(LOF _|_ CBLOF, PCA | NaiveBayes, OCSVM)
(MCD _|_ PCA, LOF, CBLOF, NaiveBayes | KNN, IForest, OCSVM)
(IForest _|_ PCA, LOF | KNN, CBLOF, NaiveBayes, OCSVM)
(FeatureBagging _|_ KNN, Y_true, PCA, LOF, IForest, CBLOF, NaiveBayes | MCD, HBOS, OCSVM)
(HBOS _|_ KNN, PCA, LOF, IForest, MCD, NaiveBayes, OCSVM, CBLOF | Y_true)
(OCSVM _|_ NaiveBayes | CBLOF, PCA)

5.4 Discussion

The proposed hybrid model resembles the stacking approach to ensemble learning, but it uses the Bayesian network to aggregate the results of base classifiers in the second layer. According to the results shown in Tab. 4 and Tab. 5, the idea of using a new probabilistic hybrid method based on Bayesian networks to detect anomalies in smart homes is promising. This method outperforms the competing approaches. Also, the proposed method is suitable for different conditions because methods are employed from different domains of novelty detection.

Evaluation results of ROC curve (Fig. 7) shows that the Bayesian network structure has made proper relation between different methods (nodes) for final anomaly detection in a way to decrease false alarm rate and increase the true positive rate.

The anomalies in the proposed method are detected using the sensory data. Hence, proposed method in comparison to abnormal activities detection methods [Ghayvat, Mukhopadhyay, Shenjie et al. (2018); Parvin, Chessa, Manca et al. (2018)], which have used pre-defined activities, can detect different anomaly types in the life pattern such as: time, sequence, and duration. For example, late waking-up can be detected using the unusual start and end times of the events.

Determining relations between used methods in the proposed model (Tab. 8) is an effective achievement that indicates the relationships between different base methods from different domain. In addition, since careful selection of the base methods can result in a better ensemble performance, the proposed approach can be used to choose (nearly) independent methods to be used in the ensemble methods.

The main limitation of this approach in comparison with the methods presented in Forkan et al. [Forkan, Khalil, Tari et al. (2015); Ordóñez, de Toledo and Sanchis (2015)] is that it is unable to detect the anomaly type to perform appropriate reactions (i.e., normal, warning, and alert emergency [Novák, Biňas and Jakab (2012)].

6 Conclusion

This paper proposed a novel multi-phase probabilistic hybrid model for anomaly detection in smart homes which is going to promote the residents' safety and health. The anomalies are detected in two steps. First, we deployed methods with different characteristics to obtain an initial guess of available anomalies from the sensory data. Then, we used a Bayesian network to hybridize their outputs. Experimental results using a real dataset found the proposed method strikingly efficient. The aim of this paper was to detect anomalies in smart homes, but the proposed framework can be applied to other areas as well. Also, in our experiments, some methods were detected to be conditionally independent of other methods. In future studies, we intend to improve the model through the combination of carefully selected methods according to their correlations. Moreover, we will extend the current model by using wearable sensors to investigate more aspects of anomaly in smart homes.

References

Alemdar, H.; van Kasteren, T. L. M.; Ersoy, C. (2017): Active learning with uncertainty sampling for large scale activity recognition in smart homes. *Journal of Ambient Intelligence and Smart Environments*, vol. 9, no. 2, pp. 209-223.

Amiribesheli, M.; Benmansour, A.; Bouchachia, A. (2015): A review of smart homes in healthcare. *Journal of Ambient Intelligence and Humanized Computing*, vol. 6, no. 4, pp. 495-517.

Amirkhani, H.; Rahmati, M. (2015): Expectation maximization based ordering aggregation for improving the K2 structure learning algorithm. *Intelligent Data Analysis*, vol. 19, pp. 1003-1018.

Amirkhani, H.; Rahmati, M.; Lucas, P. J.; Hommersom, A. (2016): Exploiting experts' knowledge for structure learning of bayesian networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 11, pp. 2154-2170.

Anderson, D. T.; Ros, M.; Keller, J. M.; Cuéllar, M. P.; Popescu, M. et al. (2012): Similarity measure for anomaly detection and comparing human behaviors. *International Journal of Intelligent Systems*, vol. 27, no. 8, pp. 733-756.

