Fuzzy Spiking Neural Network for Abnormality Detection in Cognitive Robot Life Supporting System

Dalai Tang*, Tiong Yew Tang†, János Botzheim*, Naoyuki Kubota*, Toru Yamaguchi*
*Graduate School of System Design, Tokyo Metropolitan University,
6-6 Asahigaoka, Hino, Tokyo, 191-0065, Japan
Email: tang@ed.tmu.ac.jp, {botzheim, kubota, yamachan}@tmu.ac.jp
†School of Information Technology, Monash University Malaysia,
Jalan Lagoon Selatan, 47500 Bandar Sunway, Selangor Darul Ehsan, Malaysia
Email: tang.tiong.yew@monash.edu

***Abstract*—In aging nation such as Japan, elderly people belong to the vulnerable group that constantly need healthcare and monitoring for their well-being. Therefore, an early warning system for detecting abnormality in their daily activities could save their life (e.g. heart attack, stroke and etc.). However, such early warning system must not trigger any false warning signals in order to robustly operate in real world applications. Robot interactions with human are useful to prevent false warning signals from sending out to healthcare worker. Next, the system should be able to detect short-term abnormal and also long-term abnormal behaviors of the elderly people within their normal daily life routine. Therefore, it is important to integrate informationally structured space with cognitive robot to confirm the elderly's abnormal situation with human-robot interactions before sending out warning signals to healthcare workers. In this work, we proposed an evolutionary computation based approach to optimize fuzzy spiking neural network for detecting abnormal activities in the elderly people's daily activities.**

I. Introduction

In aging nation such as Japan, according to statistical research [1], elderly people population percentage will reach 23.8% out of the total population in Tokyo in the year 2015. Therefore, elderly people's well-being is a major concern in such developed nation because large percentage of the population's lives are at stake. It is absolute essential to detect any health-related abnormal symptoms in advance to inform the healthcare personnel for their further actions. The reason is, that early warning signals could save the elderly people's life. However, such early warning signals should be accurate and precise so that healthcare personnel can act correspondingly without wasting their resources. The main reason is the emergency healthcare personnel resources are very vital therefore any false warning signals triggered are not tolerated. It is important not to dispatch any emergency healthcare personnel resources to false warning signals. The reason behind is other real emergency need of healthcare resources may happen at the same period of time. Hence, it is ideal to have a robot partner to confirm the elderly people's health status with human-robot interaction before sending out any warning signals. The human-robot interaction is also needed to prevent noise intervention from the environment that influences the sensor readings that could cause the false warning signal.

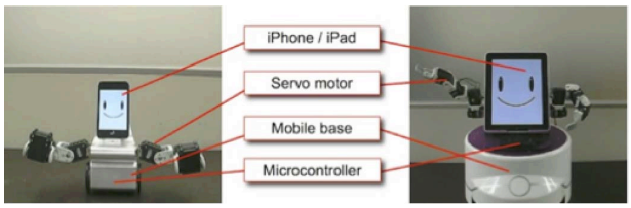

Fig. 1. The robot on the left is iPhonoid and the robot on the right is iPadrone.

The number of smart phone and tablet PC devices is becoming more ubiquitous in recent years. These smart devices are equipped with multiple sensors, wireless communication system, fast CPU and they are available at consumer affordable price. The processing power of these smart devices consists of multiple cores CPU with consumer affordable power consumption capability. Therefore, it is strategic to use smart devices as a robot partner processing unit to support the human-robot communication with elderly people. Hence, we started the robot application project on small sized tabletop robot partners called *iPadrone* and *iPhonoid* based on smart devices for information support for the elderly people (See Fig. 1).

Modern network technologies enable ubiquitous network access to wireless sensors, so that useful and timely information is available through wireless sensor networks. It is important to leverage such information from the sensor network in an early warning system. *Sensor network information service* is used for data mining and structuralization of information on user's daily activities based on machine learning techniques. Then, the information gathered on the user should be handled without the load of the user's effort. Hence, the ideal information service overhead processing should be kept to minimum as possible. Therefore, the human behaviors and location information can be timely extracted from the sensor network

978-1-4799-7560-0/15 $31.00 © 2015 IEEE
DOI 10.1109/SSCI.2015.29

information service. Sensor network information service access to information resources is essential for both people and robots. Thus, the environment should be able to provide timely information for the elderly people and robots to facilitate their conversation. Sensor network information service is a structured platform for gathering, storing, transforming, and providing information. The information gathered will form a virtual environment known as *informationally structured space* [2], [3].

