A Model for Detecting Abnormality in Activities of Daily Living Sequences Using Inverse Reinforcement Learning

Abstract—**ADL abnormality detection has been the focus of many recent healthcare studies, some of which addressed the issue by using deep learning techniques. In this paper, we provide a novel approach for examining ADL sequences to detect meaningful deviations from the individual's routine behavior. This approach can benefit older adults in several ways, including timely care, early detection of health conditions to stop them from getting worse, reducing the burden of monitoring on family members, and maximizing self-sufficiency without interfering with daily activities. We present an Inverse Reinforcement Learning (IRL)-based method for detecting behavior abnormalities in older adults through the analysis of ADL sequences. To do this, we model the problem of abnormality detection in behavior sequences as a Higher-order Markov Chain model. Using the IRL method, from observed trajectories of behavior, we infer the reward function that drives the individual to perform ADLs. The inferred reward function will then be utilized to detect potential behavior abnormalities through a threshold-based mechanism.**

1. Introduction

With the world's aging population, the need for technologies supporting healthy aging and independent living for older adults is growing. Aging often brings behavioral changes that may indicate cognitive decline or other health issues. Detecting these changes can enable early intervention by caregivers and healthcare professionals, potentially improving older adults' health outcomes and quality of life.

Tracking changes in behavior over time using ADLs data, which includes information about daily activities like eating, sleeping, and personal hygiene, is valuable. However, it is challenging to detect behavior changes from ADL data due to the complexity and variability of human behavior. Advancements in sensor technologies have promoted the monitoring of ADLs. Decreased ADL performance is associated with the progression of chronic diseases and cognitive impairment in older adults [23]. 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. [25]. Therefore, the study of life patterns in older persons can be used to quantify changes relevant to ADLs 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 detecting behavior abnormalities in older adults. The model infers the reward function from observed expert behavior, referred to as trajectories. It learns from sequences of ADLs performed by individuals to capture the underlying motivations. IRL has shown promise in modeling human behavior and inferring underlying motivations [13]. By leveraging Inverse Reinforcement Learning, the model trains the agent through a semi-supervised task, which is effective when defining the reward function is complex. Additionally, the reward function has been demonstrated to exhibit greater transferability compared to the policy function [20], leading to the development of more generalizable models. Unlike supervised methods, the model learns from expert observations rather than labeled data. It can adapt to changes in behavior over time and detect 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.

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" [9]. Because the concept of an anomaly is difficult to define precisely and is closely tied to patient behaviors and the types and course of pathology, 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 ADLs 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 [2, 14, 19, 24].

Fahad et al. [10] 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 et al. [5] 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 ADLs are taken into consideration to forecast the next activity. Nazerfard [16] presents an association rule mining module that identifies associations among ADLs that are grouped according to the start time and duration of the related ADL. The sequence of the activities is also taken into account. Cook et al. [8] 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 were used to both categorize the activities that are carried out in smart homes and to identify errors and inconsistencies in those activities. Krishna et al. [11], 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 et al. [15] 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 et al. [3] examined the problem of dementia-affected older individuals' activity recognition and abnormal 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 et al. [22], 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 et al. [12] 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.

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 IRL. By fostering a comprehensive comprehension of these concepts, we can enhance our recognition of the technical and theoretical underpinnings of our proposed approach and its potential applications in the realm 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.

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

IRL is a subfield of machine learning that aims to learn a reward function from expert demonstrations. Unlike traditional RL, 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 [20] 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 an MDP, the solution of which is a policy that maps states to actions. Since the reward function of this MDP is unknown, it is presumed that the observed agent adheres to the MDP's ideal policy.