Ankan, A.; Panda, A. (2015): *Mastering Probabilistic Graphical Models Using Python*. Packt Publishing Ltd., UK.

Bakar, U.; Ghayvat, H.; Hasanm, S.; Mukhopadhyay, S. (2016): Activity and anomaly detection in smart home: a survey. *Next Generation Sensors and Systems*, pp. 191-220.

Cardinaux, F.; Brownsell, S.; Hawley, M.; Bradley, D. (2008): Modelling of behavioural patterns for abnormality detection in the context of lifestyle reassurance. *Iberoamerican Congress on Pattern Recognition*, pp. 243-251.

Caroux, L.; Consel, C.; Dupuy, L.; Sauzéon, H. (2018): Towards context-aware assistive applications for aging in place via real-life-proof activity detection. *Journal of Ambient Intelligence and Smart Environments*, vol. 10, no. 6, pp. 445-459.

Chan, M.; Estève, D.; Escriba, C.; Campo, E. (2008): A review of smart homes-Present state and future challenges. *Computer Methods and Programs in Biomedicine*, vol. 91, no. 1, pp. 55-81.

Chandola, V.; Banerjee, A.; Kumar, V. (2009): Anomaly detection: a survey. *ACM Computing Surveys*, vol. 41, no. 3, pp. 15.

Cooper, G. F.; Herskovits, E. (1992): A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, vol. 9, no. 4, pp. 309-347.

Dahmen, J.; Cook, D. J.; Wang, X.; Honglei, W. (2017): Smart secure homes: a survey of smart home technologies that sense, assess, and respond to security threats. *Journal of Reliable Intelligent Environments*, vol. 3, no. 2, pp. 83-98.

Ding, X.; Li, Y.; Belatreche, A.; Maguire, L. P. (2014): An experimental evaluation of novelty detection methods. *Neurocomputing*, vol. 135, pp. 313-327.

Eyal, N.; Hurst, S. A.; Norheim, O. F.; Wikler, D. (2013): *Inequalities in Health: Concepts, Measures, and Ethics*. Oxford University Press.

Forkan, A. R. M.; Khalil, I.; Tari, Z.; Foufou, S.; Bouras, A. (2015): 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.

Ghayvat, H.; Mukhopadhyay, S.; Shenjie, B.; Chouhan, A.; Chen, W. (2018): Smart home based ambient assisted living: recognition of anomaly in the activity of daily living for an elderly living alone. *IEEE International Instrumentation and Measurement Technology Conference*, pp. 1-5.

Gomes, B.; Muniz, L. C. M.; da Silva e Silva, F. J.; Ríos, L. E. T.; Endler, M. (2017): A comprehensive and scalable middleware for ambient assisted living based on cloud computing and internet of things. *Concurrency and Computation: Practice and Experience*, vol. 29, no. 11, e4043.

Gomez, C.; Chessa, S.; Fleury, A.; Roussos, G.; Preuveneers, D. (2019): Internet of Things for enabling smart environments: a technology-centric perspective. *Journal of Ambient Intelligence and Smart Environments*, vol. 11, no. 1, pp. 23-43.

Häfner, S.; Baumert, J.; Emeny, R.; Lacruz, M.; Bidlingmaier, M. et al. (2012): 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.

Han, J.; Pei, J.; Kamber, M. (2011): *Data Mining: Concepts and Techniques*. Elsevier.

Heckerman, D.; Geiger, D.; Chickering, D. M. (1995): Learning Bayesian networks: the combination of knowledge and statistical data. *Machine learning*, vol. 20, no. 3, pp. 197-243.

Hoque, E.; Dickerson, R. F.; Preum, S. M.; Hanson, M.; Barth, A. et al. (2015): Holmes: A comprehensive anomaly detection system for daily in-home activities. *2015 International Conference on Distributed Computing in Sensor Systems*, pp. 40-51.

van Kasteren, T. L. M.; Englebienne, G.; Kröse, B. J. A. (2010): Transferring knowledge of activity recognition across sensor networks. In: Floréen, P., Krüger, A. and Spasojevic, M. (eds.), *Pervasive Computing, Lecture Notes in Computer Science*, pp. 283-300. Springer Berlin Heidelberg.

Koller, D.; Friedman, N. (2009): *Probabilistic Graphical Models*. Massachusetts, MIT Press.

