International Journal of Medical Informatics 112 (2018) 173–184

Contents lists available at ScienceDirect

International Journal of Medical Informatics

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

A fuzzy logic control in adjustable autonomy of a multi-agent system for an automated elderly movement monitoring application

Salama A. Mostafa[a,*], Aida Mustapha[a], Mazin Abed Mohammed[b], Mohd Sharifuddin Ahmad[c], Moamin A. Mahmoud[c]

[a] Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Johor, Malaysia
[b] Planning and Follow Up Department, University Headquarter, University of Anbar, Anbar, Iraq
[c] College of Computer Science and Information Technology, Universiti Tenaga Nasional, Selangor, Malaysia

ARTICLE INFO

Keywords:
Autonomous agent
Multi-agent system
Adjustable autonomy
Fuzzy logic
Elderly remote care

ABSTRACT

Autonomous agents are being widely used in many systems, such as ambient assisted-living systems, to perform tasks on behalf of humans. However, these systems usually operate in complex environments that entail uncertain, highly dynamic, or irregular workload. In such environments, autonomous agents tend to make decisions that lead to undesirable outcomes. In this paper, we propose a fuzzy-logic-based adjustable autonomy (FLAA) model to manage the autonomy of multi-agent systems that are operating in complex environments. This model aims to facilitate the autonomy management of agents and help them make competent autonomous decisions. The FLAA model employs fuzzy logic to quantitatively measure and distribute autonomy among several agents based on their performance. We implement and test this model in the Automated Elderly Movements Monitoring (AEMM-Care) system, which uses agents to monitor the daily movement activities of elderly users and perform fall detection and prevention tasks in a complex environment. The test results show that the FLAA model improves the accuracy and performance of these agents in detecting and preventing falls.

1. Introduction

Many studies have reported a great increase in the median age of the humans, especially those who are living in developed economies [1,2]. For instance, 17.4% of the population in the European Union have been classified as elderly (aged 65 years and above) in 2010, and this ratio is expected to reach 28.8% in 2050 [3]. As the world population continues to age, the number of elderly who are living independently or are left on their own at daytime has also been rising [4]. Some elderly people are facing health problems and require medical attention. To address the special needs of this population, many elderly remote care systems have been proposed, and some examples of these systems are presented in [5]. The elderly tend to show trembling, rigidity, and sluggishness in their movement which expose them to the risks of falling. This problem affects their ability to live independently, reduces their quality of life and can be hazardous and fatal [1,6]. Several elderly remote care systems have been proposed in the attempt to solve or mitigate this problem, and some examples of these systems are presented in [2]. These systems monitor the elderly's movement activities and daily routine patterns to prevent and detect fall situations.

Autonomous agents and multi-agent technologies have significant roles and contributions in many healthcare and elderly remote care systems [7]. For example, Kaluža et al. [3] propose an agent-based elderly remote care system that supports the independent living of the elderly. The system monitors their movement activities and automatically notifies medical personnel in case of a fall. Typically, agents in discrete and deterministic environments autonomously complete a substantial amount of tasks due to their prior knowledge about their surroundings. However, agents in complex environments that have the characteristics of uncertain, highly dynamic, or irregular workload tend to make decisions that lead to unwanted consequences [8]. These agents are deployed to handle some primitive, deducible, or critical tasks [9]. To make appropriate decisions amid such problems, agents must operate at different autonomy levels and with different autonomy properties [10,11,12]. In response to this need, several studies have attempted to model adjustable autonomy, which enables agents to operate at different autonomy levels. Some of these works have been reviewed in [8]. The adjustable autonomy in a multi-agent system is managed by grading the modifiable autonomy of agents within a specified range [11]. The grading process involves measuring the

* Corresponding author.
E-mail addresses: salama@uthm.edu.my (S.A. Mostafa), aidam@uthm.edu.my (A. Mustapha), mazin_top_86@yahoo.com (M.A. Mohammed), sharif@uniten.edu.my (M.S. Ahmad), moamin@uniten.edu.my (M.A. Mahmoud).

https://doi.org/10.1016/j.ijmedinf.2018.02.001
Received 14 November 2017; Received in revised form 7 January 2018; Accepted 1 February 2018
1386-5056/

boundaries of autonomy parameters and estimating the extent to which autonomy is distributed and adjusted among the agents [13,14].

In this paper, we propose the fuzzy-logic-based adjustable autonomy (FLAA) model to manage the autonomy of agents in multi-agent systems that operate in complex environments. This model employs fuzzy logic to linearly measure the autonomy of the agents. We implement and test this model in the automated elderly movements monitoring (AEMM-Care) system. The system observes the movement activities of the elderly to detect or prevent a fall. Given the intricate nature of fall detection and prevention tasks, the ambiguity of sensory data, and other challenges related to the application domain, we prove that the FLAA model improves the performance of the agents and the accuracy of the AEMM-Care system.

The rest of this paper is organized in the following order. Section 2 reviews the literature on agent-based elderly remote care systems, cites some previous attempts in applying fuzzy logic to manage the autonomy of agents and describes the needs and necessities of using adjustable autonomy. Section 3 illustrates the proposed FLAA model for multi-agent systems which includes the representation, measurement, distribution, and adjustment of autonomy. Section 4 discusses the prototype design, implementation, and application of the adjustable FLAA model in the AEMM-Care system. Section 5 presents the AEMM-care system test results and discusses the research outcomes. Section 6 concludes the paper and offers recommendations for future work.

2. Literature review

Our literature review reveals that the agent-based healthcare systems being used today are not using fuzzy logic to control the autonomous behavior of agents. Previous studies have also largely ignored the possible application of this technique in managing adjustable autonomy in multi-agent systems. To address these gaps, our literature review focuses on three topics of multi-agent elderly remote care systems, the use of fuzzy logic to control agents' autonomy, and a brief study of adjustable autonomy.

2.1. Multi-agent elderly remote care systems

Recent studies on healthcare or remote care have extensively explored the application of multi-agent systems in the medical field. Some of these studies have proposed and examined the use of autonomous agents to assist humans in their daily activities [5]. For example, Armentia et al. [15] propose a multi-agent elderly remote care system with a component-based architecture. The agents have adaptability and replication mechanisms. The adaptability mechanism helps agents proactively respond to critical situations involving the elderly based on the relationships among certain events. The replication mechanism provides these agents with alternative options in case of failures, emergencies, or resource constraints. The test results highlight the importance of applying advanced autonomous solutions in elderly remote care and general healthcare systems.

Monitoring the falling risks of the elderly presents a major challenge in healthcare research [1]. Those elderly with poor balance, weak legs, and other impairments in their mobility are highly susceptible to falling situations that may lead to severe or life-threatening injuries [2]. These risks underscore the need to develop a flexible, reliable and efficient system that monitors the movement activities of the elderly and helps them avoid perilous situations [16]. However, the monitoring systems that are currently available in the market have been criticized for their high cost, discomfort, or imprecision. Nevertheless, the fall detection methods proposed in the literature are generally accurate due to the simplicity of and similarities in falling patterns (i.e., most falls begin with a person losing his/her balance and ends with the same person lying on the ground). Some of these methods have even reported fall detection accuracies of as high as 90% [1]. By contrast, fall prevention is a complex process which success hinges on probabilistic and predictive methods. Given that an individuals' susceptibility to falling can be driven by physical, behavioral, and psychological factors. Previous studies have closely examined the daily movement activities and patterns of humans to successfully detect or prevent a fall. These movement activities are captured by employing various types of wearable, visual, and ambient devices [2,4]. Several techniques, such as autonomous agents and machine learning algorithms, have also been applied to tracking movement activities and identifying falling situations.

Kaluža et al. [3] propose a context-aware agent model for an elderly fall detection and prevention system that observes the movement activities and assesses the ability of the elderly to live independently. This system includes four groups of agents that are assigned to specific tasks. The first group gathers information on the movement activities of the elderly, the second group contextualizes the gathered information, the third group evaluates the movement ability of the elderly and alerts them when a risk is present, and the fourth group checks for any irregularities in the daily routine and behavior of the elderly. Although this system can successfully distinguish a movement-impaired individual from a healthy one, its four groups of agents are not using the same algorithm, thereby leading to unsatisfactory outcomes. Moreover, the differences in the autonomy and learning abilities of these agents can drive them into making inconsistent decisions that expose the system to disturbances and unnecessary procedures (e.g., merging agents to address the conflicts in their decisions) that only extend the overall computational time.

Cvetković et al. [6] propose a multi-classifier adaptive-training (MCAT) model that employs support vector machine, decision tree, and random forest classifiers to improve the movement recognition accuracy of elderly fall detection and prevention systems. This model operates in passive and active modes. The passive mode configures the confidence and adaptation parameters of the classifiers, while the active mode runs these classifiers by applying a semi-supervised learning method. The test results reveal that the MCAT model performs better in the active mode than in the passive mode only if the confidence and adaptation parameters are appropriately configured. The MCAT model achieves an average accuracy score of 82.70%, which exceeds the accuracy score of other self-training algorithms by 10.79%. However, this model only classifies the patterns of movement activities and ignores those of other activities.