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 $\tau = (s_0, a_0, s_1, a_1, ..., s_{T-1}, a_{T-1}, s_T)$, 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:

$$\max_r P(\tau|r) \tag{1}$$

To solve this optimization problem, IRL algorithms often employ the Maximum Entropy IRL framework. In this framework, the reward function is represented as a feature vector, $\mathbf{r}$, where each element corresponds to a specific feature influencing the agent's behavior. A simple representation of the reward function is a linear combination of state-action pair features, given by:

$$r(s, a) = \sum_{i=1}^{n} w_i \phi_i(s, a) \tag{2}$$

Here, $\phi_i(s, a)$ represents the i-th feature, and w_i represents the associated weight. The goal is to learn the weights that best explain the observed behavior data.

To learn these weights, IRL algorithms, such as the Maximum Causal Entropy IRL (MaxEnt IRL) algorithm, utilize a gradient-based optimization approach. MaxEnt IRL seeks to minimize the difference between observed behavior data and the behavior predicted by the learned reward function while maximizing the policy's entropy. This results in a reward function that explains the observed behavior while exhibiting maximal uncertainty about the agent's actions. The algorithm achieves this by maximizing the causal entropy, $H[r]$, which quantifies uncertainty. The probability of an expert taking action a in state s is modeled using a Softmax function, given by:

$$P(a|s) = \frac{\exp(Q(a, s))}{\sum \exp(Q(a', s))} \tag{3}$$

Here, $Q(a, s)$ represents the expected reward (or action-value) associated with action a in state s. The MaxEnt IRL algorithm aims to find the reward function $\mathbf{r}$ that maximizes the causal entropy $H[r]$ while being consistent with the observed expert behavior. The causal entropy $H[r]$ is defined as:

$$H[r] = -\sum (P(a|s) \log P(a|s)) \tag{4}$$

By solving an optimization problem using maximum entropy principles, MaxEnt IRL finds the reward function that best explains the observed expert behavior while maximizing the uncertainty about the true reward function. This allows the algorithm to capture a wide range of possible reward functions and provide a robust estimate of the underlying rewards in the given environment.

Recent research has also explored the use of deep neural networks to learn reward functions from expert demonstrations. Deep Maximum Entropy IRL [26] 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.

IRL has drawn a lot of interest from researchers in the fields of artificial intelligence and machine learning [13, 18, 17] because it satisfies two significant needs [4]: Firstly, it alleviates the need for pre-specifying the reward function, thereby removing the limitation of RL and optimal control to problems that can be easily defined with a reward function. 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 [20] points out.

4. Approach

In this section, we present our approach to detecting abnormal behavior in older adults using IRL. We input recent ADLs into the model to understand the older adult's behavior patterns and intentions.

As shown in Figure 1, the proposed method works in 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 Assessment Module 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

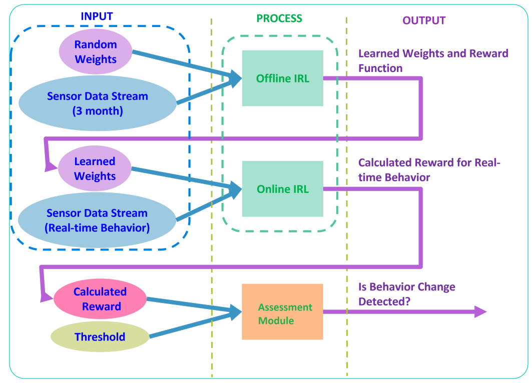


Figure 1: The IRL-based Behavior Abnormality Detection Model

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 [21]). 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 W.

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 ADLs each day. To tackle this issue, we use a sliding window (with a shift delta of size 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 W, 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.

