Cognitive Systems Research 93 (2025) 101389

Contents lists available at ScienceDirect

Cognitive Systems Research

journal homepage: www.elsevier.com/locate/cogsys

Modeling behavioral deviations in ADLs using Inverse Reinforcement Learning

Fateme Akbari [a],*, Kamran Sartipi [b]

[a] DeGroote School of Business, McMaster University, Hamilton, ON, Canada
[b] Department of Computer Science, East Carolina University, Greenville, NC, USA

ARTICLE INFO

Action editor: Serge Thill

Keywords:
Inverse Reinforcement Learning
Anomaly detection
Activities of Daily Living
Higher-order Markov Chain

ABSTRACT

The detection of abnormalities in Activities of Daily Living (ADLs) has garnered significant attention in recent studies, with many employing deep learning techniques. This paper introduces a novel approach to analyzing ADL sequences, aimed at identifying meaningful deviations from an individual's routine behavior. Our method offers several benefits for older adults, including timely care, early detection of health conditions to prevent deterioration, reduced monitoring burden on family members, and enhanced self-sufficiency without disrupting daily activities. We propose an Inverse Reinforcement Learning (IRL)-based method to detect behavioral abnormalities in older adults by analyzing ADL sequences. Our approach models the problem of abnormality detection in behavior sequences as a Markov Chain model. By applying the IRL method, we infer the reward function that motivates individuals to perform ADL from observed behavior trajectories. This inferred reward function is then used to identify potential behavior abnormalities through a threshold-based mechanism, where sequences with rewards below a specified threshold are flagged as potential abnormalities.

1. Introduction

As the world's population ages, there is an increasing need to develop technologies that can support healthy aging and enable older adults to live independently in their homes for as long as possible. One of the challenges of aging is that it often leads to changes in behavior, which can be early indicators of cognitive decline or other health issues. Detecting these changes in behavior can help caregivers and healthcare professionals intervene early, potentially improving health outcomes and quality of life for older adults.

Activities of Daily Living (ADL) data, which includes information about daily activities such as eating, sleeping, and personal hygiene, can be used to track changes in behavior over time. However, detecting behavior changes from ADL data can be challenging due to the complexity and variability of human behavior. The monitoring of ADL has been promoted by advancements in sensor technologies. Decreased ADL performance has been found to be associated with the progression of chronic diseases, including cognitive impairment, in older adults (Stineman et al., 2011). A study comparing two groups of older adults found that the activity maps of dementia patients displayed disorganized behavior patterns, and there was a notable difference in heterogeneity between the healthy group and the group with the disease (Urwyler et al., 2017). Therefore, the study of life patterns in older persons can be used to quantify changes relevant to ADL in the course of diseases. Although there is plenty of research on ADL recognition and ADL impairment detection, studying irregularities in the pattern of daily life has not been studied enough. The existing research on behavior anomaly detection in older adults has primarily focused on point anomalies, neglecting the potential of utilizing temporal features to their fullest extent. While these studies have successfully identified anomalies where individual data points deviate from the norm, they have overlooked collective anomalies that can only be detected by analyzing the sequential nature of the data. Moreover, some investigations have been limited to identifying abnormalities within specific activity classes, failing to account for higher-level analysis of activities. Thus, it is crucial to consider appropriate behavior granularity in developing effective anomaly detection methods. Additionally, it is desirable for the method to offer a generalizable solution that can be adjusted for different target users within a reasonable timeframe, enabling it to leverage pre-learned models and accelerate the learning process.

In this work, we propose an IRL-based model for behavior abnormality detection in older adults, where the reward function is inferred from the observed behavior of an expert, which is called trajectories. To be specific, observed sequences of ADL performed by an individual are fed into the model as trajectories to learn the reward function.

The proposed model takes advantage of Inverse Reinforcement Learning for training the agent to learn the behavior of the older

* Corresponding author.
E-mail address: akbarif@mcmaster.ca (F. Akbari).

https://doi.org/10.1016/j.cogsys.2025.101389
Received 5 July 2024; Received in revised form 8 July 2025; Accepted 19 August 2025
Available online 30 August 2025

adult through a semi-supervised task. Inverse Reinforcement Learning is a machine learning technique that has shown promise in modeling human behavior and inferring underlying motivations (Lin & Cook, 2020). Inverse reinforcement learning is particularly beneficial in cases where defining the reward function is challenging due to its complexity. Additionally, the reward function has been demonstrated to exhibit greater transferability compared to the policy function (Russell, 1998), leading to the development of more generalizable models. Unlike supervised methods that require labeled data for training, IRL-based models learn from trajectories that are observations of an expert performing the task. The proposed model learns a reward function that captures the underlying motivations behind the observed ADL data and uses this function to detect changes in behavior. The model can adapt to changes in behavior over time, making it well-suited for detecting early indicators of cognitive decline or other health issues.

The main contributions of this research are as follows: (1) A novel representation of the abnormality detection in ADL sequences as a higher-order Markov Chain model. (2) A semi-supervised IRL-based model for detecting behavior changes in older adults from sequences of ADL data. (3) An evaluation of the proposed model on a real-world dataset of ADL data from older adults.

The rest of the paper is structured as follows: Section 2 reviews related work in the field. Section 3 provides background on the methods we used. In Section 4, we introduce our proposed approach, followed by the presentation of our results in Section 5. Finally, we conclude the paper in Section 6.

2. Related works

Abnormal behavior can be defined as "actions that are unexpected and often evaluated negatively because they differ from typical or usual behavior" (Durand & Barlow, 2003). Because the concept of an anomaly is difficult to define precisely and is closely tied to patient behaviors and the types and course of pathologies, artificial intelligence, and more specifically machine learning techniques, have been used to learn to recognize those anomalies.

Scholars have used machine learning methods extensively to analyze ADL with the goal of providing on-time care and predicting older adults' health conditions. Many studies benefit from the availability of datasets for daily activities, including the use of machine learning methods for predicting/detecting anomalous behavior (Arifoglu & Bouchachia, 2019a; Freitas et al., 2022; Lotfi, Langensiepen, Mahmoud, & Akhlaghinia, 2012; Riboni, Bettini, Civitarese, Janjua, & Helaoui, 2015; Suryadevara, Mukhopadhyay, Wang, & Rayudu, 2013).

Fahad and Tahir (2021) propose a method for detecting behavior anomalies by taking into account two types of abnormality: missing or extra sub-events in an activity and unusual durations of the activity. They trained an H2O model to classify events using labeled activities (normal, anomaly). The main problem with such supervised models is that they must be trained using labeled data, which is time-consuming and difficult to generate.

Casagrande, Tørresen, and Zouganeli (2018) have used recurrent neural networks to forecast the future values of the activities for each sensor. When abnormal behavior is anticipated in the near future, the caregiver is informed using the projected values. Investigations into data gathering, classification, and prediction were conducted in actual homes with dementia-affected elderly residents.

In assisted living settings, temporal characteristics of ADL are taken into consideration to forecast the next activity. Nazerfard (2018) presents an association rule mining module that identifies associations among ADL that are grouped according to the start time and duration of the related ADL. The sequence of the activities is also taken into account.

Karakostas, Briassouli, Avgerinakis, Kompatsiaris, and Tsolaki (2016) present an anomaly detection approach in which the predicted user activity is represented by a task model. The predicted and actual behavior are then compared to see if any variance (anomaly) has occurred. The problem with such model-based anomaly detection approaches is that they fail to detect anomalies that have not previously occurred. Ismail, Hassan, and Alsalamah (2019) propose a context-aware framework for learning and predicting human behavior. Behavior contexts such as weekdays and the time of day are collected from residents' real-life data to improve the accuracy of activity prediction.

Cook and Schmitter-Edgecombe (2009) have developed algorithms for automatically learning separate Markov models for each of the five classes of activity (Telephone Use, Hand Washing, Meal Preparation, Eating and Medication Use, and Cleaning). These models are used to both categorize the activities that are carried out in smart homes and to identify errors and inconsistencies in those activities.

Krishna, Jain, Mehta, and Choudhary (2018) proposed a Long Short-Term Memory (LSTM)-based method for detecting anomalies in daily activity sequences, as well as a comparison of the proposed method with the Hidden Markov Model, which demonstrates comparable results for the LSTM model. Moallem, Hassanpour, and Pouyan (2019) presented an anomaly detection method in smart homes based on deep learning. They used binary sensor data to train a predictor model, which is a recurrent neural network, to predict which sensors will turn on/off and how long the event will last.