Lustrek et al. [1] develop a Ubisense system that uses locational sensors and accelerometers to collect data from an individual. They also propose a confidence model for fall detection that uses a random forest classifier for recognizing movement activities and a hidden Markov algorithm for detecting and ruling out infeasible activity patterns. The test results show that with a single accelerometer, the proposed confidence model achieves a fall detection accuracy of 79.2%, while with three accelerometers, this model achieves an average fall detection accuracy of up to 97.2%. Examining movement activity patterns can help one contextualize a falling incident and improve the fall detection accuracy of a system. However, Lustrek et al. only focus on improving the fall detection accuracy of the system and completely ignore its performance in fall prevention.

2.2. Fuzzy logic control of agents' autonomy

Fuzzy logic is a widely used technique for its features of practicality, durability, computational efficiency, certainty, and easy integration with other techniques and applications [17]. It consists of fuzzy sets and a fuzzy inference in which a local model of the system under consideration is represented by fuzzy rules [18,19]. Fuzzy logic provides a decision-making mechanism that has many uses and benefits. It can (1) deal with uncertain and vague situations [17], (2) perform mathematical analysis and approximation for linear, nonlinear, or dynamic problems [9], and (3) construct inference models for solving and controlling problems [20]. Many studies have applied fuzzy logic to

Table 1
Terminologies of autonomy representation.

term	symbol	context	relationship
level	l	a band of autonomy degrees for operational behaviors	an operational autonomy of a system
degree	∂	a measurable unit that defines the autonomy states of an agent in a cycle of actions	an operator autonomy of agents
property	p	a particular state of an autonomy degree	a characteristic of an operator autonomy
task	t	a representation for a specific collection of actions to be carried out by an agent	an operational compound of a system
action	a	a sequence of operational activities	an operational element of a system

investigate the decisions of agents including explicitly control their autonomy [18], improve their perceptions toward their environments [19], or limit the number of decision options that are made available to them [20].

Ho-Sub et al. [19] propose a novel fuzzy logic technique that helps agents understand and interpret unclear goals from uncertain environments. This technique applies a set of linguistic representations to an agent's set of fuzzy goals and then uses a fuzzy reinforcement function that processes such goals during the agent's decision-making process.

Jaafar and McKenzie [9] propose a fuzzy logic technique that controls the autonomous behavior of agents. Specifically, this technique arranges the behavior of agents in weighted levels and then manipulates these weights to defuzzify and determine appropriate behaviors for the agents. Through this defuzzification procedure, the number of decision options that are available to agents is reduced to a certain number of possible actions. This technique can guide the agents when performing specific actions in unknown and complex environments.

Couceiro et al. [18] propose a fuzzy-logic-based algorithm for managing the autonomy of agents. This algorithm initially applies context-based evaluation metrics to describe the performance of agents and then fine-tunes the autonomy constraints based on the described performance. Multi-robot systems have adopted this technique to coordinate the performance of robots in complex environments and to help them achieve their goals and overcome obstacles even with limited coordination.

2.3. Adjustable autonomy

The autonomy of agents has been extensively studied by using qualitative and quantitative approaches [21]. Qualitative approaches differentiate various levels of autonomy by comparing the linguistic variables of quality criteria, while quantitative approaches numerically express such autonomy to form an advance autonomy measurement [10]. These approaches contribute to the management of adjustable autonomy by extending such autonomy within fully autonomous and non-autonomous limits [12,13]. Subsequently, adjustable autonomy grants agents a variable range of autonomy levels to act upon [11]. These levels give the options of agents working independently, dependently or being prone to human oversight or intervention [22]. The autonomy adjustment changes some agents' activities based on a situation of exigency, to influence the agents to make desirable decisions [10]. The components of adjustable autonomy and its application to multi-agent systems have been construed in many ways to provide the benefits of [8]:

- Flexibility: Adjustable autonomy facilitates teamwork between human users and agents in controlling a system and confronting diverse situations. These parties can function at different autonomy levels when operating the system.
- Reliability: Adjustable autonomy helps human users maintain their global control over the agents and adjust the autonomy of these agents.
- Efficiency: Adjustable autonomy conducts assessments that motivate the agents to improve their autonomy. The assessment ensures the advancement of a competent agent over a less competent one because of its highly exploitable abilities.

3. The fuzzy logic-based adjustable autonomy model

This section explains how autonomy is represented, measured, distributed, and adjusted in the FLAA model. It also explains the architecture and operational behaviors of the agents that work according to the FLAA model.

3.1. Autonomy representation

Representation is an important feature that must be considered when quantitatively or qualitatively measuring autonomy. In the FLAA model, the autonomy of an agent is represented through several levels of fuzzy boundaries that hierarchically manage the autonomy of a multi-agent system. An agent may either function in a single autonomy level or switch from one level to another. Each autonomy level imposes some constraints on the behaviors of agents. However, those agents with high autonomy have less constrained behaviors, thereby allowing them to perform a wide range of tasks and actions.

An autonomy level, l, represents a band of autonomy degrees that follows a set or sets of autonomy properties, P, in which a property $p \in P$. P includes a range of fully autonomous to non-autonomous abilities that check for possibilities to allow, block, influence, mediate, increase, suspend, and/or terminate the agents' behaviors or goals. The P of an l is configured based on the relationship between tasks and actions. Table 1 presents the terminologies used in autonomy representation.

Fig. 1 shows an abstract representation of adjustable autonomy. Each l is associated to sets of possible tasks and actions (e.g., $l_1 = \{t_1 = \{a_1, a_2, ...\}, \ t_2 = \{a_1, a_2, ...\},\{\}\}$) and may have overlapping and distinct P, t, and/or a. We assume that the setting and distribution of these elements adhere to the FLAA application domain.

Before representing the autonomy of a multi-agent system, we first define the term "agent." an agent is a computer software that is capable of carrying out tasks and executing actions in a flexible and autonomous manner [23]. The operations of these agents are largely defined by their architectures [14]. In this study, we use a BDI architecture that includes three main behaviors, namely, *tasks-selection* (where an agent examines and selects the proper tasks), *actions-selection* (where the agent selects proper actions to perform these tasks), and *actions-execution* (where the agent performs the selected actions) [24]. The agent's autonomy is distributed and can be independently adjusted within these behaviors.

Let ρ_k represent an agent operating in a certain environment, E, that is indexed by k, in which $k = \{1, 2, \ldots\}$. A represents the set of all actions that can be performed by ρ_k to complete its tasks. a denotes an action, in which $a \in \mathrm{A}$. T represents all possible tasks, in which $\mathrm{T} = \{t_1, t_2, \ldots\}$ and a task $t \subseteq \mathrm{A}$. ρ_k observes an event e_x in E and then acts on this event by implementing *tasks-selection*, ρ_k^1, *actions-selection*, ρ_k^2, and *actions-execution*, ρ_k^3. Algorithm 1 shows the basic architecture of an agent.

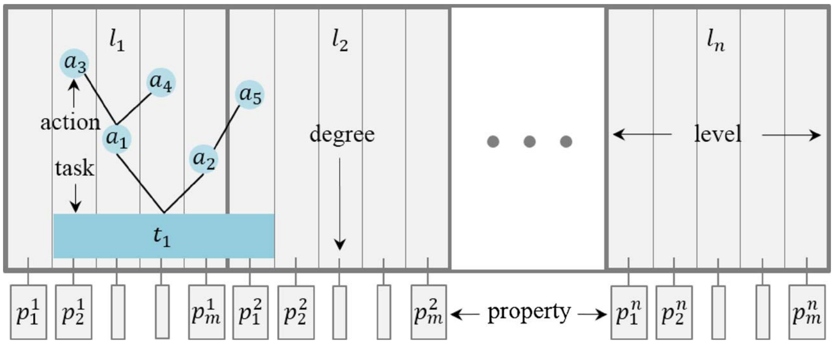

Fig. 1. The adjustable autonomy of the FLAA model.

Algorithm 1: The basic architecture of an agent

$e_x \leftarrow$ *observe* (ρ_k, E);
//ρ_k^1: *tasks-selection*
$t_i \leftarrow$ *decide* (ρ_k, e_x);
//ρ_k^2: *actions-selection*
$a_{i,\{1,2,\ldots\}} \leftarrow$ *decide* (ρ_k, t_i);
//ρ_k^3: *actions-execution*
$E' \leftarrow$ *execute* $(\rho_k, a_{i,\{1,2,\ldots\}})$;

ρ_k decides on a task or action based on the operational autonomy constraints of the available tasks or actions:

$$decide_{\varphi}(\rho_k, x) = \{x_i\colon x_i \in X, \eta x_1(c_1) \Upsilon \eta x_2(c_2) \Upsilon \ldots\}, \tag{1}$$

where *decide* denotes the selection operation to x relation, $\forall\, x_i \in$ X, x denotes either the t or a of a selection operation, φ denotes the selection decision condition, c_1, and c_2 denote the properties of the selection decision condition, Υ denotes the logical operator that connects each condition and is restricted by [$\wedge$, $\vee$] operators, and η denotes the negation operator ($\neg$) that is either existing or non-existing.