We consider the start time of ADLs as the baseline for the order of tokens in sequences. Therefore, when dealing with interleaved ADLs, wherein multiple activities are executed concurrently or overlap with each other, with one activity commencing before another is finished, it becomes necessary to organize the ADLs in a sequential order based on their respective start times [1]. In this paper, we model human behavior B as an ordered sequence of events with the size W:

$$B = e_1, e_2, ..., e_i, ..., e_W \tag{5}$$

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

$$\begin{aligned} e_i = (a_i, d_i, p_i); where \ \ & a_i \in \{activity \ \ types\} \ and \\ & d_i \in \{activity \ \ duration \ \ range\} \ and \\ & p_i \in \{period\_of\_day \ \ range\} \end{aligned} \tag{6}$$

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

$$\begin{aligned} B' = y_1, y_2, ..., y_k, ..., y_{W'}; & \\ where \ \ y_k = a_i \ if \ k \ \mathbf{rem} \ 3 = 0 \ \ and & \\ y_k = d_i \ if \ k \ \mathbf{rem} \ 3 = 1 \ \ and & \\ y_k = p_i \ if \ k \ \mathbf{rem} \ 3 = 2 & \\ s.t. \quad i = \lfloor \frac{k+2}{3} \rfloor & \end{aligned} \tag{7}$$

where W' 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 Higher-order Markov Chain Model. 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: $e_{t-W}, ..., e_t$;

Action $a_t \in$ *Actions*: the next ADL event e_{t+1};

Transition $T(s_t, a_t)$: after taking action a_t in state s_t, the agent transitions to state s_{t+1} that equals $e_{t-W+1}, ..., e_t, e_{t+1}$, which slides the behavior window one token forward.

We propose an IRL algorithm that estimates the reward function $r(s, a)$ from observations. In this model, observations are trajectories of ADLs that are performed by the older adult. We use a discount factor γ 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 Assessment Module determines the normality of the behavior by comparing the associated reward of the online behavior sequence with a predefined threshold R_{th}. In the following blocks, the Offline IRL Module as well as the online IRL and Assessment Module are presented.

Algorithm 1: Offline IRL

Require: Expert demonstrations $\tau_e = (s_1, a_1, s_2, a_2, ...)$, ADL window size W, Episode length ep_l, Hidden size $hidden_s$, Learning rate lr, Number of epochs num_epochs
Ensure: Reward function r
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
Define the optimizer (Adam) and the loss function (CrossEntropy) for the reward network
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.
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
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.

Algorithm 2: Online IRL and Assessment 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)

Pass the $Input_{seq}$ to the reward network and get the output $r(Input_{seq})$ (reward value for each activity class).
if r[actual action] $<= R_{th}$ **then**
  Return 1
else
  return 0
end if

Algorithm 2 includes an Online IRL Module that receives a trained reward function R, as well as a real-time sequence of ADLs and a predefined threshold R_{th} to determine the normality of the behavior sequence. The

TABLE 1: Example Data from CASAS-Twor Dataset.

Date	Time	SensorID	SensorState	Activity
2009-08-24	00:04:38	M047	ON	Sleep begin
2009-08-24	00:04:40	M047	OFF	
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
2009-08-24	00:05:21	M046	ON	Sleep end
2009-08-24	00:05:23	M048	ON	Wandering_in_room begin
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
2009-08-24	00:07:54	P001	576	

reward function outputs a reward value for each activity class. In the Assessment Module, the reward value of the current activity is compared to R_{th} to determine whether the activity conforms to the typical behavior pattern.

TABLE 2: ADL Types in CASAS-Twor 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
Total	1,231

5. Results

In the following subsections, we provide the results of our experiments on a real dataset and present and evaluate the proposed approach to behavior abnormality detection in smart homes. The section is divided into two subsections 5.1. Dataset, which provides an overview of the real dataset used in our experiments, and 5.2. Analysis, that 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.

5.1. Dataset

In this subsection, we introduce the public dataset that we used to evaluate the proposed Behavior Change Detection method. The CASAS-Twor dataset [7] represents sensor events collected in the WSU smart apartment testbed during the 2009-2010 academic year. The apartment housed two residents, R1 and R2, at this time and they performed their normal daily activities. A few examples from this dataset are shown in Table 1. In this dataset, thirteen types of indoor activities including bathing, bed to toilet transition, eating, entering and leaving home, housekeeping, meal preparation, personal hygiene, sleep, sleeping not in bed, wandering in room, watch TV and work are recorded using motion sensors, item sensors, door sensors, burner sensors, hot water sensors, cold water sensors, temperature sensors and electricity usage. A total of 1,402,405 readings are recorded. 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 presents some overall statistics on the Twor dataset.