Arifoglu and Bouchachia (2019b) examined the problem of dementia-affected older individuals' activity recognition and inappropriate behavior detection. Given the difficulty in getting real-world data, the research first proposes an approach for creating synthetic data that reflects on some behavioral issues of people with dementia. The second part of the study looked at Convolutional Neural Networks (CNNs), which can be used to predict patterns in activity sequences and identify abnormal behavior associated with dementia. The identification of activities is regarded as a sequence labeling issue, and anomalous behavior is highlighted based on a departure from expected patterns. Additionally, the effectiveness of CNNs is evaluated in comparison to cutting-edge techniques like Conditional Random Fields (CRFs), Hidden Semi-Markov Models, Hidden Markov Models, and Naive Bayes (NB). The outcomes show that CNNs are in a competitive position with the listed state-of-the-art methods.

Shang, Chang, Liu, Zhao, and Roy (2020), introduced a mechanism for Feature-based Implicit Irregularity Detection (FIID) that extracts regularity features through unsupervised learning and produces the likelihood of implicit irregularity. According to the proposed FIID, the regular activities that meet the time-regular and happen-frequently qualities are what define everyday behaviors as being regular. The implicit irregularity probability of the daily health state is then calculated using a multidimensional feature space that is built using these features.

Lago, Jiménez-Guarín, and Roncancio (2017) introduced contextualized behavior patterns, a long-term behavior model that takes context-related variability into account and then codifies the key ideas relating to activities in Ambient Assisted Living. This study shows that using semantic similarity makes it easier to detect behavioral changes.

Previous research has explored the use of deep learning models for abnormality detection in Activities of Daily Living (ADL) sequences. Notably, Akbari and Sartipi (2022) leveraged Bi-Directional Encoder Representations from Transformers (BERT) to analyze older adults' ADL sequences for behavior change detection. Their study highlighted the fine-tuning capabilities of Transformers, making them well-suited for supervised tasks even when large labeled datasets are unavailable. They emphasized the significance of transfer learning, showing that fine-tuning a pre-trained model for a new resident enhances detection performance. Their case study on a two-resident ADL dataset with a sequence length of 128 demonstrates empirical support for the effectiveness of Transformer-based approaches in abnormality detection scenarios. However, the model's requirement for longer sequence lengths imposes a limitation, necessitating an extended history of ADL.

While there are plenty of studies on behavior anomaly detection in older adults, temporal features are not utilized to their full potential.

Most of the studies reflect on point anomalies which is when an individual data point is different from the rest of the data. However, collective anomalies that can only be identified by considering the sequential features of data are not explored well. Some works are limited to finding abnormalities within activity classes, while there can be abnormalities that can only be detected by a higher-level analysis of activities. Therefore, appropriate behavior granularity needs to be considered. It is also important for the method to present a generalizable solution that can be tuned for different target users in a reasonable time. This feature would allow the method to start learning the behavior patterns from a pre-learned model as opposed to learning from scratch.

To address the above-mentioned issues, we hypothesize that deep learning RL-based (Reinforcement Learning) methods that have been proven effective in analyzing time series data can also be effectively applied in analyzing ADL data streams for detecting deviations from normal behavior. We propose considering temporal features of behavior to detect collective abnormalities in older adults' behavior. This research considers inter-activity dependencies to understand behavior routines. We also apply state-of-the-art RL-based methods to minimize the need for labeled data. The suggested method will also address the "cold start" issue, in which the algorithm is unable to make any conclusions about residents for whom it has not yet received sufficient training data.

3. Background

This section provides an overview of two key concepts that underpin our proposed approach to behavior abnormality detection in older adults, i.e., Markov Decision Process (MDP) and Inverse Reinforcement Learning. By providing a deeper understanding of these concepts, we can better appreciate the technical and theoretical foundations of our proposed approach and its potential applications in the field of smart home care.

3.1. Markov decision process and reinforcement learning

A process can be considered a Markov Decision Process if the decision to be taken depends only on the current state of the environment. In other words, regardless of the previous states, the agent should be able to take the proper action (make a decision) at any point in time.

Reinforcement Learning (RL) problems can be formulated as Markov Decision Processes. An MDP consists of the following basic elements: a set of states S, a set of actions A, a transition function T, and a reward function R.

A *state* represents the situation of the agent within the environment. In each state, the environment makes a collection of actions available to the agent (an action space) from which the agent can choose an *action*. The agent interacts with the environment through these actions, and in response to the agent's action, the state can change. The transition function determines the state that the agent will arrive in after taking an action.

As a part of the interaction between the agent and the environment, upon the agent's action, the environment passes a *reward* on to the agent using a reward function. The reward provides feedback to the agent about its performance, which can positively or negatively reinforce the agent's behavior. Guiding the agent through feedback can be done by providing either an immediate reward (discount factor of 0) or a discounted reward (0 < discount factor < 1).

The ultimate goal of the agent is to take actions that maximize the accumulated reward over a sequence of actions. The *policy* is referred to as a function that determines what action to take in order to maximize the accumulated discounted reward given the current state of the environment.

3.2. Inverse reinforcement learning

Inverse Reinforcement Learning (IRL) is a subfield of machine learning that aims to learn reward functions from expert demonstrations. Unlike traditional reinforcement learning, which assumes that the reward function is known in advance, IRL seeks to infer the reward function from observed behavior data. This makes IRL particularly useful in settings where the reward function is not well-defined or is difficult to specify in advance. Russell (Russell, 1998) made a suggestion that IRL may be used to provide computational models of difficult-to-specify behaviors in humans and animals. The goal of IRL is to model an agent's preferences based on observed behavior, avoiding the need to manually specify the reward function. The interaction of the observed agent with its environment is typically attributed to a Markov decision process, the solution of which is a policy that maps states to actions. Because the true reward function of this MDP is not directly observable, IRL assumes the agent is following an (unknown) optimal policy, and then works backwards from the observed state–action trajectories to recover the reward function that best explains that optimal behavior.

Formally, the goal of IRL is to find a reward function $R(s,a)$ that explains the observed behavior of an agent in a given environment. The agent's behavior is typically represented as a sequence of state–action pairs, denoted as $\tau = (s_1, a_1, \ldots, s_{T-1}, a_{T-1}, s_T)$, where s_t is the state at time t and a_t is the action taken by the agent in that state. The objective of IRL is to find a reward function that maximizes the likelihood of the observed behavior data:

$$\max_r P(\tau|r)$$

To solve this optimization problem, IRL algorithms typically rely on the Maximum Entropy IRL framework, which assumes that the reward function is a linear combination of features of the state–action pairs:

$$R(s,a) = \sum_{i=1}^{n} w_i \phi_i(s,a)$$

where, $\phi_i(s,a)$ represents the ith feature of the state–action pair, and w_i represents the weight associated with that feature. The goal of the IRL algorithm is to learn the weights w_i that best explain the observed behavior data.

The Maximum Entropy IRL framework also assumes that the agent's behavior is optimal with respect to the learned reward function. This means that the agent's actions are chosen to maximize the expected reward through exploring diverse alternatives.

To learn the weights w_i, IRL algorithms typically use a gradient-based optimization approach, such as the Maximum Causal Entropy IRL algorithm. This algorithm seeks to minimize the difference between the observed behavior data and the behavior predicted by the learned reward function, while also maximizing the entropy of the policy. This results in a reward function that explains the observed behavior data while also being maximally uncertain about the agent's actions. Recent research has also explored the use of deep neural networks to learn reward functions from expert demonstrations. Deep Maximum Entropy IRL (Wulfmeier, Ondruska, & Posner, 2015) is a variant of IRL that uses deep neural networks to model the reward function and policy. Deep Maximum Entropy IRL has several advantages over traditional IRL approaches. First, deep neural networks are capable of capturing complex, non-linear relationships between the state–action pairs and the reward function. This enables the model to learn more accurate and robust reward functions that can better explain the observed behavior data. Second, deep neural networks can handle high-dimensional input data. The basic idea behind Deep Maximum Entropy IRL is to use a deep neural network to model the reward function. The network takes as input the state–action pairs and outputs the weights of the different features in the reward function.

