A Neural Network Agent Based Approach to Activity Detection in AmI Environments

Fernando Rivera-Illingworth, Victor Callaghan, Hani Hagras
Department of Computer Science, University of Essex, Wivenhoe Park, Colchester CO4 3SQ, UK {friver, vic, hani}@essex.ac.uk

Abstract

Many countries are facing the problem of caring economically for their ageing population. One approach to this is the development of environments which possess ambient intelligence that will provide care for older people while assisting them with their everyday life activities. One function that is required in care provision is spotting abnormal behaviours as this might be an indicator of a problem requiring attention from a carer. Agent technology can be employed to detect abnormalities by first learning a normal set of personal behaviours and then detecting deviations from these. This paper presents a novel connectionist embedded agent architecture that combines the use of unobtrusive and relatively simple sensors and employs a constructive algorithm with temporal capabilities which is able to recognize different high level activities (such as "sleeping", "working at computer", "eating"), and identify abnormal behaviours. The network is trained in an online mode, and is able to adapt and expand itself as new data are made available over time or it can add new output nodes to represent new classes or accommodate the abnormal instances. The developed connectionist approach is not computationally demanding and hence it can be integrated into the limited processor-power embedded computing platforms used in intelligent domestic environments.

Keywords: Activity recognition; Ambient intelligence; Agents; Neural systems; Abnormality detection; novelty detection

1. Introduction

The population's ageing problem poses interesting challenges to the future of human society. The development of new technologies that can ease this demographical transition by providing support and delivering highly automated and comprehensive intelligent care at home constitutes a very interesting and challenging problem. Recent studies conducted by the United Nations Population Division show that the number of people aged 60 years or older was estimated to be 629 million by 2002, and is projected to grow to almost 2 billion by 2050, or approximately 21.4% of the population of the world. As a consequence, fewer people will be able to provide care to elder people by formal or informal means also with an increasing financial burden on the people in working age. In addition, it is expected that there will be a much larger number of older people living alone, so there is the necessity of new ways to provide support to them.

Despite the fact that elder people prefer to "age in place", in order for people to live independently they must be capable to perform certain activities and maintain some independence. One of the metrics used [1], [13], [16], [20] is the ability to perform basic Activities of Daily Living or ADLs, and the capacity to carry out Instrumental Activities of Daily Living or IADLs. ADLs focus on assessing the person's ability to perform basic self care activities such as eating, dressing, bathing, going to the toilet and transferring in and out of bed/chair and walking. IADLs measure activities related to independent living and include preparing meals, shopping for personal items, medication management, managing money, using the telephone, doing laundry, and transportation[1].

One of the warning signs of Alzheimer's disease is the difficulty to perform familiar tasks. People with dementia often find it hard to complete everyday tasks that are so familiar that usually we do not think about how to do them. A person with Alzheimer's may not know the steps for preparing a meal,

[1] http://www.cdc.gov/nchs/datawh/nchsdefs/adl.htm

using a household appliance, or participating in a lifelong hobby or he/she might cook a meal but forget to serve it, etc[2] .

In order to deal with the above problems, there is a need to develop new solutions to provide care for the elderly; a possible solution is the use of intelligent environments. An intelligent environment is a space that people live in (e.g. room, a house etc) in which all the services (e.g. heat, light, communication, entertainment, security etc) are managed, intelligently, by computers so as to support the occupants in their daily activities. These environments vary in complexity being in their most simple form just a collection of automated systems or in their more complex form "intelligent" systems that can be taught by the user, or even learn for themselves, how best to serve the occupant's needs.

These intelligent environments open up new and creative possibilities for supporting the lives of ordinary people in their homes. A field that promises to change in a radical way is the care one, more specifically the provision of care services at home. For example, limited resources, especially in helping with ADLs encourage the development of technologies that can assist and support the elderly person and possibly its carers. ADLs depend on regular patterns of behaviour, and by learning such habitual patterns, it could be possible to recognise significant deviations from the normal and to infer possible problems or to provide appropriate assistive control. So by using raw sensor data (sometimes incomplete and noisy), we aim to distinguish different human behaviours inside a domestic environment (to characterize high-level activities inside it, such as "sleeping", "relaxing", "eating", etc). Once the "normal" activities are detected, different kind of abnormal activities can be spotted. These abnormalities can be activities that have not been previously performed inside the environment, abnormalities in the lifestyle rhythms or abnormalities in the sequence and frequency of life activities.