For example, if task t has three actions, namely, a_1, a_2, and a_3, and an agent decides on actions for t, then the decision can be cognitively represented as $a_1(p_1) \wedge \neg\, a_2(p_2) \wedge a_3(p_3)$, where $p_{1,2,3}$ denotes the properties of different autonomy levels. The case of l: $p_1 \rightarrow \dot{a}$ denotes an action with a fully autonomous property, l: $p_2 \rightarrow \tilde{a}$ denotes an action with a semi-autonomous property, and l: $p_3 \rightarrow \bar{a}$ denotes an action with a non-autonomous property.

3.2. Autonomy measurement

We first define the measurable attributes of autonomy before attempting to measure this concept. We consider knowledge (*know*) and authority (*can*) as the measurable attributes of autonomy because of their close relationship with the agents' autonomy [14]. These attributes set the boundaries for the autonomy of agents that are operating in a certain environment [8]. The values of these attributes are used to assess the qualifications of an agent to reach a particular autonomy level. *Know* is deduced from the level of agreement in the decisions made by agents, while *can* is deduced from the external responses (assessment) to the actions performed by these agents. In other words, the know condition represents the agents' autonomy based on internal desire, while the can condition represents their autonomy based on external desire; and these conditions altogether satisfy the concept of adjustable autonomy [11]. We use fuzzy logic to measure the operational autonomy of a multi-agent system, to assess the autonomy degree of its agents, and to assign these agents to proper autonomy levels.

Let ω denote a set of active agents in a system, in which an agent $\rho_k \in \omega$. This system has three autonomy levels, $L = \{l_1, l_2, l_3\}$. Let *know* and *can* denote two fuzzy sets of linguistic input variables that have *High*, *Medium*, and *Low* qualitative values, in which $know = \{knowL, knowM, knowH\}$ and $can = \{canL, canM, canH\}$. The membership in high, medium and low levels of autonomy is represented by the trapezoidal right, middle, and left functions in the following equations [17]:

$$\mu_f(x; b_1, b_2, b_3, b_4) = \begin{cases} ,0 & x \le b_1 \\ \frac{x-b_1}{b_2-b_1}, & b_1 \le x \le b_2 \\ 1, & b_2 \le x \le b_3, \quad x \in \mathrm{R}. \\ \frac{b_4-x}{\mathrm{d}-b_3}, & b_3 \le x \le b_4 \\ 0 & b_4 \le x \end{cases} \tag{2}$$

Following Fig. 1 and Eq. (2), we construct Fig. 2 which shows the fuzzy logic-based adjustable autonomy. The figure also shows the range of input *know* and *can* variables and the degree of membership in each output autonomy level, respectively. As we have explained earlier, each autonomy degree corresponds to a specific set of autonomy properties.

We compute for the $know_{\mu}$ membership degree by measuring the extent of agreement in the decisions of agents, while we compute for

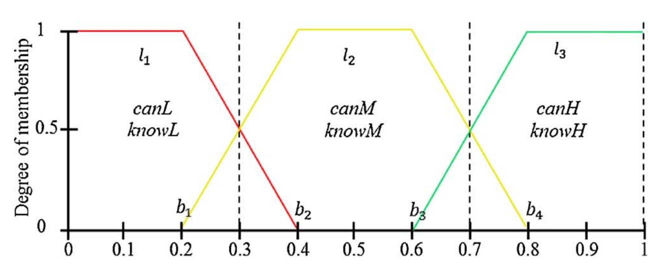

Fig. 2. Membership of autonomy levels in the FLAA model.

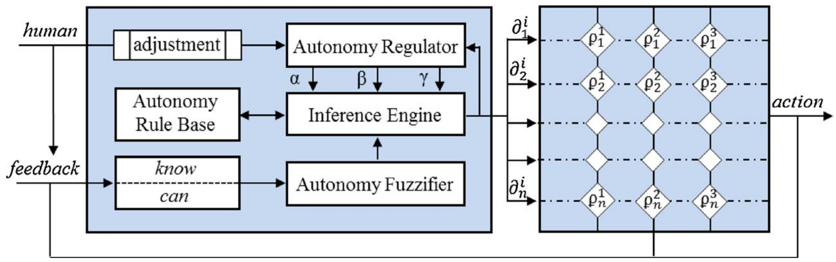

Fig. 3. The FLAA model.

the can_μ membership degree by evaluating the performance of these agents:

$$\mu_f(x;\, \rho^{1,2,3}_{1,2\ldots n}) = \begin{cases} know_\mu = \sum_{i=1}^{n} \frac{agree}{agree + disagree}, & know \\ can_\mu = \frac{1}{n}\sum_{i=1}^{m} \frac{success}{attempts}, & can \end{cases}, \quad x \in t \vee a, \tag{3}$$

where μ_f denotes the membership function of the *know* or *can* fuzzy sets, in which $\mu_f(x)$:X → [0,1], X denotes the adjustable autonomy dimension (universe of discourse) of a multi-agent system, n denotes the number of agents that are operating in a system, and m denotes the number of actions that are performed by an agent.

We then aggregate the autonomy degree of an agent by employing the following centroid function:

$$\partial_x = \frac{\int_{x1}^{x2} \mu_f(x)xdx}{\int_{x1}^{x2} \mu_f(x)dx}, \quad x \in l, \tag{4}$$

where ∂_x denotes the centroid of the autonomy region that is bounded by the interval $[x_1, x_2]$.

The value of ∂_x denotes the autonomy degree of an agent. The autonomy degree sets the l and P options for the agent during a single phase of its run cycle. The next section discusses the distribution of autonomy in a multi-agent system.

3.3. Autonomy distribution

The FLAA model distributes autonomy among the agents in a multi-agent system. This model mainly comprises an autonomy regulator, an autonomy fuzzifier, an autonomy rule base, and an inference engine. Fig. 3 shows the proposed FLAA model. The autonomy regulator allows a human user to manually adjust the autonomy parameters of the system as explained in Section 3.4. The autonomy fuzzifier receives a fuzzy set of *know* and *can* inputs and measures the overlap among different autonomy levels (as shown in Fig. 2). The autonomy rule base contains 15 rules, and a rule has the form of *IF knowX AND canX THEN levelX*.

The inference engine defuzzifies the autonomy of agents by using the measurements from the autonomy fuzzifier and the results of the autonomy rule base. The autonomy is then dynamically distributed among the three main behaviors of agents as follows:

$$\mu_\partial(k;\, \alpha,\, \beta, \gamma) = \begin{cases} \partial^1_k \Leftarrow \alpha know_k + \gamma, & \rho^1_k \\ \partial^2_k \Leftarrow \alpha know_k \wedge \beta can_k + \gamma, & \rho^2_k \\ \partial^3_k \Leftarrow \beta can_k + \gamma, & \rho^3_k \end{cases} \tag{5}$$

where α, β, and γ denote the parameters that can be adjusted by using the autonomy regulator.

The agents deliberate on the possible autonomy levels and then choose a suitable level as follows:

$$\mu_c(k;\, i) = \begin{cases} \text{true}, & \frac{\partial^i_k}{l} > 0 \\ \text{false}, & \text{otherwise} \end{cases}, \tag{6}$$

Table 2
Sample settings of the FLAA model.

level	boundary	property	task	actions
l_1	$0 \le l_1 \le 0.3$	l_1: $p^1_1 \rightarrow \dot{t} \vee \dot{a}$	$t_1 \in l_1 \wedge l_2 \wedge l_3$	$a_1 \in l_1$
l_2	$0.3 < l_2 \le 0.7$	l_2: $p^2_1 \rightarrow \dot{t} \vee \dot{a}$	$t_2 \in l_2 \wedge l_3$	$a_2 \in l_2$
		l_2: $p^2_2 \rightarrow \tilde{t} \vee \tilde{a}$		
l_3	$0.7 < l_3 \le 1$	l_3: $p^3_1 \rightarrow \dot{t} \vee \dot{a}$	$t_3 \in l_3$	$a_3 \in l_3$
		l_3: $p^3_2 \rightarrow \tilde{t} \vee \tilde{a}$		
		l_3: $p^3_3 \rightarrow \bar{t} \vee \bar{a}$		

where μ_c denotes the operational behavior of agents, ρ_k. The autonomy level choice function returns true if and only if $\frac{\partial^i_k}{l} > 0$. Otherwise, the autonomy level choice function returns false and the agent is pushed into choosing between considering other autonomy levels or properties in order to proceed or exposing its current run cycle to an adjustable autonomy operation (e.g., block behavior).

Assume that a multi-agent system has three autonomy levels and manages the adjustable autonomy of three agents. Table 2 shows the proposed settings of the FLAA model for this system. The following examples explain how the autonomy in this system is distributed and adjusted based on the FLAA model.

