Cognitive Errors Detection: Mining Behavioral Data Stream of People with Cognitive Impairment

Jianguo Hao, Sébastien Gaboury, Bruno Bouchard

LIARA Laboratory
Université du Québec à Chicoutimi (UQAC)
555, Boulevard de l'Université
Chicoutimi (Québec), Canada, G7H 2B1
{jianguo.hao1, sebastien.gaboury, bruno.bouchard}@uqac.ca

ABSTRACT

People with cognitive impairment have difficulties in planning and correctly undertaking activities of daily living due to severe deterioration in cognitive skills. As a promising solution, smart homes try to make these people live on their own with less nursing care by providing appropriate cognitive assistance while carrying out activities. For the sake of providing adequate assistance, it is necessary to understand the real intentions of residents and recognize possible anomalous trends in time during the process of performing an activity. In this paper, we analyze the abnormal behavioral patterns caused by cognitive deficits and summarize them as cognitive errors which appear frequently among people with cognitive impairment. Cognitive error detectors are designed and integrated into a unified inference engine based on Formal Concept Analysis theory. The inference engine establishes a knowledge graph hierarchically representing the interrelations between indexed activities to recognize ongoing activities, and to detect predefined cognitive errors in behavioral data streams.

CCS Concepts

•**Human-centered computing** → **Ambient intelligence;**

Keywords

Anomaly detection, sequential data analysis, formal concept analysis, data mining, ambient assisted living.

1. INTRODUCTION

Population ageing has become a crucial issue due to worldwide low fertility rates. As a common feature associated with ageing, cognitive impairment has drawn increasing attention from multidisciplinary scientific communities [1]. Besides, various physical and mental illnesses, even certain drug combinations could also cause deficits in cognition [2]. Severe deterioration in cognitive skills will induce memory difficulties, lapse of judgment and disability to schedule and undertake multiple everyday activities [3]. Compared with healthy people, people with cognitive impairment will normally cause more anomalous behaviors while performing diverse activities of daily living (ADLs), and need more care in daily life.

Because of the cognitive deficits in the planning, memory and attention aspects, living independently is difficult for people with cognitive impairment [4]. Without extra help, normal activities such as cooking activities or adherence of medical instruction could become dangerous as well [2]. The increasing needs of proper cognitive remediation lead to the emergence of smart homes, which are a typical representative of pervasive assistive environments [5].

With rapid development of information and communication technology in recent years, computer scientists would like to merge ubiquitous computing with artificial intelligence and human-computer interaction technologies to construct pervasive assistive environments providing daily health monitoring and timely interventions to older adults [6] or people in need of care. Such intelligent environments devote to offer considerate ambient assisted living experience for their residents. Moreover, the enhanced version designed for special population like people with cognitive impairment desires to draw support from cognitive assistance to avoid some potential daily threats caused by cognitive deficits. For example: forgetting to turn off the stove, excessive sodium & sugar consumption, or unintentional misuse of medication, etc.

In smart homes, ubiquitous computing involves the interactions between residents and pervasive sensor network. Resident's behavioral information and ambient states, including position, motion, device usage, energy consumption and ambient change, could be frequently collected by various hybrid sensors in a period of time [7]. These sequential data which contain temporal evolution of behaviors are essential to analyze the ongoing activity and to detect anomaly. Abnormal behavioral patterns are also hidden in these data. To achieve cognitive assistance, we should first know which activity is being done, and then identify abnormal behavioral patterns through given data streams.

The rest of the paper is structured as follows: we discuss related work about sequential anomaly detection in the

PETRA '16, June 29-July 01, 2016, Corfu Island, Greece

DOI: http://dx.doi.org/10.1145/2910674.2910689

next section. Section III introduces basic notions about our proposed inference engine. Section IV defines common cognitive errors according to the observed behavioral features and gives out corresponding solutions. Section V presents our evaluations in several criteria and the performance of inference engine will also be discussed. Finally, Section VI concludes the paper.

2. RELATED WORK

Human behavior analysis is an important issue for human-computer interaction. Considering privacy protection, we abandon common intrusive solutions based on a camera network using pattern recognition [8] and choose non-intrusive sensor-based sequential data analysis as our research object.

As a more general issue of cognitive errors detection, sequential anomaly detection has been tried to be resolved in different ways like machine learning, data mining and applied mathematics [9]. Within a sensor network, low-level temporal sensor data could be segmented into stages. Each stage describes an atomic action making up high-level activities. In other words, temporal data with behavioral information buffered in a particular period could be referred as *a sequence of actions*. Similar relations are also proposed in [10][11]. As a consequence, sequential data are temporal and contextual, and related to each other [12]. To avoid existing large semantic gaps between low-level sensor data and high-level activities [13], in this paper, we choose intermediate-level actions and high-level activities as analytic granularity.