One possible approach for activity and abnormal behaviour detection is achieved by the use of embedded agents. Embedded intelligence can be regarded as the inclusion of some of the reasoning, planning and learning processes, typical of a person, within an artefact. The embedded computers that contain such an intelligent capability are normally referred to as "embedded agents" [9]. In order to use this solution, the learning techniques used must be adaptable to the continuous changes in the environment and user behaviour. In addition, the used learning technique must be computationally undemanding to fit on devices with limited memory and processing power. The agent can be employed to determine a normal set of personal behaviours and then detect deviations associated with the onset of illness or physical disability.

The rest of the paper is organized as follows. Section 2 presents the main approaches used in human behaviour recognition systems. Section 3 describes the architecture for the iDorm, the testbed which has been used for evaluating our techniques. The embedded computing platform will be presented in Section 4 whilst the agent's connectionist architecture will be described in Section 5. Section 6 will present the experiments and results. Finally, section 7 will summarize the work.

2. Human Behaviour Recognition Systems

Several approaches have been reported for sensing and acquiring data from environments and the activities of people inside them. One possible approach is the use of cameras and microphones [2] and vision-based tracking modules [13]. This approach usually has produced good results under laboratory conditions, but faces many problems when used in real environments and can be perceived as invasive to the user's privacy. Other systems use wearable sensors like RFID tags [7], bracelets [13], or sensors attached to the body (like accelerometers). The main problem with these kinds of sensors is that they require cooperation from the user and it is possible that he/she forgets to wear them. In [17], data from a GPS sensor stream coupled with some commonsense rules are used to illustrate the possibility of inferring high-level behaviours from low-level sensors. More recently a system has been developed at MIT [15] that performs activity recognition using only using simple and easy to install "tape on and forget" sensors at a low cost inside a house.

There are two main approaches employed in human behaviour recognition systems which are the statistical and connectionist approaches (relying basically in the use of neural networks). The former

[2] http://www.alz.org/AboutAD/Warning.asp

approach is usually combined with expert analysis in order to identify correlations between sensors and activities.

For statistical methods, there has been work in developing a naïve Bayes classifier for activity labelling via indirect observation of the sensor activation signals and a context-aware experience sampling tool (ESM), which was implemented on a PDA [15]. Dependence between activities and object-involvement has been modelled using Bayesian belief net representations [7]. Although some Bayes-based approaches have proved to be good classifiers, some of them can't be applied to real domains with online learning because they can be computationally expensive. In addition, many different types of Markov processes have been tried [13]. However, Hidden Markov Models have the disadvantage of resulting in very complex networks and their learning algorithms have proved difficult to apply to problems involving numerous low-level sensors.

There have been a number of approaches based on connectionist systems. One of the most important is the work of Mozer who applied such techniques to the control of heating and lighting in a smart home referred to as the Neural Network House [14]. Other work [1] has used unsupervised temporal neural networks for security systems in order to learn from activities and detect abnormal behaviours inside a nuclear power plant.

Whilst neural networks have shown themselves capable of providing good solutions to the sort of problem domain we are addressing, it has been often argued that they suffer from the incomprehensibility of the decision making process (being essentially "black boxes"). It also has been stated [15] that adapting neural networks to the continuous changes in an intelligent space might require retraining the whole network which can be a computationally expensive process especially in cases that require on-line adaptation. However, some connectionist approaches have been developed to overcome both the neural networks' drawbacks of the incomprehensibility of the decision making process and the difficult process of adapting them to continuous changes in the environment. This will be the focus of the work presented in the following sections.

For the system presented on this paper, we are combining the use of unobtrusive and relatively simple sensors, because it seems to be a better approach in order to reach the notion of a ubiquitous system. Previous experiences [15], [17] have shown that is feasible to build such systems without compromising the quality of the results.

3. The iDorm Architecture

The proposed system has been implemented and tested on the intelligent Dormitory (iDorm) which is located at the University of Essex. The iDorm shown in Figure 1 is the test bed for ubiquitous and pervasive computing environments. It looks like any other room and it comprises of a large number of embedded sensors, actuators, processors and a heterogeneous network [9].

The iDorm is multi-user space containing areas for different activities such as sleeping, working and entertaining. It contains the normal mix of furniture, found in a typical student study/bedroom environment and has been fitted with a liberal placement of sensors such as light sensors, temperature sensors, pressure sensors, etc [9].