Example 1: In this scenario, the three agents demonstrate fully and semi-autonomous properties in three phases as shown in Fig. 4.[1] In the first phase, two agents, $\dot{\rho}^1_{1,2}$, autonomously choose $\dot{t}_1$, while the other agent, $\dot{\rho}^1_3$, autonomously chooses $\dot{t}_2$. The measured autonomy degrees in this phase are $\partial^1_1 = 0.66$, $\partial^1_2 = 0.66$, and $\partial^1_3 = 0.3$. Subsequently, the two agents, $\dot{\rho}^1_{1,2}$, can move the third agent, $\widetilde{\rho^1_3}$, into choosing $\tilde{t}_1$ because $\widetilde{\rho^1_3}$ has semi-autonomous properties. In the second phase, two agents, $\dot{\rho}^2_{1,3}$, autonomously choose $\dot{a}_1 \in l_1$, while the other agent, $\dot{\rho}^2_2$, autonomously chooses $\dot{a}_2 \in l_2$. The measured autonomy degrees in this phase are $\partial^2_1 = 0.57$, $\partial^2_2 = 0.60$, and $\partial^2_3 = 0.62$. The differences among these autonomy degrees can be ascribed to the differences between the knowledge and authority history and the results of Eq. (5). Although $\dot{\rho}^2_2$ has a sufficient degree of autonomy to perform in $l_2 (0.3 < \partial^2_2 \le 0.7)$, $\dot{\rho}^2_{1,3}$ can push $\widetilde{\rho^2_2}$ into choosing $\widetilde{a_1}$ because ρ^2_2 has semi-autonomous properties. In the third phase, the agents autonomously execute $\dot{a}_1$ because they all have a sufficient degree of autonomy to perform this action on l_1.

Example 2: In this scenario, the agents are either experiencing an intervention or have blocked behaviors as shown in Fig. 5. In the first phase, three agents, $\dot{\rho}^1_{1,2,3}$, autonomously choose $\dot{t}_1$. Given that the autonomy degree in this phase is $\partial^1_{1,2,3} = 1$, these three agents all proceed with their run cycles. In the second phase, two agents, $\dot{\rho}^2_{1,3}$, autonomously choose $\dot{a}_2$, while the other agent, ρ^2_2, autonomously chooses $\dot{a}_1$. The autonomy degrees of these agents in the second phase are $\partial^2_1 = 0.78$, $\partial^2_2 = 0.82$, and $\partial^2_2 = 0.92$. In this case, $\rho^2_{1,3}$ proceeds with its run cycle, while ρ^2_2 is blocked because its autonomy level does not indicate that

[1] $\dot{x}$, $\tilde{x}$, and $\bar{x}$ for each ρ, t, and a denote fully, semi-, and non-autonomous properties, respectively.

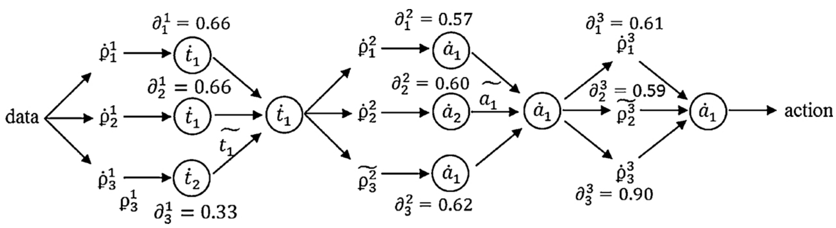

Fig. 4. The first example.

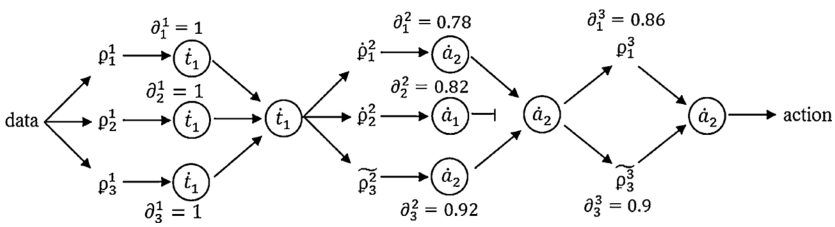

Fig. 5. The second example.

this agent has semi-autonomous properties and considers other options. Eventually, in the third phase, two agents, $\rho^3_{1,3}$, autonomously execute $\dot{a}_2$.

Those agents with non-autonomous properties follow the same mechanism of semi-autonomous agents, that is, they only follow the commands of other agents. They also can seek the inputs or commands of human users.

3.4. Autonomy adjustment

Autonomy adjustment is essentially a process of redistributing autonomy among several agents with an aim to direct their operational behaviors toward producing some desired results [8]. This process is meant for improving agents' performance, and helping them overcome complex and uncertain situations, or avoiding errors or failures [24]. As explained in the previous section, the FLAA model dynamically adjusts the autonomy of the agents and allows human users to manually adjust such autonomy by using the parameters: α (for adjusting the required autonomy knowledge), β (for adjusting the required autonomy authority), and γ (for adjusting the global autonomy). As shown in Eq. (5), the value of γ indicates the extent of change in the overall autonomy of the agents. The values of the aforementioned parameters are sent as a crisp set to the autonomy regulator (as shown in Fig. 3). Assigning positive and negative values to these parameters will increase and decrease the autonomy of agents, respectively. In sum, the adjustable autonomy process can result in any of the following modes:

- Default adjustable autonomy: The autonomy of the agents is left unadjusted, and the autonomy adjustment parameters are set to $\alpha = 1$, $\beta = 1$, and $\gamma = 0$.
- Fully autonomous: The system works beyond the established *know* and *can* conditions, and all three parameters are assigned with high values.
- Non-autonomous: All three parameters are assigned with low values, thereby blocking the autonomous behaviors of all agents.
- Imbalanced adjustable autonomy: The autonomous behaviors of agents are greatly diminished and the system operations are interrupted after the value of one parameter, either α or β, increases massively while that of the other parameter decreases massively.

4. The AEMM-Care system

The proposed FLAA model is implemented and tested in the AEMM-Care system, which detects and prevents fall incidents among the elderly by tracking their daily activities [25]. The system architecture, the testing scenario settings, and the falling situations that are investigated in the test are all adapted from [1,3] and [6]. The AEMM-Care system is situated in an environment filled with uncertainty, which can be attributed to several factors. First, both the sensory data and the data filtering process are fully exposed to noise during the data collection and preparation phases [1,3,25]. Second, as shown in Fig. 6, some human movement patterns are very similar, which may result in confusion when interpreting the recorded positioning coordinates [26]. The combination of these two factors only adds to the ambiguity of the collected data. The AEMM-Care system also operates in a complex environment, and such complexity can be ascribed to the inherent challenges in accurately detecting and interpreting the movement activities of the elderly [4,6,16,17]. The system faces an even higher degree of complexity when identifying fall situations based on the patterns of certain activities.

4.1. Data description

We use the test dataset in [25] for our investigation. This dataset is acquired by the Ubisense system [26] and contains real localization data of daily movement activities for five persons. As shown in Fig. 7, a user wears tags that are attached to his/her right and left ankles, chest, and waist, and each movement data collected by these tags is fed into the Ubisense system. The employed dataset is non-linear and contains 164,860 movement instances with 8 attributes. These attributes denote tag identification data, positioning data, and acceleration data. They are used to describe 11 human movement activities as shown in Fig. 6.

4.2. Prototype design

The AEMM-Care system comprises four main operational units. The data filter unit gathers and filters the sensory data. The data format unit organizes the data, *d*, into a set of movement attributes. The control unit identifies the movement activities of the elderly and then uses the patterns of these activities to decide on the most suitable alarm. The alarm unit notifies human users, including the elderly, relatives, and/or emergency response systems (ERS) personnel, after a fall incident is recorded. The notified users respond accordingly and evaluate the success of the alarm. Fig. 7 presents the AEMM-Care system architecture, its four operational units, and the response of the notified human users.

The control unit, which includes the FLAA model with three autonomy levels and three agents, dynamically allocates autonomy to these agents based on their performance. Each of these agents is equipped with an ordinary random forest machine learning algorithm and bagging algorithm for tree learning [3,6]. Given that each of these algorithms adopts different cross-validation folds, to ensure that each agent has a different level of learning capability. Previous studies have employed the same algorithm to accurately detect the movement