So far, we could summarize that sequences of actions with cognitive errors are a kind of contextual anomaly because normal behaviors or actions could only be anomalous in specific context [12]. However, they are usually hard to be detected. Firstly, Because of contextual and dynamic forms, it is hard to ensure that all possible anomalous situations are considered and covered in the training data sets. Moreover, labeling anomalous samples are also prohibitively expensive [12]. Fortunately, compared with normal patterns, anomalous ones are far fewer in the training data and dissimilar under particular criteria. Therefore, most of the solutions are based on these two assertions and classified as similarity-based and frequency-based methods.

Similarity-based methods are based on the assumption that normal sequential data are dissimilar in several criteria. Thus, these solutions usually focus on the methods of machine learning, like pattern classification or cluster analysis. [14] defined a similarity scoring function using longest common subsequent (LCS) to determine abnormal human behaviors among low-level sensor data. [15] clustered activities in temporal aspect and used Markov chain model to measure whether a sequence of activities is abnormal or not. [10] used a hidden semi-Markov model and durations of activities to detect abnormal deviations from normal patterns. Besides, [16] proposed a domain-independent formalism to classify possible errors.

Frequency-based methods are based on the assumption that patterns containing cognitive errors occur rarely in the training data set. They try to identify abnormal patterns with low occurrences which are seemingly biased towards the normal ones. [17] presented a model based on the support vector machine to filter out most of the normal activities, and then handle suspicious ones using kernel nonlinear regression (KNLR) model for further detection.

A key limitation of these previous studies is that they do not address the customization problem and more or less ignore the behavioral features of anomalous patterns. Thus, it is easy to suffer from high missing and false alarm rates.

In this paper, we analyze significant features existing in the behavioral data streams of people with cognitive impairment, and summarize common cognitive errors from abnormal patterns. Customer-built solutions will be proposed for each predefined error.

In our earlier work, intermediate-level actions are extracted from low-level temporal sensor data by RFID-based localization [18] and electrical devices identification [7]. Real-time activity recogntion has been realized with high accuracy in [11]. Moreover, some abnormal behavioral patterns were also analyzed in [5][19]. On the basis of these previous works, in this paper, we propose a new inference engine based on the Formal Concept Analysis (FCA) to detect predefined cognitive errors in sequences of actions.

3. PRELIMINARIES

Our inference engine is mainly made up of two parts: activity recognition agent (AR) and cognitive errors detection agent (CED). As the prerequisite for cognitive errors detection, AR agent is created on the basis of FCA theory to recognize ongoing activities in real-time, and the implementation of CED agent relies on it to detect particular features in the patterns. In the following, we briefly review the most fundamental notions of the formal concept analysis and some key implementation details of AR agent to help us better understand the principle and logic behind CED agent. More detailed explanation of AR agent is presented in [11].

3.1 Formal Concept Analysis

Formal concept analysis (FCA) is a subfield of applied mathematics which conceptually clusters associated entities and creates hierarchical interrelations among them by mathematical order theory [20]. FCA could well describe binary relations between two sets of data. Considering our issue, we introduce each FCA component in the following words.

3.1.1 Formal Context (Context)

Context is a triplet $\mathcal{K} = \langle G, M, I \subseteq G \times M \rangle$ where two non-empty sets G and M separately denote universes of activities and actions. Set I is a subset of the Cartesian product $G \times M$ representing binary relations between activities and actions. Context $\mathcal{K}$ could be visualized as a cross table consisting of $|G|$ rows and $|M|$ columns. If m is an integral action of activity g, there must have a cross in row g and column m in the corresponding cross table (see Figure. 1), expressed as gIm [20].

3.1.2 Concept-Forming Operations

Concept-forming operations are a pair of operations that are essential to generate formal concepts. Their main contribution is to quickly match all shared elements. For $A \subseteq G$, we define:

$$A' := \{m \in M \mid \text{for each } g \in A,\ gIm\} \tag{1}$$

to obtain the entire common actions that are simultaneously shared by the full activities in A.

Likewise, for $B \subseteq M$, we define:

$$B' := \{g \in G \mid \text{for each } m \in B,\ gIm\} \tag{2}$$

to obtain the entire activities that simultaneously own the full actions in B.