To train the model, Deep Maximum Entropy IRL algorithms typically use a variant of the Maximum Causal Entropy IRL algorithm, which seeks to minimize the difference between the observed behavior data and the behavior predicted by the learned reward function, while also maximizing the entropy of the policy. This results in a reward function that explains the observed behavior data while also being maximally uncertain about the agent's actions. In summary, Deep Maximum Entropy IRL is a powerful technique for learning reward functions from expert demonstrations using deep neural networks. By capturing complex, non-linear relationships between the state–action pairs and the reward function, Deep Maximum Entropy IRL enables us to develop more accurate and robust models that can be applied in a variety of settings.

IRL has drawn a lot of interest from researchers in the fields of artificial intelligence and machine learning (Lin & Cook, 2020; Oh & Iyengar, 2019; Rhinehart & Kitani, 2017) because it satisfies two significant needs (Arora & Doshi, 2021): First, it diminishes the requirement to pre-specify the reward function, which restricts the use of RL and optimal control to issues where a reward function can be simply stated. Second, a reward function can be transferred to another agent and provides a concise representation of an agent's preferences. If the subject agent and the other agent have similar environments and purposes, the learned reward function can be employed exactly as is; otherwise, it continues to serve as a valuable foundation even when the agent specifications are slightly different. In fact, compared to the observed agent's policy, the reward function is naturally more transferrable, as Russell (1998) points out.

4. Approach

In this section, we present our approach to detecting abnormal behavior in older adults using Inverse Reinforcement Learning. We input recent activities of daily living (ADL) into the model to understand the older adult's behavior patterns and intentions. In our previous work, Akbari and Sartipi (2024), we introduced a preliminary model for detecting ADL abnormalities using Inverse Reinforcement Learning. In this experiments and results to evaluate and validate the effectiveness of our proposed model.

As shown in Fig. 1, the proposed method consists of three layers: Input, Process, and Output. Sensor data logged over 2–3 months (baseline period) are processed in the offline IRL module to learn the weights of the feature vector and reward function $R(s,a)$. Then, the online IRL module receives the real-time behavior sequence of the resident and calculates its associated reward. Finally, the fusion center compares the calculated reward with a pre-defined threshold, which represents the average reward for normal sequences, to determine the normality of the real-time behavior.

4.1. Behavior representation

In order for the data to be ready to be processed in the Behavior Change Detection (BCD) module, we need to model human indoor behavior for relatively unconstrained environments.

Considering behavior as a sequence of discrete tokens (sleeping, eating, watching TV, preparing meals, etc.), two important quantities emerge: (i) *Content*: activities that constitute a behavior; and (ii) *Order*: the temporal arrangement of the constituent activities. The idea of tokenizing behavior in this work is similar to the way researchers in Natural Language Processing (NLP) have looked at documents as vectors of their constituent words (see Vector Space Model, VSM (Salton, Wong, & Yang, 1975)). Approaches such as VSM capture the content of a sequence in an efficient way. However, they completely ignore its order. Behavior is not fully defined by its activity content alone; rather, by its natural activity orderings. Therefore, a model to capture activity order in an explicit manner is needed. For this purpose, we consider a sliding window of size W over a behavior sequence to take into account all possible sequences of length T. We consider the start time of ADL as the baseline for the order of tokens in sequences. Therefore, in the case of interleaved ADL, ADL will be put in the sequence according to their start time.

In order to feed the behavior sequence into the BCD module, it needs to have a fixed length. However, behavior sequences can be of any length as people perform a different number of ADL each day. To tackle this issue, we define a sliding window (with a shift delta of 1) that allows for sliding over the dynamic-length sequences and capturing ADL dependencies. In this approach, although the length of sequences is fixed to a predefined value (sliding window length), truncating the sequences does not harm the process of capturing ADL dependencies as the dependency between the token at the truncating point and its pre- or post-tokens will be observed in the previous or next sequences, respectively when the window slides over the original sequence. The sliding window size is a parameter of the model that needs to be determined depending on the contextual features of analysis that the generated data will be used for. For example, if data are to be used for learning short patterns, it makes sense to have a small sliding window.

To determine an appropriate value for T, we need to find a small-enough number that, while it limits model complexity, is suitable for covering a representative sequence of the individual's patterns of behavior. In this paper, we model human behavior B as an ordered sequence of events:

$$B = e_1, e_2, \dots, e_i, \dots, e_W \tag{1}$$

where e_i refers to an event. We define event e_i as a 3-tuple that consists of the activity type a_i, duration d_i, and period-of-day p_i:

$$\begin{aligned} e_i = (a_i, d_i, p_i); where\ \ & a_i \in \{activity\ \ types\}\ and \\ & d_i \in \{activity\ \ duration\ \ range\}\ and \\ & p_i \in \{period\_of\_day\ \ range\} \end{aligned} \tag{2}$$

Then, we reshape B to a flat tensor B' in order to feed it into the algorithm:

$$\begin{aligned} & B' = y_1, y_2, \dots, y_k, \dots, y_T; \\ & \quad where\ \ y_k = a_i\ if\ \ k\ \mathbf{rem}\ 3 = 0\ \ and \\ & \qquad\quad y_k = d_i\ if\ \ k\ \mathbf{rem}\ 3 = 1\ \ and \\ & \qquad\quad y_k = p_i\ if\ \ k\ \mathbf{rem}\ 3 = 2 \\ & \qquad\quad s.t.\quad i = \lfloor \frac{k+2}{3} \rfloor \end{aligned} \tag{3}$$

where T is the window size and equals $3 \times W$. It is worth mentioning that activity type and period-of-day are categorical data that need to be encoded in integers so they can be fed into the BCD module. For activity duration, we also discretize the values so the model deals with categorical values. We believe that, while it does not hurt the accuracy of the model, it simplifies the model by decreasing the state space. As the range of duration in different activity types varies, we first normalize the duration for each activity type, separately. Then, an equal-width discretization method is applied to turn the duration values into categorized values.

4.2. Problem formulation

We represent the Behavior Abnormality Detection problem as a Markov Decision Process. We define the MDP elements as follows:

- ***State*** $s_t \in\ STATES$: a sliding window of size W that represents a sequence of the W latest ADL events that the older adult has performed at time t: $e_{t-W}, \dots, e_t$;
- ***Action*** $a \in\ Actions$: the next ADL event e_{t+1};
- ***Transition*** $T(s_t, a)$: after taking action a in state s_t, the agent transitions to state s' that equals $e_{t-W+1}, \dots, e_t, e_{t+1}$, which slides the behavior window one token forward.

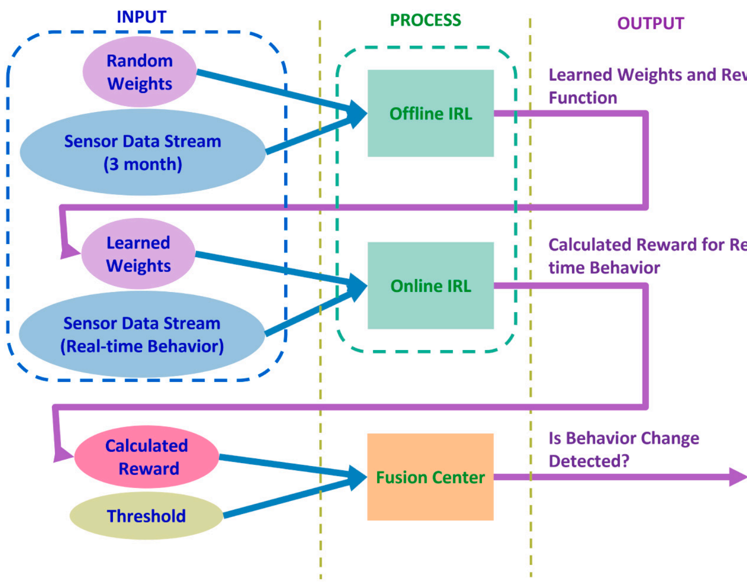

Fig. 1. The IRL-based Behavior Abnormality Detection Model.

We propose an IRL algorithm that estimates the reward function $R(s,a)$ from observations. In this model, observations are trajectories of ADL that are performed by the older adult. We use a discount factor to consider expected future rewards in the long-term reward calculation.