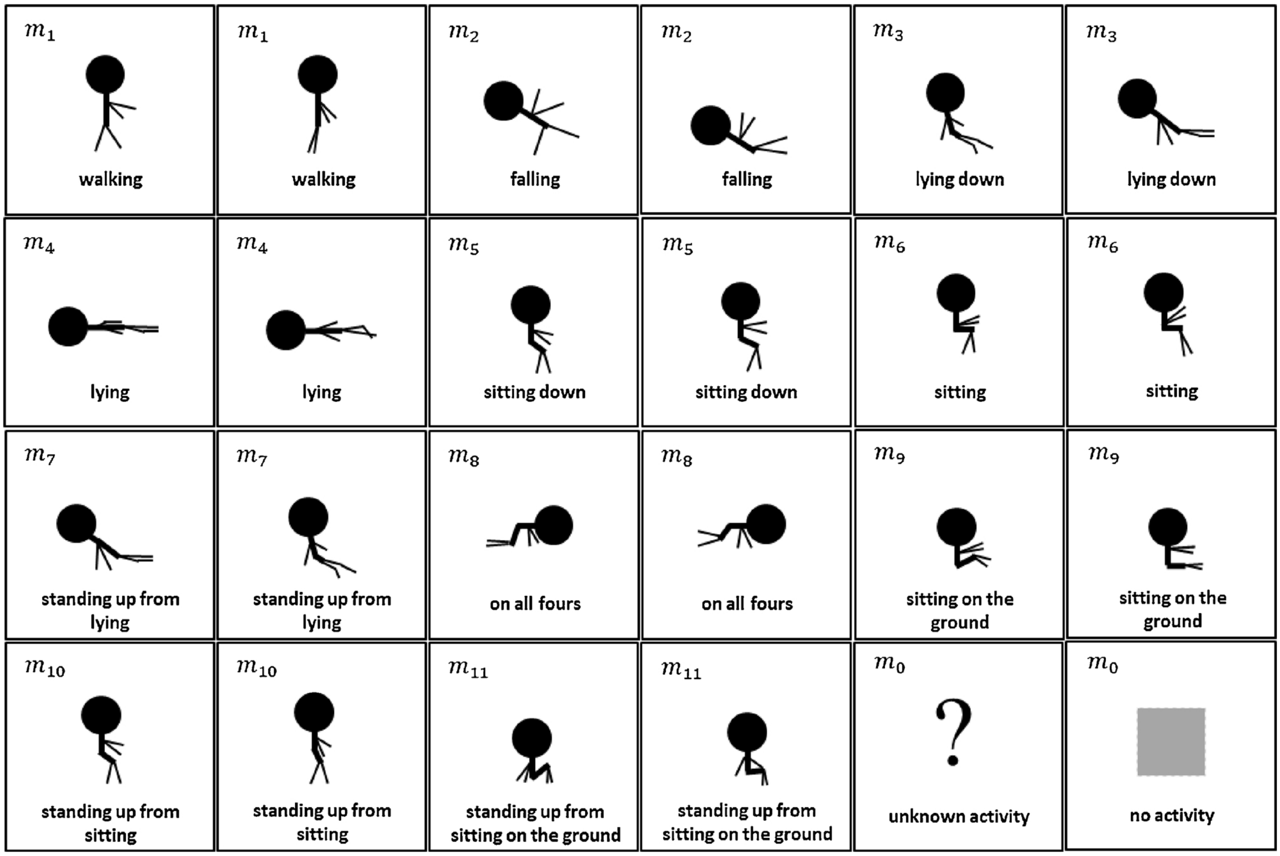

Fig. 6. The 11 elderly activities.

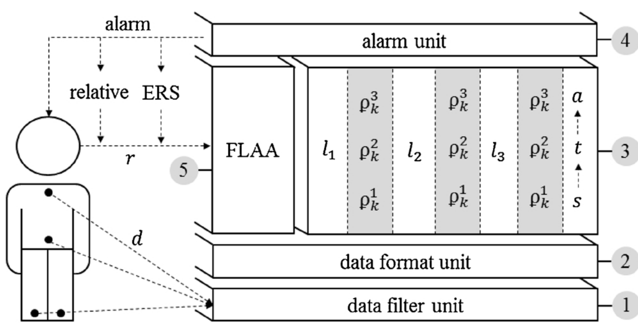

Fig. 7. The architecture of the AEMM-Care system.

Table 4
The operational autonomy setting of the AEMM-Care system.

level	boundary	property	task	actions
l_1	$0 \le l_1 \le 0.3$	l_1: $p_1^1 \to \dot{i} \vee \dot{a}$	$t_1 \in l_1 \wedge l_2$	$a_1 \in l_1$ $a_2 \in l_2$ $a_3 \in l_2$
l_2	$0.3 < l_2 \le 0.7$	l_2: $p_1^2 \to \dot{i} \vee \dot{a}$ l_2: $p_2^2 \to \tilde{i} \vee \tilde{a}$	$t_2 \in l_1 \wedge l_2 \wedge l_3$	$a_1 \in l_1$ $a_2 \in l_2$
l_3	$0.7 < l_3 \le 1$	l_3: $p_1^3 \to \dot{i} \vee \dot{a}$ l_3: $p_2^3 \to \tilde{i} \vee \tilde{a}$ l_3: $p_3^3 \to \bar{i} \vee \bar{a}$		$a_3 \in l_2$ $a_4 \in l_3$

activities of humans. Specifically, the random forest machine learning algorithm has achieved movement detection accuracies of 70.37% in [6], 77.5% in [3], and 79.2% in [1]. The agents perform their tasks in the following operational phases:

- ρ_k^1: The agent inputs the measurable attributes of the transmitted sensory data, d, into the random forest algorithm to predict the movement activities of the elderly, ρ_k^1: $d_{1,2,\ldots} \to m_{1,2,\ldots}$. Afterward, the agent maps the pattern of these activities and then selects the most appropriate task that can be performed in response to a fall situation, ρ_k^1: $(m_{1,2,\ldots} \to \mathrm{s}) \Rightarrow \mathrm{t}$;
- ρ_k^2: The agent selects the most appropriate alarm actions based on the FLAA model and the task selected in the previous phase, ρ_k^2: $\mathrm{t} \to a_{1,2,\ldots}$; and

Table 3
The operational setting of the AEMM-Care system.

situations	tasks	actions
s_1: $x \to \neg\, m_2 \wedge (m_1 \vee m_4 \vee m_5 \vee m_6)$	t_1	$s_1 \to a_1$
s_2: $x \to \neg\, m_2 \wedge ((m_7 \wedge m_4) \vee (m_{10} \wedge m_5 \vee m_6) \vee (m_{11} \wedge m_9))$		$s_2 \to a_2$
s_3: $x \to \neg\, m_2 \wedge ((m_1 \wedge m_3) \vee (m_1 \wedge m_5 \wedge (m_8 \vee m_9)))$		$s_3 \to a_2 \wedge a_3$
s_1: $x \to \neg\, m_2$	t_2	$s_1 \to a_1$
s_2: $x \to m_2 \to \neg\, m_4$		$s_2 \to a_2 \wedge a_3$
s_3: $x \to m_2 \to m_4 \to m_4$		$s_3 \to a_2 \wedge a_3 \wedge a_4$

- ρ_k^3: The agent executes the selected alarm action, ρ_k^3: $a_{1,2,\ldots} \rightarrow$ E' .

Falling tasks, situations, and actions are included in the setting of the AEMM-Care system. These actions alert human users about the fall situations detected or predicted by the agents, to which these humans respond according to the types of alarm actions (positive triggered alarm, r_1^+, negative triggered alarm, r_2^-, and negative not triggered alarm, r_3^-). The performance of the agents is then evaluated based on their responses to the detected or predicted fall situations. Table 3 presents the setting of the AEMM-Care system, which includes two tasks (where t_1 and t_2 denotes fall prevention and detection, respectively), six situations (where a situation, s, denotes a specific pattern of activities), and four alarm actions (where a_1, a_2, a_3, and a_4 denote no alarm, alarm elderly, alarm relative, and alarm ERS, respectively).

The falling tasks and alarm actions are distributed among the autonomy levels l_1, l_2, and l_3 based on the fall situations and alarm actions. Table 4 shows the relationship between the falling tasks and alarm actions with the autonomy levels in the AEMM-Care system.

The compositions in Tables 3 and 4 set the FLAA model of the AEMM-Care system but there are some other alternative settings that can be applied. The distribution of tasks and actions to each autonomy level indicates the possibility and risk of wrongly interpreting or predicting a fall and triggering the alarms. The agents select t_1 or t_2 based on their interpretation of the sensor readings of elderly activities, and the pattern of these activities. Given that t_1 represents deducible task because it has more complicated patterns, lower chances to accurately predict a falling situation but lower risk of wrongly interpreting the possibility of falling compared with t_2, and hence, t_2 represents critical task which imposes different autonomy options on the triggering of alarms. Consequently, a_1 represents primitive action as it has no consequence of wrong alarm, a_2 and a_3 represent deducible actions as they have consequence of wrong alarms and a_4 represents critical action as it has a greater consequence of wrong alarm. Algorithm 2 presents the application of the FLAA model in the AEMM-Care system.

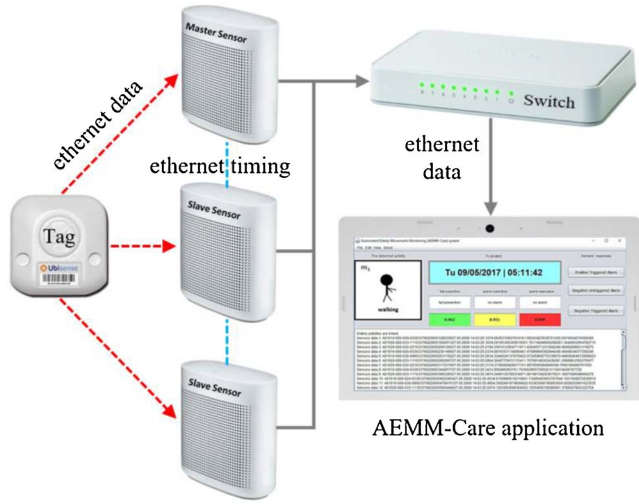

Fig. 8. The AEMM-Care system.