Figure 1: The iDorm

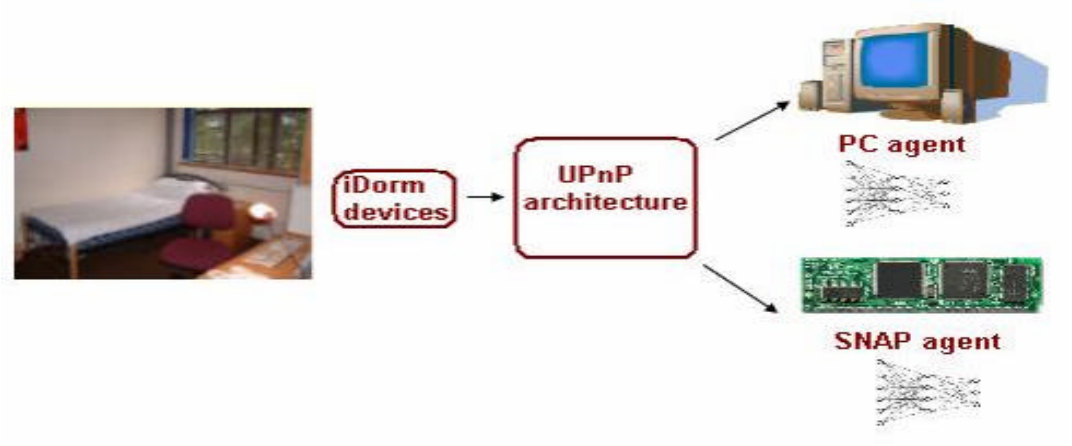

Figure 2: General architecture

A top view of the architecture of the system can be seen in Figure 2. In this Java based embedded-internet devices supporting Universal Plug & Play (UPnP) middleware is used host the agents and provide a communication infrastructure. . UPnP defines two roles of devices: *control points* which act as clients and *controlled devices* which act as servers[3]. Controlled devices are containers which embed services and other controlled devices. Services define the functionality offered by the device and control points use the services to control the device and monitor their status.

In the case of the iDorm, the controlled devices are the sensor and actuator devices (i.e. bed lamp, chair sensor, temperature sensor, etc) and the control point role is performed by a proxy PC running the UPnP stack and providing a user interface which is used to control the devices and to send the information about the state of the sensors and the action performed by the user to the agent. Although a PC is needed at the moment to run the UPnP stack, the next versions of the system will not only implement the agent on a SNAP board, but also the whole UPnP stack.

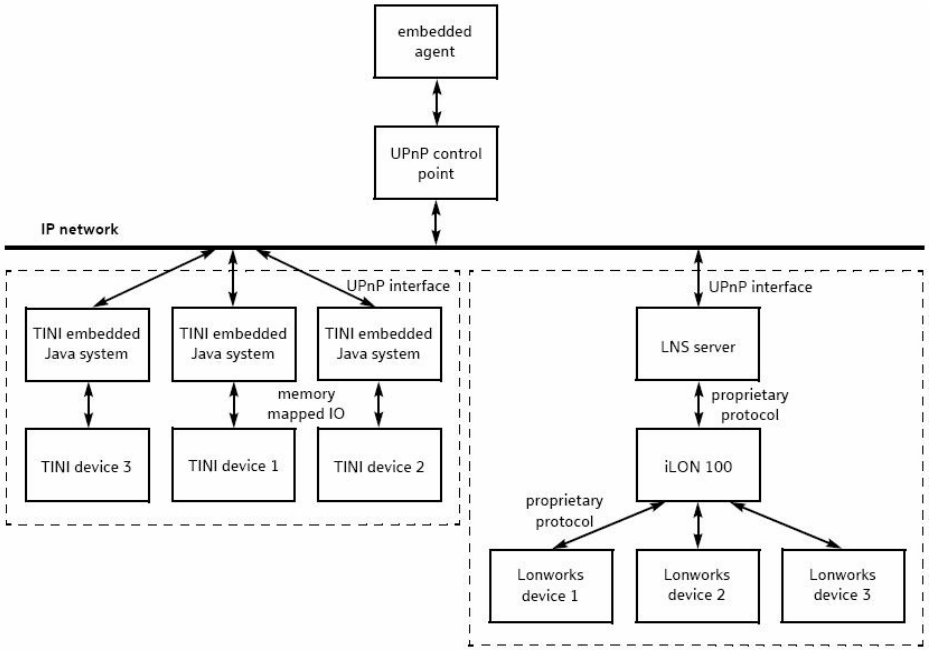

Figure 3: iDorm UPnP architecture

[3] http://www.plug-n-play-technologies.com/index.htm

The iDorm connection architecture is shown in Figure 3. The UPnP architecture wraps the sensors and actuators using an UPnP interface and connects them to the control point using an IP network. UPnP can help to connect a heterogeneous network providing access to the various embedded computing devices we use, such as TINI or SNAP boards and Lonworks devices.