Forecasting categorical data time series requires specialized techniques that can handle the nature of the data. Markov Chain models are designed to handle categorical data and can help to forecast the probability of future states or categories based on the observed patterns in the time series data. Markov Chain models are based on the assumption that the future state of a categorical variable depends only on the present state, and not on any previous states. This model can be used to predict the probability of future state transitions based on the current state.

Higher-order Markov chain models are probabilistic models that extend the concept of first-order Markov models by incorporating dependencies on preceding states beyond the immediate one. This characteristic makes these models valuable for forecasting categorical time series data, particularly when the categorical variable is affected by multiple past states [6]. In contrast to first-order Markov chain models, which solely consider the present state to determine the future state of a categorical variable, higher-order Markov chain models take into account the influence of multiple previous states. By considering these additional dependencies, higher-order Markov chain models can yield more accurate predictions, especially in scenarios where the categorical variable exhibits intricate patterns and dependencies. Nevertheless, it is important to note that higher-order models may entail greater computational complexity and data requirements compared to their first-order counterparts.

5.2. Experimentation and 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 IRL model to associate reward values to each action (an instance of activity classes; i.e., event) in a given state (i.e., ADL sequence). We then evaluated the model's performance on the test set.

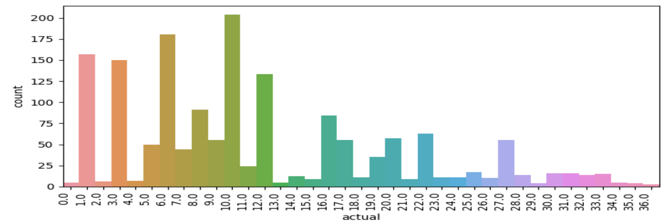
Table 3 provides a cross-reference between the activity labels and their corresponding activity codes. An activity label represents a combination of an activity and an encoded time-duration and zone. For example, the code ”4” represents the activity of bathing for a long duration in the morning, and the code ”18” represents the activity of transitioning from bed to toilet for a short duration at midnight. Table 3 serves as a reference to understand the codes that are used in the following graphs.

Activity Class Label	Activity Class 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
Wandering in Room (Short, Night)	0
Watch TV (Short, Morning)	35
Watch TV (Short, Night)	16
Work (Short, Midnight)	22
Work (Short, Morning)	6
Work (Short, Night)	10

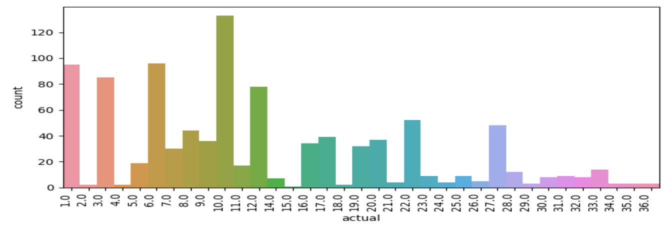
TABLE 3: Activity Class Codes

To provide an overview of the dataset, we first generated bar charts showing the distribution of train and test data over various activity classes (Figures 2a and 2b, respectively). The charts revealed that the dataset is imbalanced, with some activity classes occurring more frequently than others. Train and test sets have similar distributions of data across the various activity classes. This indicates that the test set is representative of the overall dataset.