3.1.3 Formal Concept (Concept)

Concept is a pair $(A \subseteq G, B \subseteq M)$ satisfying the conditions of $A' = B$ and $B' = A$ (i.e. $A'' = A$ or $B'' = B$) under context $\mathcal{K}$. A and B are respectively named as the *extent* and *intent* of concept. In our issue, an intent should be a subset of observable actions, and the extent is a set of all possible activities possessing the full actions in the intent. Each concept is also a stable closure under the *Concept-Forming Operations*. $\mathcal{B}(G, M, I)$ denotes a universe containing all the concepts of $\mathcal{K}$.

	m_1	m_2	m_3	m_4	m_5	m_6
g_1	×	×				
g_2	×	×	×			
g_3	×			×		
g_4	×	×	×	×		×
g_5	×					
g_6					×	×

Figure 1: an example of cross table

3.1.4 Concept Lattice (Lattice)

Lattice is an ordered lattice structure. Suppose that (A_1, B_1) and (A_2, B_2) are two concepts of $\mathcal{K}$, (A_1, B_1) is called *subconcept* of (A_2, B_2) if either $A_1 \subseteq A_2$ or $B_2 \subseteq B_1$. Meanwhile, (A_2, B_2) is a *superconcept* of (A_1, B_1), and written as $(A_1, B_1) \preceq (A_2, B_2)$. The relation $\preceq$ is called *hierarchical order* (*partially order* or simply *order*) that orders and connects such two concepts.

All the concepts in the universe $\mathcal{B}$ ordered by $\preceq$ form a *concept lattice* $\underline{\mathcal{B}}(G, M, I)$. Concept lattice is another closure in $\mathcal{K}$. Once a stable concept has been activated by external factors, it will become unstable and transform into another adjacent stable concept.

3.1.5 Hasse Diagram

Hasse diagram is a sort of line diagram in mathematical order theory that could represent all finite ordered sets [20]. It is also the visualization of concept lattice that represents all concepts as nodes in its graph-based multilayer structure. Each layer connects the others with *Galois connection* [20]. There are two special nodes in the diagram: the topmost node $\{G, \varnothing\}$ called *Supremum* and the lowermost node $\{\varnothing, M\}$ called *Infimum*. Figure 2 is the Hasse diagram of the concept lattice of the cross table in Figure 1.

3.2 Activity Recognition

As we referred above, a stable concept will be activated while its inner structure is being changed. According to predefined hierarchical orders, an unstable concept will transit to one of its connected concepts in the Hasse diagram to become stable again. Such transition is also the principle of activity recognition: when an intent is progressively extended by incoming actions, the scope of possible candidate activities is reduced.

As shown in Figure 2, the topmost Supremum indicates the initial state of recognition process and the lowermost Infimum indicates the final state. The intent of a concept

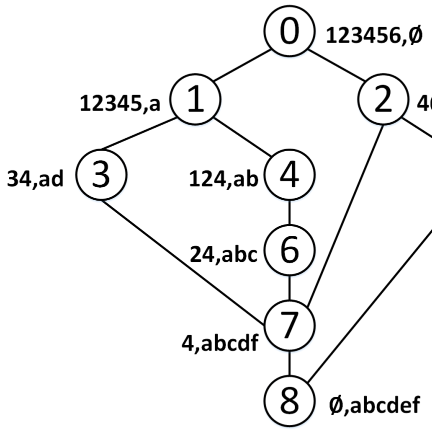

Figure 2: Hasse diagram

indicates the given observed actions and the extent indicates all possible ADLs.

With the successive loading of actions (referred as intent extensions), the cardinality of extent indicating possible activities will be progressively reduced (referred as extent reductions). Our FCA-based proposition about activity recognition concentrates on locating the topmost concept containing α. More details are presented in [11].

Suppose that a sequence of actions α is successively loaded by data stream $a \prec c \prec b \prec f \prec d$. Considering Figure 2, in the first extension, when $\alpha \leftarrow a$, node 1 is located. When $\alpha \leftarrow c$, $\alpha = \{ac\}$, node 6 is located and possible activities would be $\{24\}$. At the end of the extensions, corresponding concept is located at node 7 and activity g_4 is recognized. The trace of the locations is $1 \preceq 6 \preceq 7$.

As can be seen from Figure 1, $g_1 \subset g_2 \subset g_4$ are three quite similar activities sharing too many actions. This phenomenon is called multilevel inheritance. In AR agent, we adopt the greedy strategy to match the longest pattern until the end of the sequence. Thus, when g_1 or g_2 have been recognized, we continue the recognition process to determine whether there are new actions coming to accomplish g_3.

4. COGNITIVE ERROR DETECTORS

Due to cognitive deficits, people with cognitive impairment have difficulties in performing self-care tasks on their own. The sporadic memory loss has frequently occurred during their performance of an activity [5], and they tend to produce more abnormal behavioral patterns than healthy people. In this section, we summarize common cognitive errors and discuss how to detect them based on their behavioral features.

4.1 Problem Settings

For the reason of varied living habits or other external factors, an activity could be described by different sets of actions having different optional actions. Even if having the same set, two sequences of actions could be totally different because of different execution orders and repetitive actions. Thus, an activity usually has N_i derivative sequences possessing j different sets of actions ($N_i \gg j$). Before representing our solutions, we formally define a sequence of actions captured by sensor network in smart homes.