The informationally structured space is an environment that formats the information into structures, provides information visualization, and timely modification and access of essential information for elderly people early warning system. Therefore, the informationally structured space will provide vital information for detecting any difficulties faced by the elderly people.

II. Environment Aware Cognitive Robot Life Supporting System

Previously, abnormality detection sensor systems were proposed, for example by Van Laerhoven et al. [4] and by John Kemp et al. [5]. These approaches were used to measure abnormal activities in the environment but noise could cause these systems to create false warning signal. Therefore, it is important to have human-robot communication before identifying the situation as abnormal situation. In our proposed abnormality detection system, the system will detect two different types of abnormal conditions in the elderly people's daily activities: 1) The system will detect the short-term abnormal activities. For example, if an elderly person get a heart attack or stroke, he or she needs immediate healthcare for stabilizing his or her condition. 2) The system also will detect long-term abnormal activities where the elderly people daily activities pattern changes. For example, the elderly people frequently sleep in late hours. The long-term abnormal activities are important information for early prevention of any major health problem. Medical health consultant can act on the early warning on elderly people daily life activities pattern changes.

After an abnormal activity is detected, our proposed abnormality detection system will trigger the robot partner to ask questions to the elderly people. For example, the robot will ask the elderly people questions such as "Are you okay?", "Do you need to call for help?".

A. Fuzzy Spiking Neural Network for Abnormality Detection

Our model predicts the abnormal activity of the elderly people by fuzzy spiking neurons. One of the vital properties of spiking neurons is the temporal coding feature. In addition to that, many different types of spiking neural networks (SNN) have been applied for remembering temporal and spatial context [6]–[8]. In order to reduce the computational cost, a simple spike response model is used. The SSN is composed of fuzzy inputs values: it is a fuzzy spiking neural network (FSNN) [9]–[11]. For this research, we will detect the abnormal elderly people activities. Fig. 2 illustrates the FSNN model. Evolution

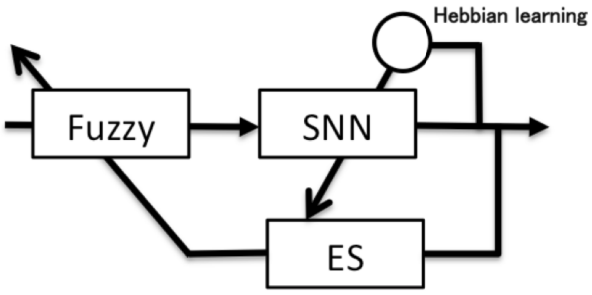

Fig. 2. Fuzzy spiking neural network model

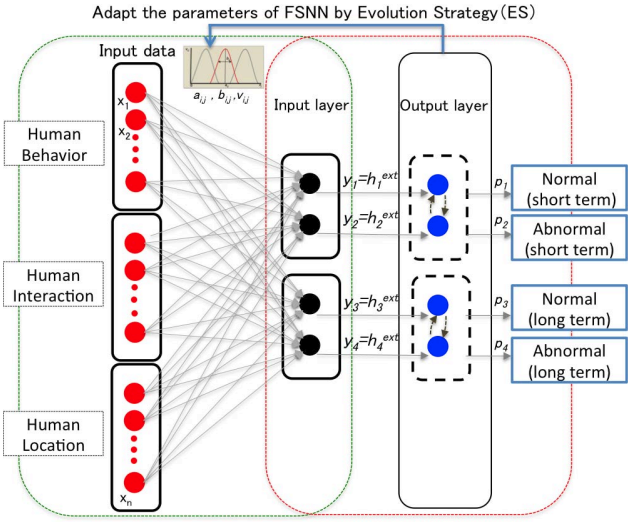

Fig. 3. Structure of the FSNN

TABLE I
HUMAN BEHAVIOR

No.	Human behavior
0	sleep
1	get up
2	rest
3	go to toilet
4	have breakfast
5	have lunch
6	have dinner
7	get home
8	out home

TABLE II
LOCATION

No.	Location
0	outdoor
1	bedroom
2	living room
3	kitchen
4	toilet
5	entrance

TABLE III
HUMAN-ROBOT INTERACTION

No.	Interaction
0	no interaction
1	OK
2	not good

strategy (ES) is used to adapt the parameters of the fuzzy membership functions applied as inputs to the spiking neural network. Furthermore, Hebbian learning [12] is used to modify the weights between the spiking neurons. The structure of the model is depicted in Fig. 3. In this proposed FSNN model, the model has two different spiking neural network modules. One is applied for the short-term abnormal activities detection (the upper boxes) and the other one is applied for the long-term abnormal activities detection (the lower boxes). The inputs of the model are presented in Tables I, II, and III.