A user interface is needed in order for the user of the room to access the information about the state of the environment, to modify the current state of it, and to provide means by which the current user activity can be queried and recorded to be used later as labels in the classification process. The user interface is shown in Figure 4.

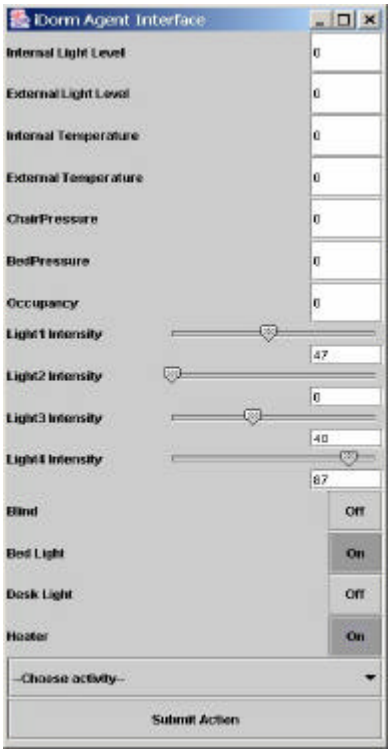

Figure 4: iDorm User Interface

This interface acts as a UPnP control point and it is used to control the iDorm environment and to send the data to the agent in order to the record the user's actions and behaviours. Every time the user wants to perform another activity or wants to change anything in the environment, the user will use the interface and the interface acting as a control point will access the services provided by the devices, changing the state of the environment as needed, sending the information to the agent, and recording the activity being performed by the user at that moment.

4. The Agent Platform

Two different types of agents were programmed using Java. The first one was programmed in a PC and is running the Java 2 Standard Edition (J2SE) of Java, using the JDK version 1.4. The second one was programmed on a SNAP board. The SNAP board is a network-ready, Java-powered plug & play reference platform. It is ideal for remote control, data processing and for managing of different systems ranging from small sensors to advanced factory equipment[4]. This SNAP board has a Cjip microprocessor developed by Imsys. This microprocessor has been designed for networked, Java-based control. It runs at 80 MHz and the SNAP board has 8 Mb DRAM + 2 Mb RAM. It is designed to use the J2ME-CLDC runtime environment, certified by Sun Microsystems. The J2ME (Java 2 Micro Edition) is a very low-footprint Java runtime environment. The CLDC (Connected Limited Device Configuration) was designed to bring the many advantages of the Java platform to connected devices that are limited in available resources. Targeted devices include cellular phones, pagers,

[4] http://www.imsys.se

mobile point-of-sale terminals, and any other device constrained in processing power, memory, and graphical capability[5].

The main purpose of programming an agent for running on a SNAP board was to show that the agent is capable of being embedded on a typical embedded-internet device that is available on intelligent environments. By using this approach, we are able to show that our neural network agent architecture is able to recognize and detect behaviours inside an environment using very limited processor power and memory of such an embedded device. As far as we know, there aren't any implementations of agents able to recognize human behaviours inside an environment that have been fitted inside an embedded processor. Thus, we contend, that this practical demonstration is an important experimental contribution to this field. Several modifications to the original J2SE Java code were needed in order to adapt the agent to the limited J2ME environment. In order to test the SNAP based agent with the iDorm data, an UDP connection was established between the UPnP control point and the SNAP based agent. The information was sent in real time and the agent was trained in an online way with each one of the incoming instances.

5. The Agent Connectionist Architecture

The vast majority of the neural networks have fixed architectures, so when new environmental data is presented, or new inputs/outputs are added, they need to be retrained. Therefore there is a need to develop an algorithm to adapt the hidden layer to any changes in the environmental data or new inputs/outputs. Three different methods have been commonly used: regularisation algorithms, pruning algorithms and constructive algorithms [11]. In constructive algorithms, the size of the hidden layer begins with a small number of units and grows (adding units) until a satisfactory solution is found. We use an Adaptive Neural Architecture derived from the ECoS paradigm proposed by Kasabov et al. [8], [10]. This network can grow dynamically. It does so by adding nodes to the hidden layer (rule nodes) whenever an example is not found to fit in the already existing structure.