We monitored the model's training progress by tracking the cross-entropy loss over 1000 epochs of training. Figure 3 shows the line chart of the model's training loss. As it can be seen, the model's loss decreases from 12 to 2.5 indicating that the model is learning to assign higher rewards to the 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. Such a decrease in cross-entropy loss indicates that the model is learning to minimize the difference between its predicted activity classes and the actual activity



(a) The distribution of activity classes in the Train Set



(b) The distribution of activity classes in the Test Set

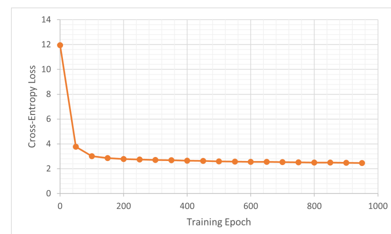


Figure 3: Cross-entropy Loss over 1,000 Epochs of Training

classes. Therefore, the trained model will allow us to detect behavior changes in a person at home 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 (Figure 4). The X-axis represents the predicted activity class by the IRL model, and 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. Let us take a closer look at row 22 (Figure 5), which represents the actual activity "Work (Short, Midnight)." By examining this row (referred to as row A), we can gain valuable insights from the HeatMap.

First, in cell A22, we observe the lightest color on the HeatMap. This indicates a high reward, suggesting that the IRL model accurately predicted and matched the actual activity class. Similarly, cell A10, corresponding to the activity "Work (Short, Night)," also displays a light color, indicating another successful prediction.

However, there are several cells in row A that stand out with darker colors. For instance, cells A2, A13, A23, and A35 represent the activities "Bed to Toilet Transition (Short, Night)," "Wandering in Room (Short, Morning)," "Personal Hygiene (Medium, Midnight)," and "Watch TV (Short, Morning)," respectively. The darker colors in these cells suggest that the model considers these activities less

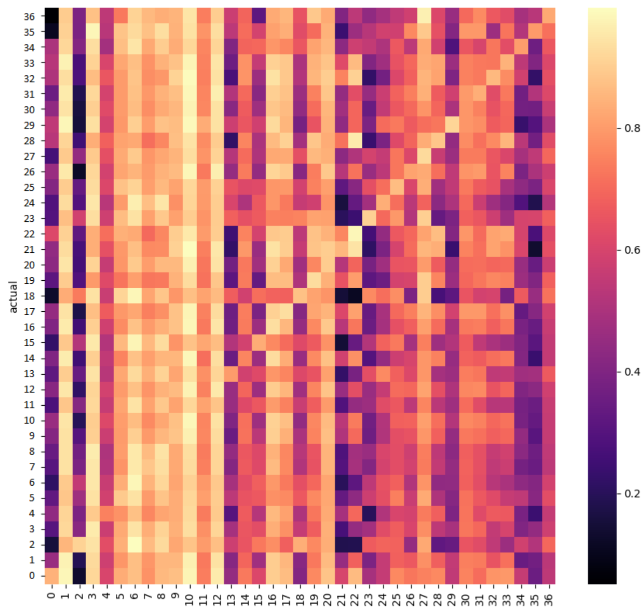


Figure 4: The average reward for predicted activity classes in the Train Set

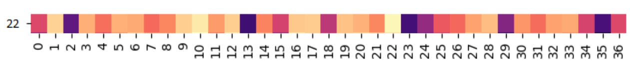


Figure 5: A Sample row from the HeatMap

probable to occur compared to the actual activity "Work (Short, Midnight)." We can observe that the Heatmap's diagonal cells in the train set have pale colors, indicating that the trained model assigns high rewards to activity classes that match the actual activity class. This implies that the model can accurately identify the majority of activity classes. Furthermore, we noticed the presence of darker cells corresponding to each actual activity class, indicating the model's ability to identify activity classes that are unlikely to occur in certain states. This feature is crucial as it enables us to detect anomalies in the data that may signify behavioral 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 in the same state with similar probabilities.

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 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 HeatMap visualizations to demonstrate the test set performance, as we did for the training set. 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. In Figure 6, depicting the HeatMap for the test set, we can observe similar trends as the HeatMap for the training set. The diagonal pattern is evident, suggesting that the model accurately predicts the majority of activity classes. Additionally, there are dark cells corresponding to each actual activity class, indicating the model's capability to identify less probable activity classes in certain states. Furthermore, we can observe the presence of lighter cells in the HeatMap, implying that the model predicts 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.