The input fuzzy inference sensory information from the informationally structured space will be performed by:

$$\mu_{A_{i,j}}(x_j) = \exp\left(-\frac{(x_j - a_{i,j})^2}{b_{i,j}}\right) \tag{1}$$

$$y_i = \prod_{j=1}^{m} v_{i,j} \cdot \mu_{A_{i,j}}(x_j) \tag{2}$$

Where $a_{i,j}$ is the middle value and $b_{i,j}$ is the width of the membership function. Next, $v_{i,j}$ and $A_{i,j}$ is the part of the j-th input to the estimation of the i-th human state. The y_i is the product of fuzzy inference and it is also the input of the spiking neurons. The internal state $h_i(t)$ of the i-th spiking neuron (also known as membrane potential) at the discrete time t is defined by:

$$h_i(t) = \tanh(h_i^{syn}(t) + h_i^{ext}(t) + h_i^{ref}(t)), \tag{3}$$

where, $h_i^{ext}(t)$ (computed by Eq. (4) is from the external environment input influences to the i-th neuron, $h_i^{syn}(t)$ (calculated in Eq. (5) incorporated the pulses from the other fully connected neurons outputs, $h_i^{ref}(t)$ (shown in Eq. (7)) is used for representing the refractoriness of the neuron. The hyperbolic tangent function is used to avoid the bursting of neuronal fires.

$$h_i^{ext}(t) = \prod_{j=1}^{m} v_{i,j} \cdot \exp\left(-\frac{(x_j - a_{i,j})^2}{b_{i,j}}\right) \tag{4}$$

In Fig. 3, $h_i^{syn}(t)$ is presented by dashed arrows.

$$h_i^{syn}(t) = \gamma^{syn} \cdot h_i(t-1) + \sum_{j=1, j\neq i}^{n} w_{j,i} \cdot h_j^{PSP}(t-1), \tag{5}$$

where γ^{syn} is a temporal discount rate, n is the number of spiking neurons, and m in Eq. (4) is the total number of inputs.

When the internal state of the i-th neuron reaches a predefined threshold level, a pulse is outputted as follows:

TABLE IV
CONFUSION MATRIX FOR A TWO-CLASS PROBLEM

	Positive prediction	Negative prediction
Positive class (abnormal data)	true positive (TP)	false negative (FN)
Negative class (normal data)	false positive (FP)	true negative (TN)

$$p_i(t) = \begin{cases} 1 & \text{if } h_i(t) \geq \theta, \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

where θ is a threshold for firing. In case of firing, R is subtracted from the $h_i^{ref}(t)$ value of neuron i as follows:

$$h_i^{ref}(t) = \begin{cases} \gamma^{ref} \cdot h_i^{ref}(t-1) - R & \text{if } p_i(t-1) = 1, \\ \gamma^{ref} \cdot h_i^{ref}(t-1) & \text{otherwise,} \end{cases} \tag{7}$$

where γ^{ref} is a discount rate of h_i^{ref} and $R > 0$. The presynaptic spike output is transmitted to the connected neuron through the weight connection. The PSP is calculated as follows:

$$h_i^{PSP}(t) = \begin{cases} 1 & \text{if } p_i(t) = 1, \\ \gamma^{PSP} \cdot h_i^{PSP}(t-1) & \text{otherwise,} \end{cases} \tag{8}$$

where γ^{PSP} is a discount rate of h_i^{PSP}.

B. Evolution Strategy for Optimizing the Parameters of FSNN

In this work, we incorporate (μ+λ)-Evolution Strategy (ES) to optimize the fuzzy spiking neural network parameters in fuzzy membership functions. In ($\mu + \lambda$)-ES μ and λ then state the number of parents and the number of offspring produced in an evolution generation correspondingly [13]. We apply ($\mu + 1$)-ES for improving the local hill-climbing search as a continuous model of generations, which terminates and initializes one individual in one evolution generation. The ($\mu + 1$)-ES can be considered as a steady-state genetic algorithm (SSGA) [14].

A candidate solution contains the parameters of the fuzzy membership functions:

$$\boldsymbol{g}_k = \begin{bmatrix} g_{k,1} & g_{k,2} & g_{k,3} & \cdots & g_{k,l} \end{bmatrix} \tag{9}$$

$$= \begin{bmatrix} a_{k,1,1} & b_{k,1,1} & v_{k,1,1} & \cdots & v_{k,n,m} \end{bmatrix} \tag{10}$$

Where n is the number of spiking neurons; m is the total number of inputs; $l = n \cdot m$ is the chromosome length of the k-th candidate solution.