The network receives the environmental data from the sensors and adapts itself to it, recognizing and categorizing the different activities and building a model of the "normal" conditions. After the model has been acquired, the system can be applied to monitor the environment. If the system detects a condition that it has never encountered before, it will label it as a novel or abnormal behaviour. The addition of a temporal component is important in order to also detect abnormalities in the sequence, frequency and duration of the activities.

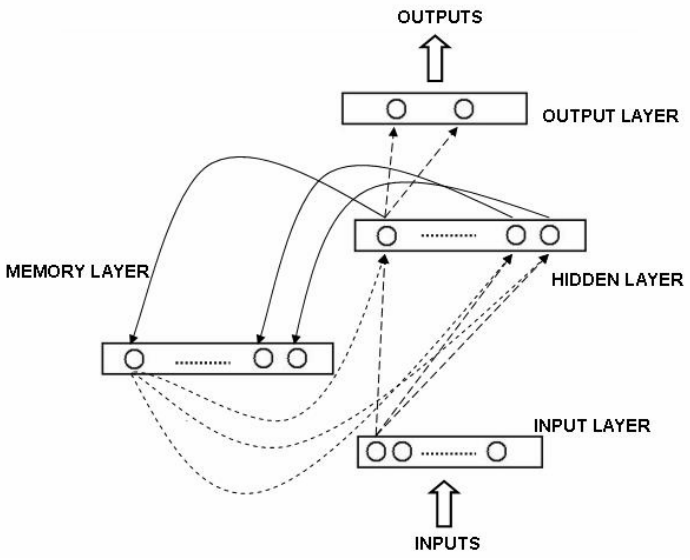

Figure 5: Network structure

[5] http://java.sun.com/products/cldc/

Figure 5 provides a diagram of the actual structure of the adaptive neural network, consisting of three layers: the input layer, the evolving hidden layer and the output layer, plus a memory layer used to represent temporal information. The dashed arrows represent fully connected layers whereas the solid line arrow represent one to one connection between neurons.

The input layer is non-processing, so it only has linear activation functions. For each node in the hidden layer a normalised distance function (a Manhattan distance) is used. The distance function *D*, shown in Equation (1) calculates the distance between the current input vector *I* and the incoming connection weight vector *Wih* (weights between the input and hidden layer) of a particular *hnode* (hidden node).

$$D_{I,hnode} = \frac{\sum_{k=1}^{n} |I_k - Wih_{k,hnode}|}{\sum_{k=1}^{n} |I_k| + \sum_{k=1}^{n} |Wih_{k,hnode}|} \quad where\ n = \#inputs \tag{1}$$

The activation function *A* of each neuron in the hidden layer is calculated using the following equation:

$$A_{hnode} = 1 - D_{I,hnode} \tag{2}$$

Only the winning, or most highly activated hidden neuron, is allowed to activate and the activation of the output neurons O_p, is shown in Equation (3), which applies a simple a product operation over the winning hidden neuron activation *A* and the hidden-to-output layer connection weights *Who*.

$$O_p = A_{hnode} \cdot Who_{hnode,p} \tag{3}$$

The learning algorithm consists of placing the input examples within the hidden (evolving layer) by either creating a new node or modifying the connection weights. The input-to-hidden layer connection weights *Wih* are updated in an unsupervised way using Equation (4) where I_i is the input vector and η_1 is the learning rate.

$$Wih(t+1) = Wih(t) + \eta_1 (I_i - Wih(t)) \tag{4}$$

The hidden-to-output layer connection weights *Who* are updated in a supervised way using Equation (5), where A_h is the activation of the winning hidden node, *Err* is the error between the desired and the calculated output and η_2 is the learning rate.

$$Who(t+1) = Who(t) + \eta_2 (A_h \cdot Err) \tag{5}$$

At the moment, two different methods of abnormality or novelty detection have been explored; they both establish a threshold in order to detect abnormal or novel examples. Although other kinds of novelty detection methods have been used, for example the use of concepts as "habituation" [12], the use of thresholds have been proposed by several authors [11] and these two methods in particular are a modification of the ones proposed in [3].

The first method for abnormality or novelty detection consisted of simply applying a threshold in the hidden layer. If the activation of the winning hidden node is greater than the threshold, the current input example will be labelled as novel.