We hypothesize that learning the reward function will enable us to understand unusual ADL sequences. The threshold-based fusion center evaluates the real-time reward and determines the normality of the behavior by comparing the associated reward of the real-time behavior sequence with a predefined threshold R_{th}. In the following blocks, the offline IRL module as well as the online IRL and fusion module are presented.

Algorithm 1: Offline IRL

Require: Expert demonstrations $\tau_e = (s_1, a_1, s_2, a_2, \ldots)$, ADL window size W, Episode length ep_l, Hidden size $hidden_s$, Learning rate lr, Number of epochs num_{epochs}
Ensure: Reward function R

1: Define the reward network R using a neural network with input size W, hidden size $hidden_s$, and output size equal to the number of activity classes
2: Define the optimizer (Adam) and the loss function (CrossEntropy) for the reward network
3: Define a custom Gym environment based on the MDP with parameters (S, A, T, R, γ) , where S is the state space, A is the action space, T is the transition function, and γ is the discount factor.
4: Train the reward network R using the state–action pairs in τ_e and the optimizer and loss function for a specified number of epochs
5: **return** R

In algorithm 1, the action space and the observation space are defined based on the number of activity classes and the number of previous activities, respectively. The reward network R is defined using a neural network with input size W, hidden size $hidden_s$, and output size equal to the number of activity classes. The optimizer and the loss function are also defined. The episode length is defined as ep_l. The log data is converted to state–action pairs, and the reward network R is trained using these pairs and the optimizer and loss function for a specified number of training epochs. The trained reward function R is returned as the output of the algorithm. Further implementation details can be found in Appendix-Algorithm 3 and Algorithm 4.

Algorithm 2: Online IRL and Fusion Module

Require: Real-time ADL Sequence $Input_{seq}$, Reward threshold R_{th}, Reward function R
Ensure: 0 (No Potential Behavior Change is Detected), 1 (Potential Behavior Change is Detected)

1: Pass the $Input_{seq}$ to the reward network and get the output $R(Input_{seq})$ (reward value for each activity class).
2: **if** R[actual action] $\leq R_{th}$ **then**
3: **return** 1
4: **else**
5: **return** 0
6: **end if**

Algorithm 2 includes an online IRL module that receives a trained reward function R, as well as a real-time sequence of ADL and a predefined threshold R_{th} to determine the normality of the behavior sequence. The reward function outputs a reward value for each activity class. In the fusion center, the reward value of the current activity is compared to R_{th} to determine whether the activity conforms to the typical behavior pattern.

5. Results

In the following subsections, we provide the results of our experiments on a real dataset to evaluate the performance of the proposed approach. Presents an evaluation of the proposed approach to behavior abnormality detection in smart homes. The section is divided into two subsections: Dataset and Analysis. In the Dataset subsection, we provide an overview of the real dataset used in our experiments. The Analysis subsection presents the results of our experiments, including a quantitative evaluation of the proposed approach in terms of its ability to detect potential behavior changes. By presenting a thorough evaluation of our proposed approach, we aim to provide a foundation for future research in this area and to inspire new approaches to improving the quality of care for older adults.

Table 1
Example data from CASAS-Aruba dataset.

Date	Time	SensorID	SensorState	Activity
2010–11–04	00:03:50	M003	ON	Sleeping begin
2010–11–04	00:03:57	M003	OFF	
2010–11–04	00:15:08	T002	21.5	
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
2010–11–04	05:40:43	M003	OFF	Sleeping end
2010–11–04	05:40:51	M004	ON	
2010–11–04	05:40:52	M005	OFF	BedToToilet begin
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
2010–11–04	05:43:30	M004	OFF	BedToToilet end

Table 2
Dataset statistics.

Number of records	Number of ADL
1,719,558	6,477

Table 3
ADL types in CASAS-Aruba dataset.

ADL type	Number of records
eating	26
enter_home	83
housekeeping	55
leave_home	147
meal_preparation	128
personal_hygiene	439
sleep_not_in_bed	4
sleeping	246
wandering_in_room	14
watchTV	89

5.1. Dataset

In this section, we introduce the public dataset that will be used for the sake of evaluating the proposed Behavior Change Detection method. The CASAS-Aruba dataset (Cook, 2010) consists of activities that a woman performed at home during a period of seven months. A few examples from this dataset are shown in Table 1. In this dataset, eleven types of indoor activities were included. Meal preparation, Relaxing, Eating, Working, Sleeping, Washing Dishes, Bed to Toilet, Entering Home, Leaving Home, Housekeeping, and Respiration were recorded using motion sensors, door sensors, and temperature sensors. As shown in Table 1, start and end times for each activity were recorded, making it possible to calculate the duration of the activity. Also, the time ordering of activities was captured. Table 2 and Table 3 present some overall statistics on the Aruba dataset.

Fig. 2 illustrates the behavior trend of the Aruba resident for one month. Looking at the plot, we can see that there is a clear pattern in the activities over the course of the month. For example, there are periods where the woman is predominantly sleeping or eating, followed by periods where she is predominantly working or engaging in other activities. Additionally, we can see that there is some variation in the activities from day to day, with some days showing more variety in activities than others.

Forecasting categorical time series data presents unique challenges that require specialized methodologies capable of handling discrete variables. Markov Chain models are particularly well-suited for this task, as they predict the probability of future states based on observed patterns in the data. These models operate under the assumption that the future state of a categorical variable depends solely on the present state, disregarding any prior states. This property allows for efficient estimation of state transition probabilities, facilitating accurate forecasting in various applications.

However, first-order Markov models, which consider only the immediate past state, may not capture the complex dependencies inherent in many real-world scenarios. To address this limitation, higher-order Markov models extend the dependency to multiple preceding states. By incorporating information from several past states, these models can better account for intricate patterns and temporal dependencies in the data. Ching, Fung, and Ng (2004) demonstrated that higher-order Markov models could significantly enhance forecasting accuracy in categorical time series by capturing these extended dependencies.

In our approach, we represent the current state by concatenating the last N activities, effectively creating a window that spans N time steps. This method captures temporal dependencies that a first-order model would overlook. By considering a sequence of past activities, the model gains access to more historical information, enabling it to make more informed predictions and decisions.

Within the framework of Reinforcement Learning (RL), the Markov Decision Process (MDP) relies on the Markov property, which states that the future is independent of the past given the present state. By defining the state to include the last N activities, we preserve the Markov property because all relevant historical information is encapsulated within the current state. Nonetheless, the underlying process remains influenced by multiple past states, characteristic of a higher-order Markov model.

This modeling approach has significant practical implications. By formulating the problem as a higher-order Markov model, we can capture more complex patterns and dependencies in the data. This leads to more accurate predictions and the development of more effective policies when training an RL agent. Consequently, our model can better adapt to the nuances of categorical time series data, improving overall forecasting performance and decision-making processes.

5.2. Analysis

In this study, we developed an IRL model to detect behavior changes in older adults from ADL data and evaluated its performance using the CASAS dataset.

We split the CASAS dataset into train and test sets with a 70–30 ratio. Using the train set, we trained our inverse reinforcement learning model to associate reward values to each action (activity class) in a given state (ADL sequence) from sequences of activities of daily living (ADL) data. We then evaluated the model's performance on the test set.

Table 4 shows the activity codes and their corresponding activity labels. These codes are used to identify different activities that are performed by the individual. For example, the code "4" represents the activity of bathing for a long duration in the morning, and the code "18" represents the activity of transitioning from bed to toilet for a short duration at midnight. Overall, this table serves as a reference to understand the codes that are used to represent these activities in the following graphs.

To provide an overview of the dataset, we first generated a bar chart showing the distribution of data over various activity classes (Fig. 3). The chart revealed that the dataset is imbalanced, with some activity classes occurring more frequently than others.

We trained the model using **Adam** optimizer, learning rate of 0.001 and **window size** of $W = 10$ on Colab environment with a T4 GPU, 14 GB of system RAM and 15 GB of GPU RAM. We monitored its training progress by tracking the **cross-entropy loss** over the 1000 epochs of training. Fig. 4 shows the line chart of the model's training loss. As can be seen, the model's loss decreases from 12 to 2.5 over the 1000 epochs of training, indicating that the model is learning to assign higher rewards to next activities that conform to the behavior pattern of the individual and lower rewards to abnormal activities. This suggests that the model is able to capture the underlying patterns in the data.