To simplify the description about our research issue, in the real scenarios, we define that a sequence of actions α_j describing activity A_i should be a container (not a set) of:

- essential actions set $E = \bigcap_{i=1}^{N_i} \alpha_i$, which contains all essential actions existing in all N_i derivative sequences of A_i.

  For example, 'boil water' and 'pour water into a teacup' are two essential actions for the activity 'prepare a cup of tea', because they exist in any execution sequence α_i describing a process of making a cup of tea, no matter who does it.

- optional actions set $O = \bigcup_{i=1}^{N_i} \alpha_i - \bigcap_{i=1}^{N_i} \alpha_i$, which indicates different optional actions for A_i.

  For example, 'add milk' could be somebody's personal taste when drinking tea, but not exist in all the sequences describing 'prepare a cup of tea'. So it is a typical optional action.

- possible irrelevant actions set I that $I \cap \bigcup_{i=1}^{N_i} \alpha_i = \emptyset$.

  For example, 'take out pasta from cabinet' is an irrelevant action for 'prepare a cup of tea' and it will not exist in any its normal execution sequence.

- possible redundant actions $R \subseteq \bigcup_{i=1}^{N_i} \alpha_i$, which contains all the actions existing in entire N_i derivative sequences of A_i.

So we give out our generic symbolic representation of a sequence of actions α_j in the form of a triplet:

$$\alpha_j = (\{E \cup O' \cup I' \cup R'\}, \preceq_j, C) \tag{3}$$

where $O' \subseteq O$, $I' \subseteq I$, $R' \subseteq R$, and $\preceq_j$ defines a possible permutation of the union (i.e. an execution order). C is a set of causal constraints limiting the permutation $\preceq_j$. Thus, we could assert that α_j is a normal sequence of actions without cognitive errors if and only if set E is complete, sets I' and R' are empty, and $\preceq_j$ satisfies all the constraints in C.

From the definitions above, we could find out that different sets and their permutations play key roles in the constitution of cognitive errors. In the following words, we present how to detect each error using our inference engine.

4.2 Cognitive Error Definitions

In this part, by observing and tracking the daily life of people with cognitive impairment, first of all, we analyze abnormal behavioral patterns appearing in their activities of daily living. And then, through behavioral features analysis, we give out costumer-built solutions for each summarized cognitive errors.

4.2.1 Initialization

Initialization problem is related to the short-term memory loss. The typical behavioral feature is about doing nothing at the beginning phase while performing an activity. A simple solution is to set a temporal threshold to detect whether a resident starts to do something at the early stages or not. In this paper, initialization error will not be considered in the parts of evaluation and discussion.

4.2.2 Omission of essential actions

An omission is a failure to do something that ought to be done, but was forgotten, according to the initial planning. It is a very usual scenario in daily life, even for healthy persons. Sometimes, there is only a limited influence for performing an activity. For example, there is no big deal if a resident forgets to do the optional actions in set O like personal preferences. However, in most of the time, the omission of essential actions will disrupt the integrity of the implementation (e.g. forgetting to add some ingredients while cooking) and the quality of accomplishment will also be affected. In some extreme cases, it will lead serious or fatal consequences (e.g. forgetting to turn off the oven after use).

As we analyzed above, the optional actions in set O are less important than the ones in set E, and bring less trouble while being omitted. Due to the set-based dual structure of concepts, it is easy to check the final completion of implementation using set theory: if the universal actions of an activity A_i is denoted as U_i, the forgotten actions could be calculated as the relative complement $S^C = U_i - S$, where S is current sequence of observed and finished actions. It is worthy to mention that U_i could be quickly obtained by executing the concept-forming operation A_i' or searching the cross table.

Example: suppose that the actions in sequence $\alpha = \{a \prec c \prec b \prec f\}$ is successively loaded. Considering Figure 2, node 7 is located at the end of the extensions. To check the completion of g_4 indicating in the extent, we compare current sequence $\alpha = \{acbf\}$ with $g_4' = \{abcdf\}$, and the complement $g_4' - \alpha = \{d\}$ is not an empty set, so action d is omitted during the execution of g_4.

4.2.3 Unreasonable repetition

The reason of redundant information existing in the data stream could be various: periodic sampling of sensors, reasonable intention or anomaly etc. In our issue, the redundant information should be the repetitive actions existing in a sequence of actions. In most cases, repetitive actions are harmless, even reasonable and necessary to accomplish an activity. For example, we need to regularly check the degree of cooking or intermittently stir the ingredients while preparing a meal. In the other extreme cases, unreasonable repetitive actions will lead to potential threats like excessive consumption (condiments or medications).

The simplest solution is to check whether the incoming action exists in the current sequence of actions α. To distinguish the unreasonable repetition caused by cognitive deficits with the reasonable ones, we define a weighted array to measure the harm degree of each action being repetitive. For this reason, the detection accuracy of harmful redundancy could be reinforced and the false-positive alert warning harmless redundancy could be reduced.