The second method takes the entire output pattern into account. The Euclidean distance is calculated as follows:

$$d = \sqrt{\sum_{p=1}^{P} (O_p - TO_p)^2} \quad \text{where } P = \#\text{outputs} \qquad (6)$$

Where d represents the distance between this output pattern O and each one of the target patterns TO is used during the training phase and if the smallest of these distances is above a preset threshold then the input pattern is considered as novel.

The network provides an online expansion of the output layer of the network in order to accommodate new classes. The online expansion of the output layer has been used for the abnormality detection process. Although a threshold can be used to detect the abnormalities in the environment, this detection method wasn't really part of the network. With the online creation of new nodes, the network "grows" a new output to accommodate the abnormal examples and trains the network only for those examples, without needing to retrain the whole network. In this way, the abnormality detection is truly embedded within the network structure and the flexibility of the network to be expanded with new outputs/classes is shown at the same time.

The memory layer is a structure that allows the network to have temporal capabilities. Many of the abnormalities that can be detected are not only related with the appearance of new activities but also with the temporal order in which those take place. The architecture of the recurrent network is similar to the one proposed in [6].

The network has feedback connections from the hidden layer of neurons back to the same nodes, and these context units simply hold a copy of the activations of the hidden units from the previous time step. These context units or memory layer allow the network to capture temporal dependencies between the presented data examples from the data stream.

6. Experiments and results

An *activities* dataset was built by using the data collected by 18 sensors (both environmental and furniture based) over a period of 4 days. 8 different activities were detected inside the environment. To assist this process, during the experiments, the user was asked to describe the action he was performing via a simple user interface.

An agent monitored the user's actions inside the environment over a period four days. Whenever the user changed the state of the environment, the agent recorded a 'snapshot' of the current inputs (sensor states). A total of 471 instances were recorded. The data collected inside the iDorm provided the initial set of experiments in which our agent has been tested so as to initially show that this type of technologies can be employed for activity and behaviour recognition and offers a possible solution to the provision of care services at home

At the moment three different types of experiments have been conducted. Firstly, the system has been tested in order to asses its performance for activity recognition tasks. Secondly, other experiments have been done in order to test the abnormality detection process, working with two different methods to spot the abnormalities. Lastly, a set of experiments were conducted to test the addition of a temporal layer and its impact in the detection of abnormalities. In subsection (6.1) we will present the results associated with activity recognition and in subsection (6.2), we will show the results related to abnormality detection. Subsection (6.3) will present the abnormality detection process using a network with a memory layer.

6.1 Activity recognition results

The first step towards the detection of abnormal activities inside an environment is to detect and recognize a normal set of activities. Our network was tested in order to assess its performance for activity recognition. The activities dataset was divided into 4 partitions that corresponded to the data of each one of the 4 days. This data was divided in training and test sets using the *stratified fourfold cross-validation* method. The results of the 4 fold tests and the confusion matrix for the testing phase of the run with a higher recognition rate are shown in the Table (1) and Table (2).

	Train	Test
1st run	87.81%	74.57%
2nd run	87.35%	76.42%
3rd run	86.63%	86.59%
4th run	88.75%	76.69%
Average	**87.64%**	**78.57%**

Table 1 Results of activity recognition test

		Predicted class							
		A = Other	B = Listen music	C =Computer work	D = Reading	E = Desk work	F = Resting Napping	G = Out of room	H = Sleeping
Actual class	A	1	0	0	1	0	0	0	0
	B	1	6	0	0	0	1	0	0
	C	0	0	18	0	6	0	0	0
	D	0	0	1	16	0	0	0	1
	E	0	0	0	0	14	0	0	0
	F	0	1	0	0	0	4	0	0
	G	0	0	0	0	0	1	23	0
	H	0	0	0	0	0	0	0	2

Table 2 Confusion matrix for activity recognition

From the confusion matrix, it can be seen that due to the limited number of sensors installed in the environment, activities such as "Desk work" and "Computer work" are not classified as correctly as activities such as "Out of room", where the sensor data provides more information. One possible way to enhance the recognition rate of the system, besides tuning the threshold and transfer functions, would be the installation of more sensors inside the environment, as they can provide more environmental information allowing the network to distinguish and classify the activities with a finer granularity level.

6.2 Abnormality Detection Results

All the following tests were performed using *stratified fourfold cross-validation* where the 4 partitions corresponded to the data of each one of the 4 days.