The decrease in cross-entropy loss over the training epochs indicates that the model is learning to minimize the difference between its

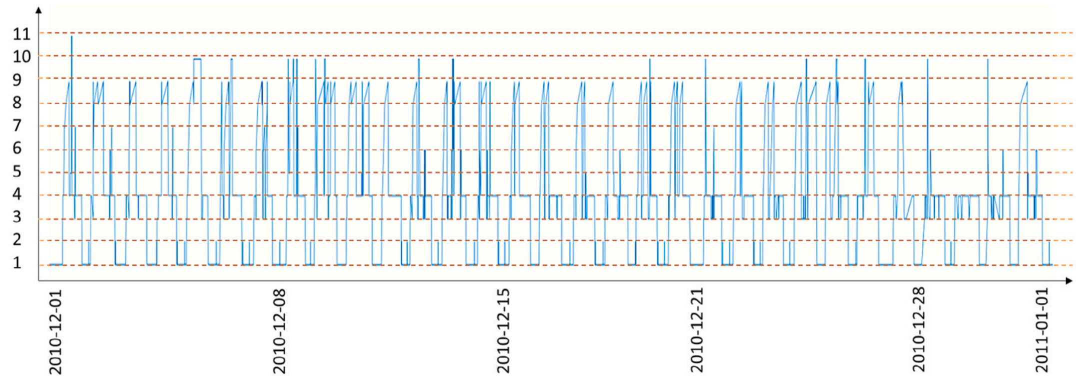

Fig. 2. The Behavior Trend of the Aruba Resident over Time.

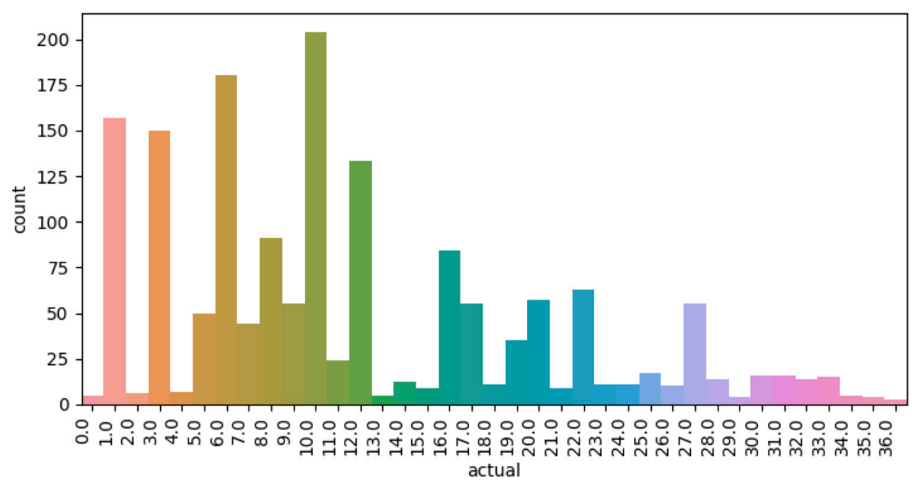

Fig. 3. The distribution of Activity classes in the Train Set.

predicted activity classes and the actual activity classes. This is an important feature of the model, as it allows us to detect behavior changes in older adults more accurately and efficiently. Next, we analyzed the model's performance using a HeatMap that displays the normalized average reward for each predicted and actual activity class (Fig. 5). The X-axis represents the predicted activity class by the IRL model, while the Y-axis represents the actual activity class. The lighter colors in the HeatMap indicate a higher reward, while the darker colors indicate a lower reward. We observed that the Heatmap's diagonal is apparent, which indicates that the trained model is able to give high rewards to activity classes that match the actual activity class. This suggests that the model can correctly identify the majority of activity classes. Additionally, we noticed that there are dark cells corresponding to each actual activity class, which indicates that the model is able to identify activity classes that are not very probable to occur in some states. This is an important feature of the model, as it allows us to identify anomalies in the data that may indicate behavior changes.

However, we also observed that apart from the light cells in the diagonal, there are other light cells present in the HeatMap. This is because, in each state, there is more than one single activity class that is possible to occur due to the diverse nature of the behavior patterns of an individual. This suggests that the model may sometimes predict multiple activity classes with similar probabilities. To further evaluate the model's performance, we define a metric 'Alternative Activity Reward Accuracy (AARA)' to measure how accurately the model assigns high rewards to activity classes that could be alternatives to the true activity classes. We do this by calculating the ratio of high-reward activity classes present in the training set when added to the end of the current state S.

Also, 'Low-Reward Irrelevance Rate (LRIR)' metric calculates the proportion of low-reward activity classes that are not present in the training set when added to the end of the current state S. A higher ratio suggests that a significant portion of low-confidence predictions are correctly identifying irrelevant classes.

$$\begin{aligned} \text{AARA} &= \frac{\sum_{i=1}^{n} \mathbb{1}(r_i > \theta \wedge s\&a_i \in \text{TrainSet})}{\sum_{i=1}^{n} \mathbb{1}(r_i > \theta)} \\ \text{LRIR} &= \frac{\sum_{i=1}^{n} \mathbb{1}(r_i < \theta \wedge s\&a_i \notin \text{TrainSet})}{\sum_{i=1}^{n} \mathbb{1}(r_i < \theta)} \end{aligned} \tag{4}$$

where:

- n is the total number of predictions.
- r_i is the reward for the predicted activity class i.
- θ_h is the threshold for high rewards.
- θ_l is the threshold for low rewards.
- s is the current state (i.e. previous W-1 ADL events)
- a_i is the predicted activity class.
- $\mathbb{1}$ is the indicator function, which is 1 if the condition is true and 0 otherwise.
- TrainSet is the set of ADL sequences in the training data.

θ_h is set to 0.85 while θ_l is set to 0.15 according to reward statistics and the fact that rewards are normalized. An AARA of **0.96** and an LRIR of

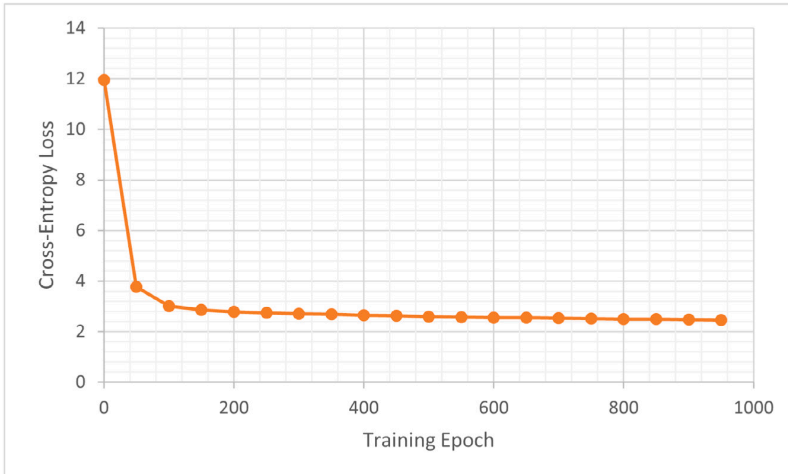

Fig. 4. Cross-entropy Loss over 1,000 Epochs of Training.