4.2.4 Mixture of irrelevant actions

People with cognitive impairment often forget current planning or confuse with another one, and then add irrelevant actions into the current ongoing activity. From Equation 3, we could see that irrelevant actions set I of activity A_i has no intersection with the relevant one $E \cup O$. In other words, an extension caused by an incoming action a is acceptable for current planning if and only if $a \in E \cup O$. Thus, full elements in I will be excluded by all the concepts containing A_i.

After a new extension, if updated α is no longer compatible with any concept except the Infimum, there are probably one or more actions are irrelevant which have mixed into the current sequence, especially the last incoming one should be suspected.

Example: suppose sequence α is successively extended by $\{a \prec c \prec e \prec d \prec b \prec f\}$. Node 6 is located after the first two extensions $\alpha \leftarrow ac$. In the third round, $\alpha \leftarrow e$, updated $\alpha = \{ace\}$ is incompatible with current planning because there is no subconcept (A, B) having $\alpha \subseteq B$ except the Infimum. As a consequence, last incoming e will be treated as an irrelevant action which have to be removed from the initial cache and put it aside, into a new created cache indicating another planning. At the end of the extensions, node 7 is located and the irrelevant action e is identified.

We summarize the logic above and represent it as pseudocode in Algorithm 1. Cache P_0 always denotes the initial planning of a resident. A new incoming action a is observed and loaded for extension at step 3. Step 4 to 7 is to check whether there exists one or more caches in P_i are compatible with current observed actions. If incoming action a is irrelevant to all existing caches (step 9), then create a new cache to save it (step 10 to 11). After extensions, we choose the longest cache, P_0 in most of time, as the normal sequence of actions performing A_i (step 12), and the actions in the other caches will be treated as irrelevant actions.

Algorithm 1: detect mixture of irrelevant actions

Data: sequence α, lattice $\mathcal{L}$, caches P_i.
Result: set of irrelevant actions $\mathcal{I}$.

```
1  begin
2      while α do
3          a ← α.popleft
4          foreach P_i do
5              if ∃(A, B) ∈ L, P_i ∪ a ⊆ B then
6                  P_i ← P_i ∪ a
7          end
8      end
9      if ∄(A, B) ∈ L, P_i ∪ a ⊆ B then
10         P_{i+1} ← a
11         P_i ← P_i + P_{i+1}
12     P_M ← max(size(P_i))
13 end
```

4.2.5 Causal conflict

Suppose two actions, $\alpha_i \prec \alpha_{i+m}$, appear successively in the sequence $\alpha = \{\alpha_0 \prec ... \prec \alpha_i \prec ... \prec \alpha_{i+m} \prec ... \prec \alpha_n\}$. If causal constraints C has limited that α_{i+m} must occur before α_i, represented as $\alpha_{i+m} \preceq \alpha_i$, then there is a causal conflict in the sequence.

In this paper, we manually define causal constraints and then verify the causalities among actions in α. For any action α_i in the sequence, we generate its causal pairs by scanning all the actions on its right. If one generated pair (α_i, α_j) has the opposite one (α_j, α_i) in C and no α_j appeared before α_i, then the sequential execution $\alpha_i \prec \alpha_j$ is against the predefined causality. The time complexity of causal conflict check is $T(O(n^2))$.

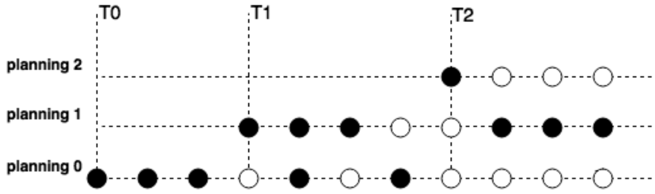

Figure 3: cognitive distraction.

4.2.6 Cognitive distraction

Cognitive distraction is similar to adding irrelevant actions. Compared to original planning, the two errors have the same feature that they are mixed irrelevant actions into their sequences of actions, but cognitive distraction has created a transformation of quantitative into qualitative changes. Different from the mixture of irrelevant actions, this error could be classified as a collective anomaly [12]. The feature of cognitive distraction is that at the beginning of the sequence of actions, all the performed actions belong to a real expected long-term planning. At a specific singular point, the performed actions started to distract from the original objective.

Figure 3 is an example of cognitive distraction. Planning 0 is used to indicate the original planning of a resident and Planning 1 and 2 denote his/her distracted traces. A Black point represents a hit that the loaded action used for extension in this step is accepted by the positioned cache and the Hasse diagram, and a white one indicates a missing.