Algorithm 2: The application of the FLAA model in the AEMM-Care system

begin;
$< d, m, s, t, l, \partial, r, \alpha, \beta, \gamma, \rho > \leftarrow$ *initialize* $(d, m, s, t, l, \partial, r, \alpha, \beta, \gamma, \rho)$;
while (d <> null) do
  //ρ_k^1: *tasks-selection*
  for each active ρ_k^1 of GO do
    $d_k \leftarrow$ *read* (ρ_k^1, sensors);
    $s_k \Leftarrow m_k \leftarrow$ *observe* (ρ_k^1, d_k);
    $t_i \leftarrow$ *decide* (ρ_k^1, s_k);
  end-for
  $< \alpha, \beta, \gamma > \leftarrow$ *update* (α, β, γ);
  $\partial^1_{1,2,\ldots} \leftarrow$ FLAA($\rho^1_{1,2,\ldots}$, $t_{1,2}$);
  if (*obtain* ($\rho^1_{1,2,\ldots}$, $l_{1,2,3}$)) then
    //ρ_k^2: *actions-selection*
    for each active ρ_k^2 of GO do
      $a_{i,j} \leftarrow$ *decide* (ρ_k^2, t_i);
    end-for
    $< \alpha, \beta, \gamma > \leftarrow$ *update* (α, β, γ);
    $\partial^2_{1,2,\ldots} \leftarrow$ FLAA ($\rho^2_{1,2,\ldots}$,($a_{i,j}$, $r_{i,j}$));
    if (*obtain* ($\rho^2_{1,2,\ldots}$, $l_{1,2,3}$)) then
      //ρ_k^3: *action-execution*
      $< \alpha, \beta, \gamma > \leftarrow$ *update* (α, β, γ);
      $\partial^3_{1,2,\ldots} \leftarrow$ FLAA ($\rho^3_{1,2,\ldots}$, $r_{i,j}$);
      if (*obtain* ($\rho^3_{1,2,\ldots}$, $l_{1,2,3}$)) then
        alarm $\leftarrow$ *execute* ($\rho^3_{1,2,\ldots}$, $a_{i,j}$);
        $r_{i,j} \leftarrow$ *respond* (*human*, *alarm*);
      end-if
    end-if
  end-if
end-while
end;

Table 5
The AECMM-Care system test results in the fully autonomous mode (i).

test1	task		action		response			success (%)	
	type	number	type	number	r_1^+	r_2^-	r_3^-	action	task
run1	t_1	1132	a_1	735	497	162	76	67.62	47.59
			a_2	467	212	192	63	45.40	
			a_3	269	80	123	66	29.74	
	t_2	611	a_1	527	458	48	21	86.91	71.52
			a_2	122	84	25	13	68.85	
			a_3	122	84	25	13	68.85	
			a_4	83	51	22	10	61.45	
run2	t_1	1227	a_1	796	562	139	95	70.60	53.91
			a_2	506	272	185	49	53.75	
			a_3	289	108	122	59	37.37	
	t_2	517	a_1	440	383	41	16	87.05	70.32
			a_2	102	68	27	7	66.67	
			a_3	102	68	27	7	66.67	
			a_4	69	42	21	6	60.87	
run3	t_1	1189	a_1	771	526	177	68	68.22	48.13
			a_2	490	196	213	81	40.00	
			a_3	282	102	105	75	36.17	
	t_2	551	a_1	445	395	35	15	88.76	73.60
			a_2	103	75	20	8	72.82	
			a_3	103	75	20	8	72.82	
			a_4	70	42	17	11	60.00	

Table 6
The AEMM-Care system test results in the default adjustable autonomy mode (ii).

test2	task		action		response			success (%)	
	type	number	type	number	r_1^+	r_2^-	r_3^-	action	task
run1	t_1	1018	a_1	614	480	82	52	78.18	60.72
			a_2	405	228	110	67	56.30	
			a_3	197	94	61	42	47.72	
	t_2	671	a_1	524	475	27	22	90.65	82.56
			a_2	118	93	16	9	78.81	
			a_3	118	93	16	9	78.81	
			a_4	61	50	5	6	81.97	
run2	t_1	1073	a_1	598	432	98	68	72.24	57.86
			a_2	409	237	115	57	57.95	
			a_3	159	69	54	36	43.40	
	t_2	620	a_1	574	491	49	34	85.54	77.50
			a_2	130	96	19	15	73.85	
			a_3	130	96	19	15	73.85	
			a_4	56	43	6	7	76.79	
run3	t_1	987	a_1	655	590	42	23	90.08	73.63
			a_2	419	311	71	37	74.22	
			a_3	228	163	35	30	71.49	
	t_2	609	a_1	544	504	22	18	92.65	77.28
			a_2	129	91	23	15	70.54	
			a_3	129	91	23	15	70.54	
			a_4	65	49	9	7	75.38	

4.3. Prototype implementation and testing

The AEMM-Care system adopts the Ubisense architecture which contains the hardware of ultrasonic sensors, tags, network switch and a computer system and the software of the system application. Fig. 8 presents an overview of the AEMM-Care system. We implement the AEMM-Care system modules in Java and the FLAA model as well as the agents in Java Agent Development framework (Jade). Jade has attracted wide usage in the development of agent-based applications [27]. The FLAA Model aims to reduce the spurious transition effects of the AEMM-Care system. This model stimulates accurate collaborative results even if an agent misinterprets an activity or pattern of a set of activities. It dynamically shifts the agents' autonomy toward a direction that can improve their fall detection and prevention accuracy.

The AEMM-Care system test involves performing two test series, where a test series consists of three independent runs. In the first test series (i), the system operates in full-autonomy and beyond the FLAA conditions. In the second test series (ii), the system operates in default adjustable autonomy and according to the FLAA conditions (as explained in Section 3.4). The test scenario includes six situations of movement activities with different complexity levels, and an alarm must be raised in four of these situations (as shown in Table 2). This test setup has three objectives. First, to ensure that the agents monitor the movement activities of the elderly, interpret the patterns of these activities, respond to falling incidents, successfully complete their fall detection and prevention tasks, and perform the necessary alarm actions. Second, to ensure that the FLAA model sets appropriate limits of the autonomy levels and dynamically distribute autonomy to the agents based on their performance. Third, to obtain comprehensive results that validate the importance of the research contributions.

5. Results and discussion

The AEMM-Care system achieves irregular success rates in the fully autonomous mode (i), and such irregularity can be ascribed to the variances in the random forest agent's learning of different cross-validation folds. t_1 and its actions have much lower success rates than t_2 and its actions because predicting those movement activity patterns that lead to fall situations is a complex process. The success rates of both t_1 and t_2 also do not change after multiple runs. The success rate of t_1 reaches its peak value of 53.91% in run2 and lowest value of 47.59% in run1, while the success rate of t_2 reaches its peak value of 73.60% in run3 and lowest value of 70.32% in run2. Table 4 shows the results of the AEMM-Care system after three runs in the fully autonomous mode.

Similarly, in the default adjustable autonomy mode (ii), the AEMM-Care system shows irregular success rates, and t_1 and its actions have lower success rates than t_2 and its actions. The FLAA model also fails to address the high complexity in predicting the movement activity patterns of the elderly. However, both t_1 and t_2 show obvious improvements in their success rates after multiple runs. Specifically, the success rate of t_1 reaches its peak value of 73.63% in run3 and lowest value of 57.86% in run2, while that of t_2 reaches its peak value of 82.56% in run1 and lowest value of 77.28% in run3. Table 5 shows the results of the AEMM-Care system test after three runs in the default adjustable autonomy mode (Table 6).

The success rates of t_1 in the adjustable autonomy mode are obviously greater than those in the fully autonomous mode (49.88% in run1 and 64.07% in run2). The FLAA model also remarkably increases r_1^+, decreases r_2^-, and slightly decreases r_3^-. Fig. 9 shows the changes in the r_1^+, r_2^-, and r_3^- responses of t_1 in the fully autonomous (i) and adjustable autonomy (ii) modes.

The success rates of t_2 in the default adjustable autonomy mode are slightly higher than those in the fully autonomous mode (71.81% in run1 and 79.11% in run2). The FLAA model remarkably increases r_1^+, decreases r_2^-, and slightly increases r_3^-. Fig. 10 shows the changes in the r_1^+, r_2^-, and r_3^- responses of t_2 in these two autonomy modes.