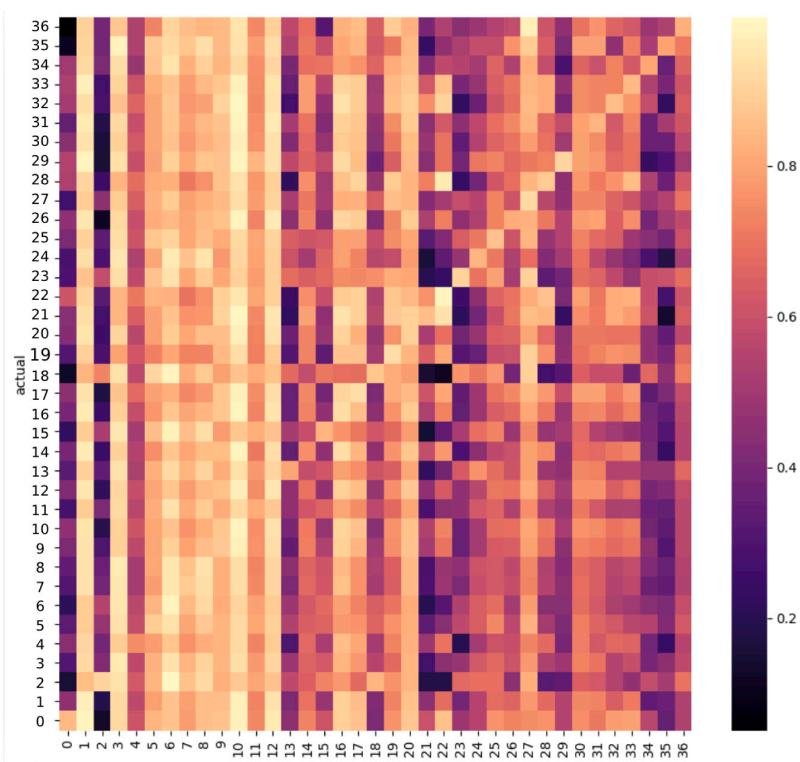

Fig. 5. The average reward for predicted activity classes in the Train Set.

0.93 indicate that the model performs significantly well in capturing the behavior patterns.

The HeatMap graph clearly shows that certain activities cannot be substituted with others in a typical situation. For instance, sleeping for a medium duration at night cannot be replaced with bed to toilet transition, eating, or a long personal hygiene. Additionally, less frequently occurring activities like wandering in the room are generally associated with lower average reward values, except when they actually occur. This is reflected in the graph as the diagonal cells for such activities remain light, while almost all other cells in the column are dark.

To ensure that our IRL model was able to generalize well, we evaluated its performance on a separate test set that was not used during the training phase. We used the same bar chart and HeatMap visualizations to demonstrate the test set performance, as we did for the training set. Fig. 6 shows the bar chart for the test set, which has a similar distribution of data across the various activity classes

Table 4
The mapping of activity classes to activity codes.

Activity label	Activity code
Bathing (Long, Morning)	4
Bathing (Medium, Morning)	24
Bathing (Short, Morning)	15
Bed to Toilet Transition (Short, Midnight)	18
Bed to Toilet Transition (Short, Night)	2
Eating (Medium, Night)	21
Eating (Short, Night)	32
Enter Home (Short, Midnight)	33
Enter Home (Short, Morning)	14
Enter Home (Short, Night)	9
Leave Home (Medium, Midnight)	36
Leave Home (Short, Midnight)	19
Leave Home (Short, Morning)	8
Leave Home (Short, Night)	30
Meal Preparation (Medium, Night)	26
Meal Preparation (Short, Midnight)	28
Meal Preparation (Short, Morning)	7
Meal Preparation (Short, Night)	20
Personal Hygiene (Long, Morning)	25
Personal Hygiene (Long, Night)	34
Personal Hygiene (Medium, Midnight)	23
Personal Hygiene (Medium, Morning)	5
Personal Hygiene (Medium, Night)	31
Personal Hygiene (Short, Midnight)	27
Personal Hygiene (Short, Morning)	3
Personal Hygiene (Short, Night)	12
Sleep (Medium, Night)	29
Sleep (Short, Midnight)	17
Sleep (Short, Morning)	11
Sleep (Short, Night)	1
Wandering in Room (Short, Morning)	13
Watch TV (Short, Morning)	35
Watch TV (Short, Night)	16
Work (Short, Midnight)	22
Work (Short, Morning)	6
Work (Short, Night)	10
Wandering in Room (Short, Night)	0

as the training set. This indicates that the test set is representative of the overall dataset and that the model is able to generalize effectively beyond the training data. Overall, the evaluation of the IRL model on the test set provides further evidence of its robustness and effectiveness in accurately predicting activity patterns and rewards for residents in smart homes. Fig. 7 shows the HeatMap for the test set, which displays similar results as the HeatMap for the training set. The diagonal is apparent, indicating that the model is able to accurately predict the majority of activity classes, and there are dark cells corresponding to each actual activity class, suggesting that the model is able to identify activity classes that are not very probable to occur in some states. Additionally, there are light cells present in the HeatMap, which suggests that the model is able to predict multiple activity classes with similar probabilities in some states.

The fact that our IRL model produced similar results for the test set as for the training set suggests that the model is not overfitted to the training data and can effectively generalize to new, unseen data. This is a crucial feature of the model, as it enables us to apply it to new datasets with confidence, thereby improving our ability to detect behavior changes in older adults more accurately and efficiently.

The robustness of our IRL model is particularly important in the context of homecare, where residents' behavior patterns can vary widely and change over time. By accurately predicting these patterns and detecting any changes, our model can help caregivers and researchers to better understand the needs and preferences of individual residents, and to develop tailored interventions that improve their quality of life. Overall, the ability of our IRL model to effectively generalize to new datasets is a significant advantage that enhances its practical utility in real-world care settings.

5.2.1. Synthetic sequence injection

To evaluate the ability of our models to identify behavior changes, we introduced synthetic abnormal sequences into the dataset. Based on existing literature, changes in physical activity levels, alterations in rest periods between tasks, changes in sleep patterns, forgetting to complete tasks, and repeating tasks are all included in the symptom profiles of diseases such as Alzheimer's, heart disease, urinary tract infections, diabetes, and others. We therefore introduced these changes into the activities of daily living (ADL) sequences in the CASAS dataset. We also inject samples of behavior abnormalities by rearranging ADL and manipulating activity duration. For example, while in the original ADL sequences eating occurs after meal preparation, we reversed the ADL order to inject partially misordered sequences.

- ADL duration alterations: In 50% of the abnormal sequences, we modify ADL durations by randomly scaling ADL lengths by ±20%, ±60%, and ±90%.
- ADL rearrangement: In 30% of sequences, we swap the order of two semantically related ADLs (e.g., swapping "meal preparation" and "eating") to create a partially misordered pattern.
- Random shuffling of ADLs: In a further 20% of sequences, we randomly permute the entire ADL list, disrupting all temporal dependencies.

Table 5 provides a summary of abnormalities injected into the data. After injecting %10 synthetic abnormal sequences, SMOTE method is used to oversample the abnormal sequences to ensure a balanced dataset. We used the augmented labeled dataset to evaluate the performance of our fusion module.

Table 6 presents the performance metrics for the proposed approach, including accuracy, precision, recall, and F1 score, at different threshold values. The results show that using lower threshold values increases the number of false positives, indicating that more normal ADL sequences are incorrectly classified as abnormal. Conversely, higher threshold values result in a decrease in recall, indicating that the model is more likely to miss abnormal cases. We recommend selecting a threshold value that balances precision and recall. To aid in this decision, we also report the F1 score, which is the harmonic mean of precision and recall. This score provides a single metric that combines both precision and recall, making it useful for selecting an appropriate threshold.

To compare the performance of our proposed model with baseline models, we implemented both an LSTM classifier and a Transformer-based classifier. In our implementation of the LSTM model, we adopted the architecture proposed by Zerkouk and Chikhaoui (2019), utilizing a hidden size of 64 and an embedding size of 200. For the BERT model, we employed the 'AutoModelForSequenceClassification' from the Hugging Face Transformers library, using the pretrained 'bert-base-uncased' model.

To ensure a fair, threshold-independent comparison, we report both F1 at each model's optimal decision threshold (selected via a validation-set grid search over [0.1, 0.9]) and ROC-AUC, which does not require threshold tuning. Further, to account for the randomness inherent in SGD-based training, we retrained every method (ours and all baselines) 10 times with different random seeds. Table 7 reports the mean ± standard deviation of F1, precision, recall, accuracy and ROC-AUC over these runs. According to Table 7, both LSTM and BERT achieve substantially higher recall than precision, indicating many false positives on the positive class. Their imbalance in recall versus precision is reflected in moderate F1 scores (0.67 and 0.69, respectively) and relatively lower ROC-AUC values (0.79 and 0.82). By contrast, the IRL model not only attains the highest F1 (0.76) but also leads in ROC-AUC (0.90), demonstrating a more balanced trade-off between true-positive and false-positive rates and superior threshold-independent performance.