Two different trials were conducted, each one of them using a different method in order to find abnormalities. The first method (Detection A in the table) used the threshold placed in the hidden layer and the abnormal examples are stored in a separate structure. The second method (Detection B in the table) implements the new way to detect abnormalities in which the network can "grow" a new output/class node to represent that abnormal example.

The training and test sets where built as follows: In both tests one activity (for these experiments, the "Out of room" activity) was left out during the training phase, that is, the training set didn't have any instances corresponding to that class. For the test phase, instances of all the activities were present, and the network tried to spot as abnormal the activity previously removed during the training phase.

The parameters of the network where set as follows: learning rate = 0.1, sensitivity threshold = 0.90, error threshold = 0.10, anomaly sensitivity threshold = 0.88.

	Detection A	Detection B
1st run	88.98%	77.12%
2nd run	82.11%	86.99%
3rd run	90.72%	86.60%
4th run	94.74%	96.99%
Average	89.14%	86.93%

Table 2 Abnormality detection with two different methods

The threshold in the hidden layer (detection method A) gives better results, however, the detection method B provides a more elegant solution to the problem, because the abnormality detection is embedded into the structure of the network and proves that the network can be expanded as new examples are presented or it can add new classes to accommodate the abnormal instances.

More extensive analysis and comparisons between the approaches used in this work and the ones used for other works in the abnormality detection field can be found in [18]. The results show that our system performs as good as other more complex systems, uses much less time to be trained has the advantage to be trained online and it can be incrementally adapted to new information.

6.3 Results of Abnormality Detection using a network with a memory layer

These experiments were conducted as the first step in order to address the problem of the temporality relationships in the network. Two versions of the network were tested one without the memory layer and the other with it. The aim of the work was to detect abnormalities occurring due to activities that have not been previously seen by the network. The results are summarized below.

	Without Memory	With Memory
1st run	88.98%	94.92%
2nd run	82.11%	79.67%
3rd run	90.72%	93.81%
4th run	94.74%	94.74%
Average	89.14%	90.79%

Table 3 Abnormality detection related to the use of a memory layer

As it can be seen from table 3, the network using a memory layer shows better results. The addition of a memory layer gives the network the ability to spot a larger variety of abnormalities by capturing the temporal dependencies of the data. Although at this moment the results don't show a big difference between the two approaches, it must be remembered that these are the first experiments testing the temporal capabilities of the network to detect abnormalities, and the only abnormalities tested were the ones in which an activity not previously seen before was introduced in the testing phase. Nevertheless, it is interesting to see that in some cases the use of a temporal approach can improve the results by almost 6%.

More experiments are to be conducted in which the network will be tested not only with activities not previously seen, but with activities that have been previously seen but differ in sequence and frequency. Those experiments will present more conclusive results and are expected to prove the full benefits of adding a memory layer.

7. Conclusions and future work

In this paper, we have presented a novel solution to care at home using a connectionist agent based approach. The system combines the use of unobtrusive and relatively simple sensors with the application of a connectionist agent running on an embedded device. High level activities such as "sleeping", "working at computer", "eating", etc can be characterized with the use of only simple sensors and environmental data. Finer granularity levels in the spotted activities and behaviours are expected to be achieved in the future by increasing the size of our test facilities, by further tuning of thresholds and transfer functions and by installing a larger array of sensors.

To our knowledge, there aren't any mechanisms that can be embedded into processors of the size used in this (80 Mhz, 8+2 Mb RAM) that are able to recognize human behaviours inside a domestic environment. Thus, an agent able to do run in such a constrained platform is an important contribution of our work. Although at the moment a PC is still used to run the UPnP stack, it is possible to run UPnP inside the SNAP board and that will be implemented in the next version of the system.

In addition, the system is also able to learn in an online (one-pass) and incremental way, adapting itself to new data as they are made available over time and it can be applied to environments with changing dynamics. Our system is able to perform as good as more complex systems, needing only to be trained for one epoch (the examples need to be presented only once to the network in order to train it), uses an abnormality detection method that is embedded into the structure of the network and proves that the network can be expanded as new examples are presented or it can add new classes to accommodate the abnormal instances.

This paper is reporting on research in progress, so more work will needed in order to fully test the system, as well as its ability to spot abnormalities related to sequence and timing of life activities and the impact of the thresholds in the system's performance. The first results of the use of the network with a temporal layer are encouraging and have shown that the abnormality recognition process benefits from the use of the temporal components by allowing the network to keep an internal memory.