The variations in Figs. 9 and 10 indicate that the performance of agents and the accuracy of the results that the default adjustable autonomy mode outperforms the fully autonomous mode. The improvement in the results of the default adjustable autonomy mode can be explained by the effect of the FLAA model on the agents' selection of fall prevention and detection tasks and on their corresponding actions. Such effect is reflected in the changes in the negative triggered alarm, r_2^-, and the negative not triggered alarm, r_3^-, in both the default adjustable autonomy and fully autonomous modes. Specifically, r_2^- decreases while r_3^- increases in the default adjustable mode. The reduced r_2^- shows that the FLAA model decreases the instances of triggering false or undesirable alarms, while the increased r_3^- shows that the model blocks the agents' behaviors that can trigger desirable alarms. Such behavior can be attributed to the mismatch between the autonomy levels of the

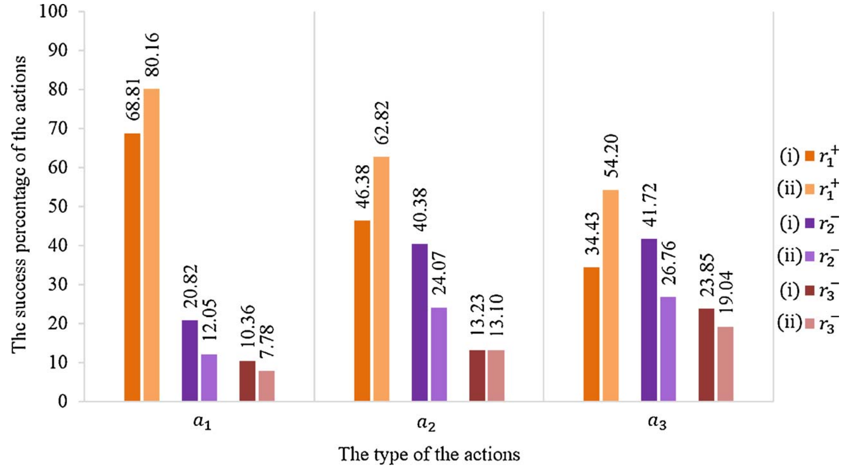

Fig. 9. The variation in the humans' responses to the alarm actions of the AEMM-Care system.

agents and the autonomy degrees of their tasks or actions.

Fig. 11 shows the autonomous behavior of agents in the default adjustable autonomy mode when performing fall prevention and detection tasks and when executing alarm actions. It also presents the average autonomy degrees recorded in each run phase and operational autonomy levels. The agents' performance in t_1 and t_2 and their corresponding alarm actions in the three runs of t_2 result in slightly imbalanced adjustable autonomy levels because these agents have reached a high agreement (knowledge) on their task and action selection yet their alarm actions (authority) have achieved low success rates. Some variations can also be seen in the selection agreement/disagreement and alarm success/failure of these agents in specific phases of their run cycles. On the one hand, these agents have reached a high and low agreement in the *tasks-selection* phase, $\rho^{1}_{1,2,3}$, of t_2 and t_1, respectively. The FLAA model also blocks the attempts of unqualified agents because of their unsatisfactory *know* condition (i.e., low agreement), which indicates that these agents have insufficient knowledge about the elderly movement activities that they are supposed to monitor. Therefore, the sensory data on elderly activities are highly uncertain and need to be revised. On the other hand, a high and low alarm success rate is achieved in the *actions-execution* phase, $\rho^{3}_{1,2,3}$, of t_2 and t_1, respectively. The FLAA also blocks the attempts of those agents with an unsatisfactory *can* condition (i.e., low success rate), which suggests that these agents still make the wrong decisions even if they have sufficient knowledge and high agreement on the elderly activities that they are supposed to monitor. These poor decisions can be ascribed to the very complex nature of detecting and interpreting the movement activity patterns of the elderly. Therefore, the operational behaviors of agents in their decision-making cycle need to be modified.

In sum, the FLAA model provides an autonomy management mechanism that overlaps the current performance of a system with the aggregation of its past performance, and such overlap, in turn, can direct the agents' autonomy toward achieving a competent performance. This model is able to reduce the spurious transition effects of the AEMM-Care system and guaranteeing competent results even when an agent misinterprets an activity or a pattern of a set of activities. It successfully improves the fall detection and prevention accuracy of agents and consequently increases the satisfaction of users with the AECMM-Care system. The performance improvements can be attributed to the flexibility granted by the model that enables all agents to operate at different autonomy levels and obtain various autonomy properties. Apart from adjusting the autonomy of agents based on their performance, the FLAA model allows human users to manually adjust the autonomy of these agents according to their preferences. The FLAA model is proven to be useful for multi-agent systems that are operating in complex environments where some agents tend to make wrong decisions and execute wrong actions.

This study offers several contributions to research and practice. First, it proposes the FLAA model, which uses fuzzy logic to formulate the adjustable autonomy of a multi-agent system, and then test this model in AEMM-Care system, a remote elderly movement monitoring system. Second, it proposes random forest agents with three adjustable autonomous run phases. Third, it highlights the importance of recognizing specific activity patterns of an elderly to improve the

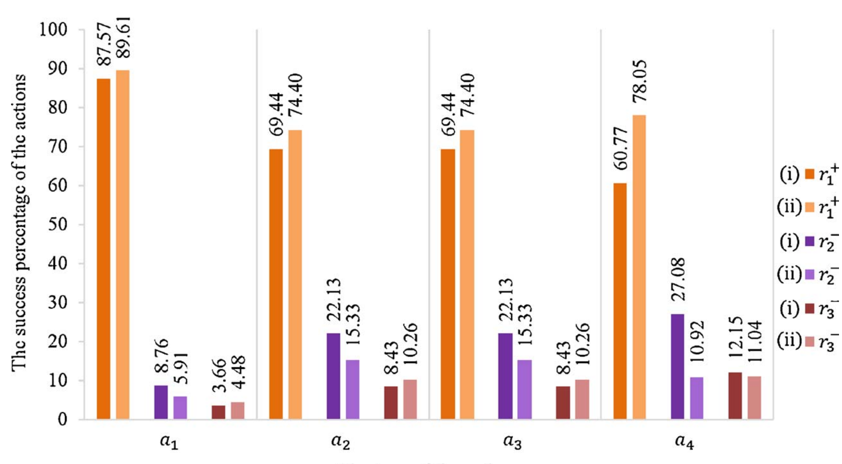

Fig. 10. The variation in the humans' responses to the alarm actions of the AEMM-Care system.

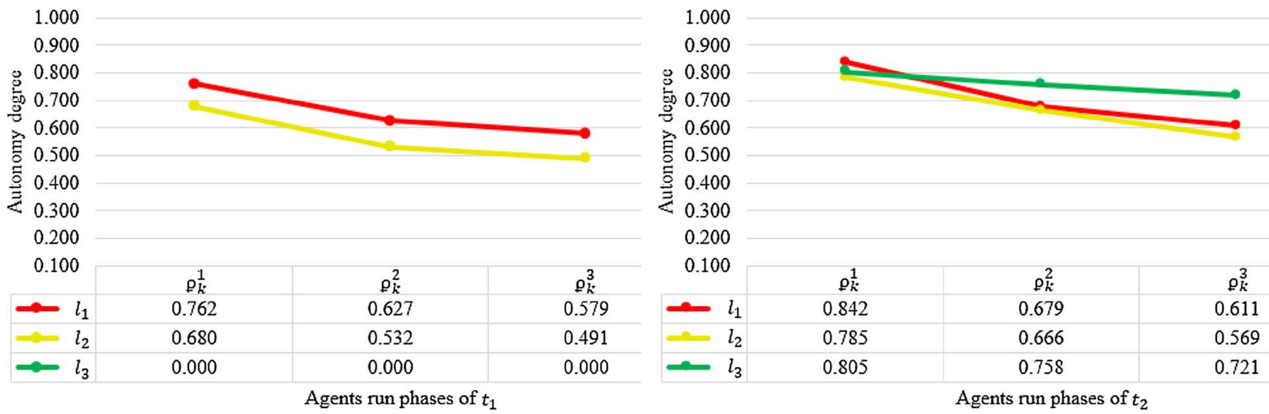

Fig. 11. The average autonomy degrees of the agents.

accuracy of preventing and identifying the elderly falling situations. Unlike other works that only focus on classifying independent movement activities (i.e., sitting or walking), this work classifies the pattern of dependent movement activities (i.e., the fall situations depicted in Table 3). In this way, this study accounts for both the complexity of classifying movement activity patterns and the related error margin that results from dependent classification. However, as a limitation of this study, it does not investigate the behavior of multi-agent systems in the long term or examine the effect of the agents' learning on their fall detection and prevention accuracies. Moreover, the test data do not reflect the patterns of some real-world movement activities or fall situations.

6. Conclusion and future work

We perform this research with an aim to develop a model for managing the adjustable autonomy of multi-agent systems that are operating in complex environments. The FLAA model quantifies the autonomy of agents based on knowledge and authority criteria, sets proper autonomy levels for these agents, and manages adjustable autonomy of the agents based on their performance. The autonomy management helps the agents to make highly flexible and efficient decisions and guides them toward achieving reliable actions. The autonomy adjustment is effective on the tasks deliberation, actions selection and actions execution behaviors of the agents. It is dynamically performed by the model and can be imposed by human users within the agents run cycles.

We test the FLAA model in the AEMM-Care system to detect and prevent elderly fall situations. The test results show that the system achieves an impressive 79.11% success rate in detecting falls and a moderate 64.07% success rate in preventing falls. This finding can be ascribed to the complex nature of predicting those activity patterns that can result in a fall. These results all point toward how managing the autonomy of a system can boost the agents' performance, the efficiency of their decisions, and the reliability of their actions. The knowledge autonomy conditions ensure the continuity of the agents' cycle, while the autonomy authority conditions may block some alarms due to faulty actions resulting from the poor decisions of agents.