Our experimental results indicate that BERT and LSTM are not particularly effective in this context, despite their success in various other applications. We attribute this to the limited size of our

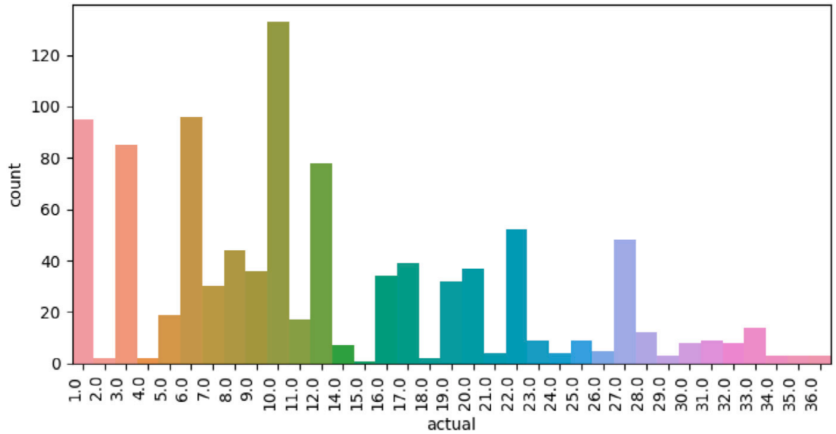

Fig. 6. The distribution of Activity classes in the Test Set.

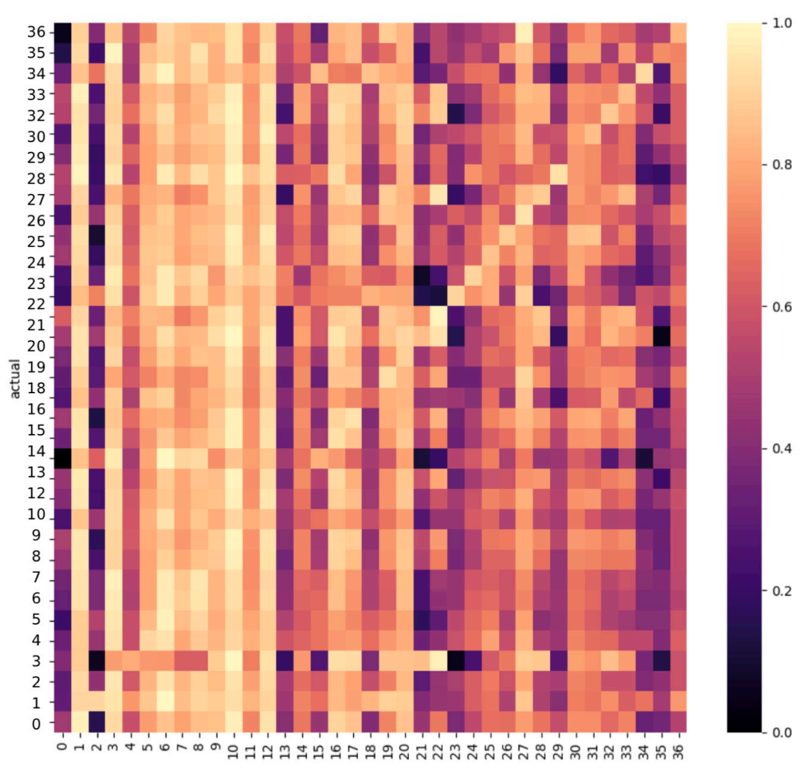

Fig. 7. The average reward for predicted activity classes in the Train Set.

Table 5
Summary of injected abnormality types.

Alteration type	Fraction injected	Parameters
Duration alterations	50%	Scale ADL durations by ±20%, ±60%, ±90%
ADL rearrangement	30%	Swap order of two related ADLs (e.g., Meal Prep ↔ Eating)
Random shuffling	20%	Random permutation of entire ADL sequence

training dataset, which is insufficient for effectively training deep neural network models such as these. Additionally, the pre-training of BERT involves general-domain text sequences, which may not translate well to the specific nuances of ADL sequences without substantial fine-tuning. LSTM networks also face challenges in managing longer sequences, which can further affect performance.

Table 6
Evaluation metrics for different thresholds.

Threshold	Accuracy	Recall	Precision	F1
0.6	0.73	**0.87**	0.65	0.744
0.75	**0.74**	0.84	0.66	0.739
0.85	0.73	0.81	0.72	**0.762**
0.9	**0.74**	0.76	**0.73**	0.744

Table 7
Evaluation metrics for baseline models; Mean ± standard deviation of evaluation metrics over 10 independent runs; F1 is reported at each classifier's optimal threshold.

Model	Accuracy	Recall	Precision	F1 (best-threshold)	ROC-AUC
LSTM	0.55 ± 0.03	$\mathbf{0.89 \pm 0.04}$	0.54 ± 0.05	0.672 ± 0.02	0.78 ± 0.03
BERT	0.51 ± 0.02	0.86 ± 0.03	0.58 ± 0.04	0.693 ± 0.02	0.82 ± 0.02
IRL (th=0.85)	$\mathbf{0.73 \pm 0.01}$	0.81 ± 0.02	$\mathbf{0.72 \pm 0.01}$	$\mathbf{0.762 \pm 0.02}$	$\mathbf{0.90 \pm 0.01}$

To enhance the adaptability of Transformer models like BERT to specialized tasks, future research should prioritize fine-tuning with domain-specific data. This approach could potentially mitigate the limitations observed in our study and improve model efficacy in similar applications.

The results demonstrate that our IRL model effectively detects behavior changes in older adults from ADL sequences with high accuracy. The model successfully identifies activity classes unlikely to occur in certain states, aiding healthcare professionals in detecting anomalies and potential behavior changes.

Our evaluation on the augmented dataset highlights the effectiveness of our approach in identifying behavior changes linked to various diseases. By accurately detecting these changes, our method has the potential to enhance the quality of care provided to residents in smart homes.

6. Conclusion

This research utilizes state-of-the-art Inverse Reinforcement Learning algorithms to address the problem of behavior abnormality detection in smart home settings. The proposed model introduces a novel representation of an individual's recorded activities of daily living (ADL) as a higher-order Markov Decision Process. An offline IRL algorithm is then used to infer the underlying reward function of the individual, followed by an online IRL algorithm in collaboration with the fusion center to determine the abnormality of the observed behavior.

We evaluated the effectiveness of the proposed approach using an augmented real dataset, and the results showed that the model is capable of detecting potential behavior changes with an F1 score of %76.2. This demonstrates the model's ability to accurately identify abnormal behavior patterns in smart home residents, providing caregivers and researchers with a valuable tool for improving the quality of care and developing tailored interventions.

While the proposed model represents a significant advancement in the field of behavior abnormality detection, there are also some limitations that should be noted. One such limitation is the fact that as the length of the ADL sequences increases, there is a corresponding increase in the size of the state space, which can make it challenging to train the model effectively.

Specifically, as the length of the ADL sequences grows, the number of possible states in the MDP increases exponentially. This means that with an increase in the length of the ADL sequences, there is a need for more training data to adequately cover the expanded state space. This can be a significant challenge in practice, particularly when dealing with limited or sparse datasets.

Despite this limitation, our proposed model represents a promising approach to behavior abnormality detection in smart home settings. By leveraging the power of IRL algorithms and machine learning techniques, we can gain deeper insights into the behavior patterns of residents and develop more effective interventions to improve their quality of life.