The cognitive distraction really happens in the fourth extension and T_1 indicates this singular position. The loaded action a_4 has not been accepted by the Planning 1 due to its irrelevance. Once an action is not acceptable for all existing caches, we need to put it in a new one. There is only one black point at the moment of new cache creation. Moreover, if an action is compatible with more than one cache, it must be distributed into each compatible cache. At the end of the extensions, we choose the longest cache having the most compatible actions as the normal sequence of actions. If the longest cache is not Planning 0, we could assert that the resident has distracted from his/her real objective.

5. EXPERIMENTS

In this section, we first introduce the characteristics of experimental data sets, and then evaluate the performance of detecting predefined cognitive errors.

5.1 Data sets

Our experiments were carried out on two data sets created by LIARA laboratory: real data set *RDATA* consisting of seven activities captured by the sensor network of LIARA's smart home, and synthetic data set *EDATA* containing predefined cognitive errors.

Two benchmark data sets, *ADLActivities* and *Activities WithErrors* (hereafter referred as *CASAS* and *ECASAS*) [6], were also used to evaluate the performance of our inference engine. Both of them describe the same ADLs, but the latter contains errors in its data stream. So we use the former data set to train our inference engine in order to detect the errors in the latter. The use of the benchmark data sets mentioned above is an attempt because their records are low-level sensor data streams, not the higher-level sequences of actions.

Considering complicated scenarios and more interactions between activities, we choose kitchen activities as our main research objects. Statistical information about adopted activities named in camel case is shown in Table 1.

Table 1: Statistical information of kitchen activities

activities	No. component actions
PrepareCoffee	14
PrepareCoffeeWithoutSugar	11
PrepareCoffeeWithoutMilk	11
PrepareSpaghetti	18
PrepareSandwich	15
PrepareCereal	8
PreparingToastsAndEggs	20

Raw records in *RDATA* contain four fields: temporal information, spatial-topological information [18], identified actions and corresponding activities. To simplify the problem, we refined the records and only kept the information about identified actions and corresponding activities. We extracted the same fields from *ECASAS*.

To avoid underfitting, for each indexed activity in *RDATA*, we kept the set of actions unchanged, but disrupted the internal execution orders under the condition of following the causal constraints among actions. Thus, we obtained sufficient derived sequences of actions that could be used for training models or generating test cases with cognitive errors. On the basis of these derived sequences, we randomly changed their inner structures (e.g. removing, adding, repeating, splicing and swapping actions) to create the test data set *EDATA* with mentioned cognitive errors.

5.2 Evaluation

In our experimentation, we carried out all the evaluations on one laptop with tech specs of Intel Core i7 Processor 2.4GHz and 8GB RAM, under Ubuntu 14.04.

Table 2 sketches the accuracies about cognitive errors detection applied on the two test data sets. 3-fold cross-validation was also used to avoid overfitting.

From the listed results in Table 2, we could see that our engine received excellent detection rates in four cognitive errors except the cognitive distraction. One of the reasons is that the detection accuracy of cognitive distraction error depends on the singular position when the distraction occurs. Figure 4 shows the F-measure at different singular positions. The precision at each position is always equal to 1 (TP=1.0 and FN=0.0). It is worth mentioning that the result of causal conflict detection was based on the manually defined causal constraints (marked as “M”).

For *CASAS*, there are only two predefined errors existing in the test samples: omission (did not turn the water off, did not turn the burner off, did not bring the medicine container, did not use water to clean and did not dial a phone number) and repetition (dialed a wrong phone number and redialed, duplicate sampling of motion sensors, etc.). We used “-” to represent the nonexistent results in Table 2. Furthermore, we evaluated its results under evaluation metrics, including precision, recall and F-measure in Table 3.

5.3 Discussion

Table 2: cognitive error detections in EDATA

cognitive errors	data sets/ACC	
	EDATA	CASAS
omission of essential actions	100%	88.5%
mixture of irrelevant actions	100%	-
unreasonable repetition	100%	100%
causal conflict	100% (M)	-
cognitive distraction	$\geq$ 97.8%	-

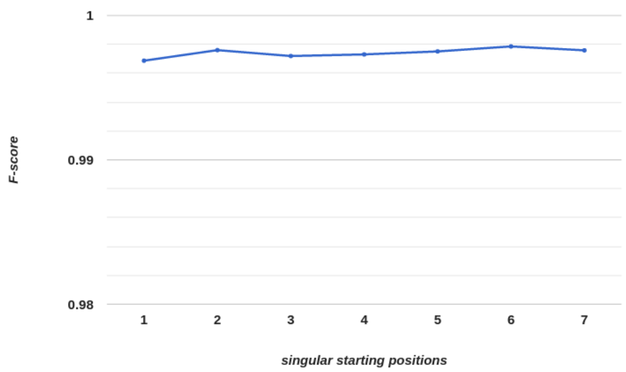

Figure 4: cognitive distraction detection in EDATA at different singular positions

With reference to the architecture sketched in Figure 5, in this section, we provide a detailed overview of our inference engine and summarize the advantages and drawbacks in terms of cognitive errors detection.