We test the FLAA model based on the knowledge and authority attributes. The fuzzy logic technique greatly expedites the adoption, evaluation, and comparison of these attributes with other autonomy attributes, including confidence, consistency, trust, and motivation. Using these other attributes may further improve the accuracy of fall detection and prevention systems and consequently, increase the satisfaction of the system users. Additionally, the influence of different autonomy settings and manual autonomy adjustment on the behavior and performance of agents presents an interesting subject that we shall explore in our future work.

Authors contributions

The contributions of this work are represented by the following points:

- The first contribution of this work lies in its development of the FLAA model, which uses fuzzy logic to formulate the adjustable autonomy of a multi-agent system. This model is applied in a remote elderly movement monitoring system.
- The second contribution of this work lies in its proposal of random forest agents with three adjustable autonomous run phases.
- The third contribution of this work lies in the recognition of specific patterns of activities to identify and prevent elderly falling situations. The related work classifies the pattern of independent elderly movement activities, such as walking, sitting, or lying, while our work also classifies the pattern of dependent movement activities, such as the fall situations presented in Table 3. Subsequently, the complexity of the classification process and its related error margin that results from dependent classification are both accounted for in our work.

Conflict of interest and authorship confirmation

- All authors have participated in (a) conception and design, or analysis and interpretation of the data; (b) drafting the article or revising it critically for important intellectual content; and (c) approval of the final version.
- The Article I have submitted to the journal for review is original, has been written by the stated authors and has not been published elsewhere.
- The Images that I have submitted to the journal for review are original, was taken by the stated authors, and has not been published elsewhere.
- This manuscript has not been submitted to, nor is under review at, another journal or other publishing venue.
- The authors have no affiliation with any organization with a direct or indirect financial interest in the subject matter discussed in the manuscript.

Summary points

- We conduct this research with the aim to establish an adjustable autonomy model for multi-agent systems that facilitates efficiency and flexibility to the agents' decisions and reliability to their actions.
- We propose a Fuzzy Logic-based Adjustable Autonomy (FLAA) model to manage the autonomy of multi-agent systems that perform in complex environments.

- We apply the FLAA model in an Automated Elderly Movements Monitoring system (AEMM-Care). The AEMM-Care system monitors elderlies' daily activities, carries out fall prevention and fall detection tasks and performs fall alarm actions.
- The AEMM-Care system results of the fall prevention show an intermediate success rate of 64.07%. This is due to the complexity of predicting the pattern of the activities that might lead to the falling situation. The results of the fall detection task are encouraging in which the system is able to achieve 79.11% success rate.

Acknowledgements

This project is sponsored by the postdoctoral grant of Universiti Tun Hussein Onn Malaysia (UTHM) under Vot D004 and partially supported by the Tier 1 research grant scheme of UTHM under Vot U893.

References

[1] M. Lustrek, H. Gjoreski, N.G. Vega, S. Kozina, B. Cvetkovic, V. Mirchevska, M. Gams, Fall detection using location sensors and accelerometers, IEEE Pervasive Comput. 14 (4) (2015) 72–79.
[2] I. Azimi, A.M. Rahmani, P. Liljeberg, H. Tenhunen, Internet of things for remote elderly monitoring: a study from user-centered perspective, J. Ambient Intell. Hum. Comput. 8 (2) (2017) 273–289.
[3] B. Kaluža, B. Cvetković, E. Dovgan, H. Gjoreski, M. Gams, M. Luštrek, V. Mirchevska, A multi-agent care system to support independent living, Int. J. Artif. Intell. Tools 23 (01) (2014) 1440001.
[4] L. Almazaydeh, K. Al-Otoon, A. Al Dmour, K. Elleithy, A panoramic study of fall detection technologies, Int. J. Comp. Sci. Issues (IJCSI) 13 (3) (2016) 62.
[5] D. Isern, A. Moreno, A systematic literature review of agents applied in healthcare, J. Med. Syst. 40 (2) (2016) 43.
[6] B. Cvetković, B. Kaluža, M. Gams, M. Luštrek, Adapting activity recognition to a person with Multi-Classifier Adaptive Training, J. Ambient Intell. Smart Environ. 7 (2) (2015) 171–185.
[7] E. Marcon, S. Chaabane, Y. Sallez, T. Bonte, D. Trentesaux, A multi-agent system based on reactive decision rules for solving the caregiver routing problem in home health care, Simul. Modell. Pract. Theory 74 (2017) 134–151.
[8] S.A. Mostafa, M.S. Ahmad, A. Mustapha, Adjustable autonomy: a systematic literature review, Artif. Intell. Rev. (2017) 1–38.
[9] J. Jaafar, E. McKenzie, Decision making method using fuzzy logic for autonomous agent navigation, Electr. J. Comp. Sci. Inf. Technol. 3 (1) (2011).
[10] S. Mostafa, S.S. Gunasekaran, M.S. Ahmad, A. Ahmad, M. Annamalai, A. Mustapha, Defining tasks and actions complexity-levels via their deliberation intensity measures in the layered adjustable autonomy model, Intelligent Environments (IE), 2014 International Conference, IEEE, 2014, pp. 52–55.
[11] J.M. Bradshaw, P.J. Feltovich, H. Jung, S. Kulkarni, W. Taysom, A. Uszok, Dimensions of adjustable autonomy and mixed-initiative interaction, International Workshop on Computational Autonomy, Springer, Berlin Heidelberg, 2003, pp. 17–39.
[12] S.A. Mostafa, M.S. Ahmad, M. Annamalai, A. Ahmad, S.S. Gunasekaran, A conceptual model of layered adjustable autonomy, Advances in Intelligent Systems and Computing, Springer Verlag, 2013, pp. 619–630.
[13] L. Lin, M.A. Goodrich, Sliding autonomy for UAV path-planning: adding new dimensions to autonomy management, Proceedings of the 2015 International Conference on Autonomous Agents and Multi-agent Systems (2015) 1615–1624 International Foundation for Autonomous Agents and Multi-agent Systems.
[14] S.A. Mostafa, M.S. Ahmad, A. Ahmad, M. Annamalai, A. Mustapha, A dynamic measurement of agent autonomy in the layered adjustable autonomy model, Studies in Computational Intelligence, Springer-Verlag, 2014, pp. 25–35.
[15] A. Armentia, U. Gangoiti, R. Priego, E. Estevez, M. Marcos, A multi-agent based approach to support adaptability in home care applications, IFAC-PapersOnLine 48 (10) (2015) 1–6.
[16] S. Iqbal, W. Altaf, M. Aslam, W. Mahmood, M.U.G. Khan, Application of intelligent agents in health-care: review, Artif. Intell. Rev. 46 (1) (2016) 83–112.
[17] J.S.R. Jang, C.T. Sun, E. Mizutani, Neuro-fuzzy and Soft Computing, a Computational Approach to Learning and Machine Intelligence, (1997).
[18] M.S. Couceiro, J.A. Tenreiro Machado, R.P. Rocha, N.M. Ferreira, A fuzzified systematic adjustment of the robotic Darwinian PSO, Rob. Auton. Syst. 60 (2012) 1625–1639.
[19] H.S. Seo, S.J. Youn, K.W. Oh, A fuzzy reinforcement function for the intelligent agent to process vague goals, Fuzzy Information Processing Society, 2000. NAFIPS. 19th International Conference of the North American, IEEE, 2000, pp. 29–33.
[20] Y. Li, P. Musilek, L. Wyard-Scott, Fuzzy logic in agent-based game design, IEEE, Fuzzy Information, 2004. Processing NAFIPS'04. IEEE Annual Meeting, vol. 2, 2004, pp. 734–739.
[21] H. Hexmoor, C. Castelfranchi, R. Falcone (Eds.), Agent Autonomy, vol. 7, Springer Science & Business Media, 2012.
[22] N.R. Jennings, L. Moreau, D. Nicholson, S. Ramchurn, S. Roberts, T. Rodden, A. Rogers, Human-agent collectives, Commun. ACM 57 (12) (2014) 80–88.
[23] S.A. Mostafa, M.S. Ahmad, A. Mustapha, M.A. Mohammed, A concise overview of software agent research, modeling, and development, Softw. Eng. 5 (1) (2017) 8–25.
[24] S.A. Mostafa, M.S. Ahmad, A. Mustapha, M.A. Mohammed, Formulating layered adjustable autonomy for unmanned aerial vehicles, Int. J. Intell. Comput. Cybern. 10 (4) (2017) 430–450.
[25] B. Kaluža, V. Mirchevska, E. Dovgan, M. Luštrek, M. Gams, An agent-based approach to care in independent living, International Joint Conference on Ambient Intelligence, November, 2010, pp. 177–186 Springer.
[26] P. Steggles, S. Gschwind, Ubisense—A Smart Space Platform Technical Report, Ubisense, 2005.
[27] F.L. Bellifemine, G. Caire, D. Greenwood, Developing Multi-agent Systems with JADE vol. 7, John Wiley & Sons, 2007.