CRediT authorship contribution statement

Fateme Akbari: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Kamran Sartipi:** Writing – review & editing, Methodology, Conceptualization.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the authors used GPT-4o in order to improve language. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix

Algorithm 3: RewardNet Neural Network

Data: Input tensor x of size `W`
Result: Output tensor of size `No_ActivityClasses`
Class RewardNet{
Initialize(W, hidden_size, No_ActivityClasses, dropout_prob=0.5);
  Define fully connected layer $fc1$: $\mathbb{R}^{W} \rightarrow \mathbb{R}^{hidden\_size}$;
  Define activation function $ReLU$;
  Define dropout layer $dropout$: Dropout with probability $dropout\_prob$;
  Define fully connected layer $fc2$: $\mathbb{R}^{hidden\_size} \rightarrow \mathbb{R}^{No\_ActivityClasses}$;
Forward(x);
  $x \leftarrow fc1(x)$;
  $x \leftarrow ReLU(x)$;
  $x \leftarrow dropout(x)$;
  $x \leftarrow fc2(x)$;
  return x;

Algorithm 4: ADLEnv Environment

```
Data: Log file path log_file, window size W, episode length
    ep_len
Result: Trained Reward Network
Class ADLEnv inherits from gym.Env;
Initialize(log_file, W, ep_len, hidden_size = 32, lr = 0.001,
    num_epochs = 100);
begin
    Load activity logs from log_file;
    Define activity_classes as a dictionary mapping
        activity names to indices;
    Initialize action_space as Discrete with size equal to
        number of activities;
    Initialize observation_space as MultiDiscrete
        with W activities;
    Initialize reward_net with input size W, hidden size
        hidden_size, and output size equal to number of activities;
    Set optimizer to Adam with learning rate lr;
    Set loss function to CrossEntropyLoss;
    Initialize state as a zero vector of size W;
    Set current_step to 0;
    Set episode length to ep_len;
end
Function reset();
begin
    Reset state to zeros;
    Reset current_step to 0;
    return state;
end
Function step(action);
begin
    Compute reward using get_reward(state, action);
    Update state by appending action and removing the oldest
        activity;
    Increment current_step;
    Set done to True if current_step ≥ ep_len;
    return state, reward, done, {};
end
Function train_reward_net(save_path);
begin
    Prepare input-output pairs from log data using window size
        W;
    Convert inputs and targets to tensors;
    for epoch = 1 to num_epochs do
        Forward pass: compute outputs from
            reward_net(inputs);
        Compute loss using loss_fn(outputs, targets);
        Backpropagation: optimize reward_net parameters;
        if epoch mod 50 == 0 then
            Print current epoch and loss;
        end
    end
    Save reward_net parameters to save_path;
end
Function get_reward(obs, action);
begin
    Convert observation obs to input tensor;
    Forward pass through reward_net to get output tensor;
    return reward corresponding to action;
end
```

Data availability

Data will be made available on request.

References

Akbari, F., & Sartipi, K. (2022). A transformer-based model for older adult behavior change detection. In *2022 IEEE 10th international conference on healthcare informatics* (pp. 27–35). IEEE.

Akbari, F., & Sartipi, K. (2024). A model for detecting abnormality in activities of daily living sequences using inverse reinforcement learning. In *Proceedings of the 39th ACM/SIGAPP symposium on applied computing* (pp. 1031–1033). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3605098.3636085.

Arifoglu, D., & Bouchachia, A. (2019a). Abnormal behaviour detection for dementia sufferers via transfer learning and recursive auto-encoders. In *2019 IEEE international conference on pervasive computing and communications workshops (perCom workshops)* (pp. 529–534). IEEE.

Arifoglu, D., & Bouchachia, A. (2019b). Detection of abnormal behaviour for dementia sufferers using convolutional neural networks. *Artificial Intelligence in Medicine*, *94*, 88–95.

Arora, S., & Doshi, P. (2021). A survey of inverse reinforcement learning: Challenges, methods and progress. *Artificial Intelligence*, *297*, Article 103500.

Casagrande, F. D., Tørresen, J., & Zouganeli, E. (2018). Sensor event prediction using recurrent neural network in smart homes for older adults. In *2018 international conference on intelligent systems* (pp. 662–668). IEEE.

Ching, W. K., Fung, E. S., & Ng, M. K. (2004). Higher-order Markov chain models for categorical data sequences. *Naval Research Logistics*, *51*(4), 557–574.

Cook, D. J. (2010). Learning setting-generalized activity models for smart spaces. *IEEE Intelligent Systems*, *2010*(99), 1, Publisher: NIH Public Access.

Cook, D. J., & Schmitter-Edgecombe, M. (2009). Assessing the quality of activities in a smart environment. *Methods of Information in Medicine*, *48*(05), 480–485.

Durand, V. M., & Barlow, D. (2003). *Abnormal Psychology*. Toronto: Thomson Learning.

Fahad, L. G., & Tahir, S. F. (2021). Activity recognition and anomaly detection in smart homes. *Neurocomputing*, *423*, 362–372.

Freitas, M., de Aquino Piai, V., Dazzi, R., Teive, R., Parreira, W., Fernandes, A., et al. (2022). Identification of abnormal behavior in activities of daily life using novelty detection. In *International conference on mobile and ubiquitous systems: computing, networking, and services* (pp. 559–570). Springer.

Ismail, W. N., Hassan, M. M., & Alsalamah, H. A. (2019). Context-enriched regular human behavioral pattern detection from body sensors data. *IEEE Access*, *7*, 33834–33850.

Karakostas, A., Briassouli, A., Avgerinakis, K., Kompatsiaris, I., & Tsolaki, M. (2016). The dem@ care experiments and datasets: a technical report. arXiv preprint arXiv:1701.01142.

Krishna, K., Jain, D., Mehta, S. V., & Choudhary, S. (2018). An lstm based system for prediction of human activities with durations. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, *1*(4), 1–31.

Lago, P., Jiménez-Guarín, C., & Roncancio, C. (2017). Contextualized behavior patterns for change reasoning in ambient assisted living: A formal model. *Expert Systems*, *34*(2), Article e12163.

Lin, B., & Cook, D. J. (2020). Analyzing sensor-based individual and population behavior patterns via inverse reinforcement learning. *Sensors*, *20*(18), 5207.

Lotfi, A., Langensiepen, C., Mahmoud, S. M., & Akhlaghinia, M. J. (2012). Smart homes for the elderly dementia sufferers: identification and prediction of abnormal behaviour. *Journal of Ambient Intelligence and Humanized Computing*, *3*(3), 205–218.

Moallem, M., Hassanpour, H., & Pouyan, A. (2019). Anomaly detection in smart homes using deep learning. *Iranian (Iranica) Journal of Energy & Environment*, *10*(2), 126–135.

Nazerfard, E. (2018). Temporal features and relations discovery of activities from sensor data. *Journal of Ambient Intelligence and Humanized Computing*, 1–16.

Oh, M. h., & Iyengar, G. (2019). Sequential anomaly detection using inverse reinforcement learning. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 1480–1490).

Rhinehart, N., & Kitani, K. M. (2017). First-person activity forecasting with online inverse reinforcement learning. In *Proceedings of the IEEE international conference on computer vision* (pp. 3696–3705).

Riboni, D., Bettini, C., Civitarese, G., Janjua, Z. H., & Helaoui, R. (2015). Fine-grained recognition of abnormal behaviors for early detection of mild cognitive impairment. In *2015 IEEE international conference on pervasive computing and communications (perCom)* (pp. 149–154). IEEE.

Russell, S. (1998). Learning agents for uncertain environments. In *Proceedings of the eleventh annual conference on computational learning theory* (pp. 101–103).

Salton, G., Wong, A., & Yang, C.-S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, *18*(11), 613–620.

Shang, C., Chang, C.-Y., Liu, J., Zhao, S., & Roy, D. S. (2020). FIID: feature-based implicit irregularity detection using unsupervised learning from IoT data for homecare of elderly. *IEEE Internet of Things Journal*, *7*(11), 10884–10896.

Stineman, M. G., Xie, D., Pan, Q., Kurichi, J. E., Saliba, D., & Streim, J. (2011). Activity of daily living staging, chronic health conditions, and perceived lack of home accessibility features for elderly people living in the community. *Journal of the American Geriatrics Society*, *59*(3), 454–462.

Suryadevara, N. K., Mukhopadhyay, S. C., Wang, R., & Rayudu, R. (2013). Forecasting the behavior of an elderly using wireless sensors data in a smart home. *Engineering Applications of Artificial Intelligence*, *26*(10), 2641–2652.

Urwyler, P., Stucki, R., Rampa, L., Müri, R., Mosimann, U. P., & Nef, T. (2017). Cognitive impairment categorized in community-dwelling older adults with and without dementia using in-home sensors that recognise activities of daily living. *Scientific Reports*, *7*(1), 1–9.

Wulfmeier, M., Ondruska, P., & Posner, I. (2015). Maximum entropy deep inverse reinforcement learning. arXiv preprint arXiv:1507.04888.

Zerkouk, M., & Chikhaoui, B. (2019). Long short term memory based model for abnormal behavior prediction in elderly persons. In *How AI impacts urban living and public health: 17th international conference* (pp. 36–45). Springer.