Lara, O. D.; Labrador, M. A. (2013): A survey on human activity recognition using wearable sensors. *IEEE Communications Surveys and Tutorials*, vol. 15, no. 3, pp. 1192-1209.

Lesani, F. S.; Ghazvini, F. F.; Amirkhani, H. (2019): Smart home resident identification based on behavioral patterns using ambient sensors. *Personal and Ubiquitous Computing*, pp. 1-12.

Liao, Z.; Kong, L.; Wang, X.; Zhao, Y.; Zhou, F. et al. (2017): A visual analytics approach for detecting and understanding anomalous resident behaviors in smart healthcare. *Applied Sciences*, vol. 7, no. 3, pp. 254.

Murphy, K. P. (2007): *Performance Evaluation of Binary Classifiers*. University of British Columbia.

Nazerfard, E.; Cook, D. J. (2015): CRAFFT: an activity prediction model based on bayesian networks. *Journal of Ambient Intelligence and Humanized Computing*, vol. 6, no. 2, pp. 193-205.

Ni, Q.; García Hernando, A. B.; Pau de la Cruz, I. (2016): A context-aware system infrastructure for monitoring activities of daily living in smart home. *Journal of Sensors*, vol. 2016.

Novák, M.; Biňas, M.; Jakab, F. (2012): Unobtrusive anomaly detection in presence of elderly in a smart-home environment. *ELEKTRO*, pp. 341-344.

Ordóñez, F. J.; de Toledo, P.; Sanchis, A. (2015): Sensor-based Bayesian detection of anomalous living patterns in a home setting. *Personal and Ubiquitous Computing*, vol. 19, no. 2, pp. 259-270.

Parvin, P.; Chessa, S.; Manca, M.; Paterno, F. (2018): Real-time anomaly detection in elderly behavior with the support of task models. *Proceedings of the ACM on Human-Computer Interaction*, vol. 2, no. EICS, pp. 15.

Pimentel, M. A.; Clifton, D. A.; Clifton, L.; Tarassenko, L. (2014): A review of novelty detection. *Signal Processing*, vol. 99, pp. 215-249.

Risteska Stojkoska, B.; Trivodaliev, K.; Davcev, D. (2017): Internet of things framework for home care systems. *Wireless Communications and Mobile Computing*, vol. 2017.

Shams Shirazi, S. (2017): *A Novelty Detection Tool Based on Parallel Coordinates Plot*. École Polytechnique de Montréal.

Shin, J. H.; Lee, B.; Park, K. S. (2011): 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.

Song, Y.; Wen, Z.; Lin, C. Y.; Davis, R. (2013): One-class conditional random fields for sequential anomaly detection. *Twenty-Third International Joint Conference on Artificial Intelligence*.

Steen, E. E.; Frenken, T.; Eichelberg, M.; Frenken, M.; Hein, A. (2013): 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.

Stojkoska, B. L. R.; Trivodaliev, K. V. (2017): A review of Internet of Things for smart home: Challenges and solutions. *Journal of Cleaner Production*, vol. 140, pp. 1454-1464.

Suryadevara, N. K.; Mukhopadhyay, S. C. (2015): *Smart Homes: Design, Implementation and Issues*. Springer.

Theoharidou, M.; Tsalis, N.; Gritzalis, D. (2017): Smart Home Solutions: Privacy Issues. *Handbook of Smart Homes, Health Care and Well-Being*, pp. 67-81.

Tonejc, J.; Güttes, S.; Kobekova, A.; Kaur, J. (2016): Machine learning methods for anomaly detection in BACnet networks. *Journal of Universal Computer Science*, vol. 22, no. 9, pp. 1203-1224.

Yang, A. Y.; Jafari, R.; Sastry, S. S.; Bajcsy, R. (2009): Distributed recognition of human actions using wearable motion sensor networks. *Journal of Ambient Intelligence and Smart Environments*, vol. 1, no. 2, pp. 103-115.

Yin, J.; Yang, Q.; Pan, J. J. (2008): Sensor-based abnormal human-activity detection. *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 8, pp. 1082-1090.

Zhao, Y.; Nasrullah, Z.; Li, Z. (2019): PyOD: a python toolbox for scalable outlier detection. *Journal of Machine Learning Research*, vol. 20, no. 96, pp. 1-7.

Zhu, C.; Sheng, W.; Liu, M. (2015): 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.