References

[1] K. Adair, and P. Argo, "Classification of Behavior Using Unsupervised Temporal Neural Networks", Proc. 1997 IEEE International Conference on Systems, Man, and Cybernetics, pp. 2584-2589, October 1997.

[2] S. Allin, A, Bharucha, J. Zimmerman, D. Wilson, M. Roberson, S. Stevens, H. Wactlar, and C. Atkeson, "Toward the Automatic Assessment of Behavioral Disturbances of Dementia", UbiHealth 2003, Seattle, 2003

[3] M.F. Augusteijn, and B.A. Folkert, "Neural network classification and novelty detection", *International Journal of Remote Sensing*, 1999.

[4] C. Campbell, and K. Bennet, "A linear programming approach to novelty detection", Proc. of Advances in Neural Information Processing Systems 13 (NIPS'00), Cambridge, MA, 2000.

[5] F. Doctor, H. Hagras, and V. Callaghan, "An Fuzzy Embedded Agent Based Approach for Realising Ambient Intelligence in Intelligent Inhabited Environments", to appear in *IEEE Trans. On Systems, Man and Cybernetics, Part A: Systems and Humans*.

[6] Elman, J. L. : "Finding structure in time", Cognitive Science, vol. 14(2), pp. 179-211, 1990.

[7] K. Fishkin, H. Kautz, D. Patterson, M. Perkowitz, and M. Philipose, "Guide: Towards Understanding Daily Life via Auto-Identification and Statistical Analysis", UbiHealth 2003. Seattle, WA, Oct 12, 2003.

[8] A. Ghobakhlou, M. Watts, and N. Kasabov, "Adaptive speech recognition with evolving connectionist systems", *Information Sciences*, 156, pp. 71-83, 2003.

[9] H. Hagras, M. Colley, V. Callaghan, G. Clarke, and H. Duman, "A Fuzzy Incremental Synchronous Learning Technique for Embedded Agents Learning and Control in Intelligent Inhabited Environments", Proc 2002 IEEE Int'l Conf on Fuzzy systems, Hawaii, pp. 139-145, 2002.

[10] N. Kasabov, *Evolving connectionist systems: Methods and applications in bioinformatics, brain study and intelligent machines*, London: Springer, 2002.

[11] M. Markou, and S. Singh, "Novelty detection: a review—part 2: Neural network based approaches", *Signal Processing*, 83, pp. 871-888, 2003.

[12] S. Marsland, J. Shapiro, and U. Nehmzow, "A Self-organising Network that Grows When Required", *Neural Networks*, 15(8-9), pp. 1041-1058, 2002.

[13] A. Mihailidis, B. Carmichael, J. Boger, and G. Fernie, "An Intelligent Environment to Support Aging-in-Place, Safety, and Independence of Older Adults with Dementia", Ubi-Health 2003. Seattle, WA, Oct 12, 2003.

[14] M. Mozer, "The Neural Network House: An Environment that Adapts to its Inhabitants", Proc. of the AAAI Spring Symposium on Intelligent Environments, pp. 110--114, 1998.

[15] E. Munguia, S. Intille, and K. Larson, "Activity Recognition in the Home Using Simple and Ubiquitous Sensors", Pervasive Computing, Second International Conference, PERVASIVE 2004, Vienna, Austria, pp. 158-175, April 21-23, 2004.

[16] E. Mynatt, and W. Rogers, "Developing technology to support the functional independence of older adults" Ageing International, 2002

[17] D. Patterson, L. Liao, D. Fox, and H. Kautz, "Inferring High Level Behavior from Low Level Sensors", Fifth Annual Conference on Ubiquitous Computing (UBICOMP 2003), Seattle, WA, 2003.

[18] F. Rivera-Illingworth, V. Callaghan, H.Hagras, "A Connectionist Embedded Agent Approach for Abnormal Behaviour Detection in Intelligent Health Care Environments", 2004 IEEE International Conference on Systems, Man & Cybernetics, The Hague, The Netherlands. 2004.

[19] G. Vasconcelos, M. Fairhurst, and D. Bisset, "Investigating feedforward neural networks with respect to the rejection of spurious patterns", *Pattern Recognition Letters*, vol.16, pp. 207-212, 1995.

[20] K. Zita Haigh, and H. Yanco, 2002. "Automation as Caregiver: A Survey of Issues and Technologies", AAAI-02 Workshop on Automation as Caregiver: The Role of Intelligent Technology in Elder Care, pp. 39-53. July 2002