First of all, our smart home collected low-level sensor data from a sensor network in a period of time. With the help of our previous work [7][18], human-unreadable sensor data have been successfully transformed into atomic actions cached in continuous sequences of actions.

Then, inside the inference engine, activity recognition agent was trained by historical behavioral data. Interrelations between different activities, referred as sets of shared actions, have been hierarchically represented as nodes in a Hasse diagram. In this way, activity recognition could be formulated as a graph matching problem. Moreover, a RMSD-based (root-mean-square deviation) assessment was used to assess and select the most probable activity among several possible candidates during the recognition process [11].

Next, after the features analysis of common abnormal behavioral patterns, combined the advantage of FCA in discrete mathematics, we gave out different solutions for detecting predefined cognitive errors.

Omission of essential actions and unreasonable repetition are two cognitive errors strongly related to the set theory of discrete mathematics. Through simple algebra of sets and binary operations on sets, they could be easily detected. As shown in Table 2, repetitive actions in the sequence were 100% detected, but not all of them are unreasonable. For example, in CASAS, due to the deployment of motion sensors and periodic sampling, sequences are filled with repetitive events. The existence of motion sensors in CASAS also affects the result of the omission error detection. Irregular movements of residents produce massive derivative sets

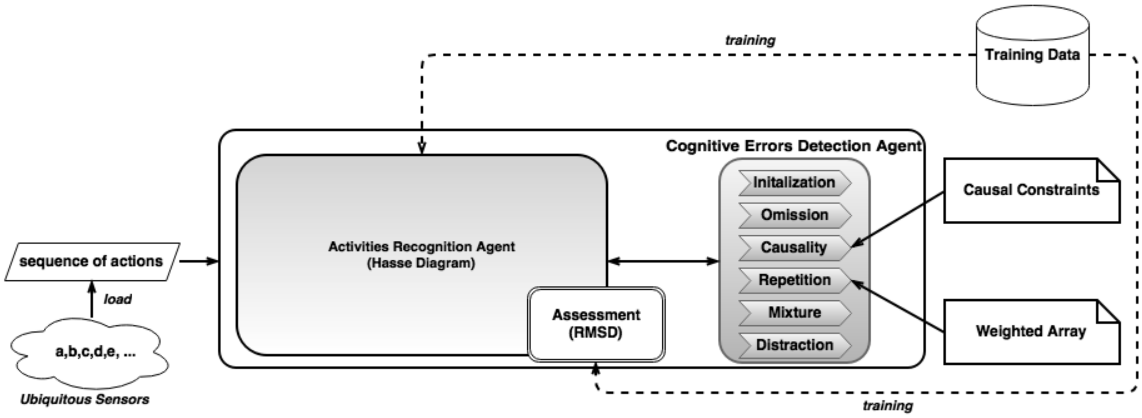

Figure 5: architecture of FCA-based inference engine

Table 3: cognitive errors detection in CASAS

cognitive errors	precision	recall	F-score
omission	0.656250	1.0	0.792453
repetition	1.0	1.0	1.0

of actions having negligible movements as elements of the optional actions set O. Thus, the repetition and omission existing in the sequence of sensor data will lead high false-positive rate (12.3%).

As we mentioned in Section 4.2.3, some of the repetitive actions are also necessary to ensure the quality of activity implementation. In order to reduce the false-positive rate and to increase the true-positive rate at the same time, it is worthy to note that a weighted array was defined for the unreasonable repetition error to automatically adjust the detection sensitivity on the basis of the severity of each repetitive action. This strategy will also be imported to omission detection in the future work.

To detect causal conflicts in a sequence, compared to simple binary operations on set, the biggest challenge to overcome is the source of causal constraints. As the result shown in Table 2, causal constraints defined by human expert are accurate and easy to be deployed into conflict detection, but the definition was also prohibitively expensive.

The rest two cognitive errors, mixture of irrelevant actions and cognitive distraction, are more complex than the others because of the ambiguous singular position between original intention and the abnormal one. Multilevel inheritance and varied singular positions also aggravate the complexity of situations. In the worst case, some samples with distraction errors will be identified as a series of repetition errors in this case.

With our new semi-supervised approach, we do not need to consider imbalanced class distribution. Only normal classes corresponding to normal behavior could be used for building a model to identify anomalies in the test data.

6. CONCLUSION

Due to the deterioration in cognitive skills, people with cognitive impairment frequently produce cognitive errors when they carry out activities of daily living. In this paper, we formulated the most common cognitive errors existing among people with cognitive impairment. Under the unified hierarchical FCA-based inference engine, we proposed a series of customer-built detectors to detect predefined cognitive errors in the sequences of actions. Moreover, we also defined several dynamic mechanisms to reduce the false-positive rate according to predefined weights. Unlike the other similarity or frequency-based approaches, our frameworks do not require the fault samples should be available in advance.