5.3. Embedding Abnormal Activities

To evaluate the ability of our models to identify behavior changes, we embedded synthetic abnormal sequences into the dataset. Based on existing literature, changes in physical activity levels, alterations in the 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, embedded such abnormal sequences into the ADL sequences of the CASAS dataset. After injecting these abnormalities, we used the augmented labeled dataset to evaluate the performance of our Assessment Module in terms of detecting embedded abnormal activities.

Table 4 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

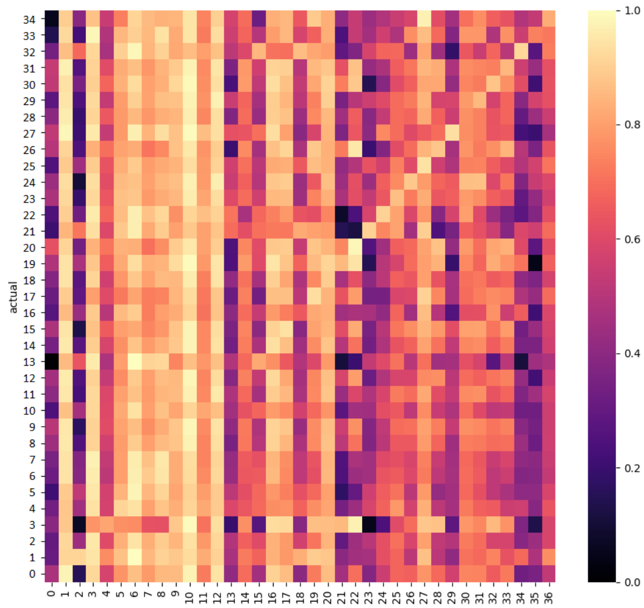


Figure 6: The average reward for predicted activity classes in the Train Set

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 used a threshold value that balances the precision and recall. To support 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.

The evaluation of the augmented dataset demonstrated the effectiveness of our approach in identifying behavior changes associated with different diseases. By accurately detecting these changes, our approach has the potential to enhance the quality of care provided to residents in smart homes.

Threshold	Accuracy	Recall	Precision	F1-Score
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 4: Evaluation metrics for different thresholds.

In order to compare the performance of our proposed approach with the most powerful sequential models, Table 5 presents the results of the comparison between the proposed approach and the LSTM model for detecting abnormal behavior sequences. The proposed approach achieved an accuracy of 73%, outperforming the LSTM model, which achieved an accuracy of 69%. This indicates that the proposed approach is more effective in accurately identifying abnormal behavior patterns in the given dataset. Moreover, the proposed approach also demonstrated higher precision (72% vs. 71%), recall (81% vs. 77%), and F1-score (76.2% vs. 73.8%) compared to the LSTM model. These results suggest that the proposed approach yields better overall performance in terms of both precision and recall, striking a balance between correctly identifying abnormal behavior instances and minimizing false positives. Thus, the proposed approach shows promise as an advanced method for detecting abnormal behavior sequences in the context of smart home environments.

TABLE 5: Comparison of Abnormal Behavior Detection Approaches

Approach	**Accuracy**	**Recall**	**Precision**	**F1-Score**
Proposed Approach	73	81	72	76.2
LSTM Model	69	77	71	73.8

Overall, the results demonstrate that our IRL model is able to detect behavior changes in older adults from sequences of ADL data with a high degree of accuracy. The model is able to identify activity classes that are unlikely to occur in some states, which can help healthcare professionals detect anomalies and potential behavior changes.

6. Conclusion

This research utilizes state-of-the-art IRL 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 ADLs as a higher-order MDP. 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 Assessment Module 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 limitation is the increase in state space size with longer ADL sequences, posing challenges in effective model training.