However, our approach has severe constraints on the training data. To ensure the correctness and completeness of the constructed Hasse diagram model, training data are required to cover diverse sequences of actions containing varied sets of actions executing the same activities as many as possible. Insufficient training sample will cause a high false alarm rate in detecting omission of essential actions (i.e. wrongly identify the subset derivative sequences) and mixture of irrelevant actions (i.e. wrongly identify the superset derivative sequences).

Our approach also need to ameliorate the detection performance by considering more rigorous logic to solve the multilevel inheritance problem. As a potential solution, in our future work, we will integrate the knowledge about ontology to extract the common part of component actions among the activities with multilevel inheritance and furthermore improve accuracy in the imperfect parts. To reduce prohibitively expensive work of causal constraint definition, we will to design a heuristic method to generate those causal rules automatically. Furthermore, we also would like to test our approach in more complicated scenarios like multitasking and interleaving modesd.

7. ACKNOWLEDGMENTS

The authors would like to thank their main financial sponsors: the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Canadian Foundation for Innovation (CFI).

8. REFERENCES

[1] National Institute on Aging. *Alzheimer's Disease Fact Sheet.* NIH Publication, 2015.

[2] National Institute on Aging. *Talking with Patients about Cognitive Problems.* NIH Publication, 2011.

[3] P. Barberger-Gateau and C. Fabrigoule. Disability and cognitive impairment in the elderly. *Disability and rehabilitation*, 19(5):175–193, 1997.

[4] Audrey Serna, Hélène Pigot, and Vincent Rialle. Modeling the progression of alzheimer's disease for cognitive assistance in smart homes. *User Modeling and User-Adapted Interaction*, 17(4):415–438, 2007.

[5] P.C. Roy, S. Giroux, B. Bouchard, A. Bouzouane, C. Phua, A. Tolstikov, and J. Biswas. A possibilistic approach for activity recognition in smart homes for cognitive assistance to alzheimer's patients. In *Activity Recognition in Pervasive Intelligent Environments*, pages 33–58. Springer, 2011.

[6] D.J Cook and M. Schmitter-Edgecombe. Assessing the quality of activities in a smart environment. *Methods of information in medicine*, 48(5):480, 2009.

[7] C. Belley, S. Gaboury, B. Bouchard, and A. Bouzouane. Nonintrusive system for assistance and guidance in smart homes based on electrical devices identification. *Expert Systems with Applications*, 42(19):6552–6577, 2015.

[8] J.K. Aggarwal and M.S. Ryoo. Human activity analysis: A review. *ACM Computing Surveys (CSUR)*, 43(3):16, 2011.

[9] M. Gupta, J. Gao, C. Aggarwal, and J. Han. Outlier detection for temporal data. *Data Mining and Knowledge Discovery*, 5(1):1–129, 2014.

[10] T. Duong, H. Bui, D. Phung, and S. Venkatesh. Activity recognition and abnormality detection with switching hidden semi-markov model. In *CVPR'05*.

[11] J. Hao, S. Gaboury, and B. Bouchard. Real-time activity recognition and prediction in smart home by formal concept analysis. 2016.

[12] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM computing surveys*, 2009.

[13] Heng-Tze Cheng. Learning and recognizing the hierarchical and sequential structure of human activities. 2013.

[14] K. Park, Y. Lin, V. Metsis, Z. Le, and F. Makedon. Abnormal human behavioral pattern detection in assisted living environments. In *PETRA'10*, 2010.

[15] T. Zhao, H. Ni, X. Zhou, L. Qiang, D. Zhang, and Z. Yu. Detecting abnormal patterns of daily activities for the elderly living alone. In *HIS*, pages 95–108. Springer, 2014.

[16] N. El-Kechaï and C. Després. A plan recognition process, based on a task model, for detecting learner's erroneous actions. In *Intelligent tutoring systems*, pages 329–338. Springer, 2006.

[17] J. Yin, Q. Yang, and J. Pan. Sensor-based abnormal human-activity detection. *Knowledge and Data Engineering*, 20(8):1082–1090, 2008.

[18] D. Fortin-Simard, J. Bilodeau, S. Gaboury, B. Bouchard, and A. Bouzouane. Human activity recognition in smart homes: Combining passive rfid and load signatures of electrical devices. In *Intelligent Agents*, pages 22–29. IEEE, 2014.

[19] D. Fortin-Simard, J. Bilodeau, S. Gaboury, B. Bouchard, and A. Bouzouane. Method of recognition and assistance combining passive rfid and electrical load analysis that handles cognitive errors. *International Journal of Distributed Sensor Networks*, 501:643273, 2015.

[20] B. Ganter and R. Wille. *Formal concept analysis: mathematical foundations*. Springer Science, 2012.