As the length of ADL sequences grows, the MDP's state space exponentially expands, necessitating more data to cover it adequately. This is particularly challenging when working with limited or sparse datasets.

Nevertheless, our proposed model shows promise for behavior abnormality detection in smart home settings. By combining IRL algorithms and machine learning techniques, we can gain insights into resident behavior patterns and develop effective interventions to enhance their quality of life.

References

[1] F. Akbari, K. Sartipi, and N. Archer. Synthetic behavior sequence generation using generative adversarial networks. *ACM Transactions on Computing for Healthcare*, 2022.

[2] K. A. Alaghbari, M. H. Md. Saad, A. Hussain, and M. R. Alam. Activities recognition, anomaly detection and next activity prediction based on neural networks in smart homes. *IEEE Access*, 10:28219–28232, 2022. doi: 10.1109/ACCESS.2022.3157726.

[3] D. Arifoglu and A. Bouchachia. Detection of abnormal behaviour for dementia sufferers using convolutional neural networks. *Artificial intelligence in medicine*, 94:88–95, 2019.

[4] S. Arora and P. Doshi. A survey of inverse reinforcement learning: Challenges, methods and progress. *Artificial Intelligence*, 297:103500, 2021.

[5] F. D. Casagrande, J. Tørresen, and E. Zouganeli. Sensor event prediction using recurrent neural network in smart homes for older adults. In *2018 International Conference on Intelligent Systems (IS)*, pages 662–668. IEEE, 2018.

[6] W. K. Ching, E. S. Fung, and M. K. Ng. Higher-order markov chain models for categorical data sequences. *Naval Research Logistics (NRL)*, 51(4):557–574, 2004.

[7] D. J. Cook and M. Schmitter-Edgecombe. Assessing the quality of activities in a smart environment. 48(5): 480–485. Publisher: Schattauer GmbH.

[8] D. J. Cook and M. Schmitter-Edgecombe. Assessing the quality of activities in a smart environment. *Methods of information in medicine*, 48(05):480–485, 2009.

[9] V. M. Durand and D. Barlow. Abnormal psychology, 2003.

[10] L. G. Fahad and S. F. Tahir. Activity recognition and anomaly detection in smart homes. *Neurocomputing*, 423:362–372, 2021.

[11] K. Krishna, D. Jain, S. V. Mehta, and S. Choudhary. 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, 2018.

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

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

[14] A. Lotfi, C. Langensiepen, S. M. Mahmoud, and M. J. Akhlaghinia. Smart homes for the elderly dementia sufferers: identification and prediction of abnormal behaviour. *Journal of ambient intelligence and humanized computing*, 3(3):205–218, 2012.

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

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

[17] M.-h. Oh and G. Iyengar. Sequential anomaly detection using inverse reinforcement learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & data mining*, pages 1480–1490, 2019.

[18] N. Rhinehart and K. M. Kitani. First-person activity forecasting with online inverse reinforcement learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3696–3705, 2017.

[19] D. Riboni, C. Bettini, G. Civitarese, Z. H. Janjua, and R. Helaoui. Fine-grained recognition of abnormal behaviors for early detection of mild cognitive impairment. In *2015 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 149–154. IEEE, 2015.

[20] S. Russell. Learning agents for uncertain environments. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 101–103, 1998.

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

[22] C. Shang, C.-Y. Chang, J. Liu, S. Zhao, and D. S. Roy. 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, 2020.

[23] M. G. Stineman, D. Xie, Q. Pan, J. E. Kurichi, D. Saliba, and J. Streim. 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, 2011.

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

[25] P. Urwyler, R. Stucki, L. Rampa, R. Müri, U. P. Mosimann, and T. Nef. 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, 2017.

[26] M. Wulfmeier, P. Ondruska, and I. Posner. Maximum entropy deep inverse reinforcement learning. *arXiv preprint arXiv:1507.04888*, 2015.