The amount of abnormal data is much smaller than the amount of normal data in general. It is imbalanced data. In this scenario, if we only use the accuracy information to evaluate the GA result, then it would not provide good information. The reason is that we also have to consider comprehensiveness for the evaluation of the result. Therefore we apply F-measure metric for the fitness value evaluation. Table IV shows the

confusion matrix for a two-class problem. The fitness value of the k-th candidate solution is calculated as:

$$f_k = \frac{2P_k R_k}{(P_k + R_k)} \tag{11}$$

where P_k, R_k are precision and recall of the k-th candidate solution.

Precision is calculated as:

$$P_k = \frac{TP}{(TP + FP)} \tag{12}$$

Recall is calculated as:

$$R_k = \frac{TP}{(TP + FN)} \tag{13}$$

where TP, FP and FN is the total number of true positive, false positive, and false negative cases in the k-th candidate solution, respectively.

The number of correct estimation rates is calculated as:

$$c_k = \sum_{i=1}^{n} c_{k,i} \tag{14}$$

where $c_{k,i}$ is the number of correct estimation rates of the i-th neuron. We compare each output of the FSNN in the time sequence with the corresponding desired output. If the FSNN's output is the same as the desired output, then we count this as a matching condition. The number of matchings for the i-th neuron is $c_{k,i}$. In $(\mu + 1)$-ES, only one existing solution is replaced with the candidate solution generated by crossover and mutation. We use elitist crossover and adaptive mutation. Elitist crossover randomly selects one individual, and generates one individual by combining genetic information between the selected individual and the best individual in order to obtain feasible solutions from the previous estimation result rapidly. The newly generated individual replaces the worst individual in the population after applying adaptive mutation on the newly generated individual. The inheritance probability of the genes corresponding to the i-th rule (i-th spiking neuron) of the best individual is calculated by:

$$p_i = \frac{1}{2} \cdot (1 + c_{best,i} - c_{k,i}) \tag{15}$$

where $c_{best,i}$ and $c_{k,i}$ are the number of correct estimation rates of the best individual and the randomly selected k-th individuals, respectively. With Eq. (15) we can bias the selection probability of the i-th genes from 0.5 to the direction of the better individual's i-th genes between the best individual and the k-th individual. Thus, the newly generated individual can inherit the i-th genes (i-th rule) from that individual which the better i-th gene has. After the crossover operation, an adaptive mutation is performed on the generated individual:

$$g_{k,h} \leftarrow g_{k,h} + \alpha_h \cdot (1 - t/T) \cdot N(0,1) \tag{16}$$

where $N(0,1)$ indicates a normal random value; α_h is a parameter of the mutation operator (h stands for identifying the three subgroups in the individual related to a, b, and v); t is the current generation; and T is the maximum number of generations. The computational cost of ES is generation size + population size, $T + P$.

C. Hebbian Learning

As illustrated in Fig. 2, besides optimizing the parameters of the fuzzy membership functions by evolution strategy, the weights between the spiking neurons are modified by Hebbian learning. The weights can be adjusted dynamically in the simulation process. If the condition $0 < h_j^{PSP}(t-1) < h_i^{PSP}(t)$ is satisfied, the weight parameter, $w_{j,i}$ is trained based on the Hebbian learning rule [12]:

$$w_{j,i} \leftarrow \tanh\left(\gamma^{wgt} w_{j,i} + \xi^{wgt} h_j^{PSP}(t-1) h_i^{PSP}(t)\right), \tag{17}$$

where γ^{wgt} is a discount rate of the weights and ξ^{wgt} is a learning rate.

III. Experiments

This section shows comparison results and analyzes the performance of the proposed method. The graph in Fig. 4 is the input data description; there are 18 inputs: 9 human behavior inputs, 6 human location inputs, and 3 human interaction inputs. In the proposed structure there are 2 SNN modules: the short- and the long-term modules. In the output layer there are 4 outputs: the short-term normal, short-term abnormal, long-term normal and the long-term abnormal output. 22 days training data set and 2 days test data set were used in the experiments. The parameters of the neural network are presented in Table V, where S stands for short-term and L stands for long-term. Figure 4 shows a simulation window of the experiment. The upper window shows 3 different kinds of input data: the human behavior, the human location and the human interaction input data. The blue line is the input data from life log; the red line is the PSP output for the SNN's input layer. The lower window shows the human state estimation process by the SNN module. The blue line is the input data of each neuron node; the red line is the internal state; the green line is the training data; the purple line is the estimation result.

TABLE V
Parameters of the Neural Network

	M	N	γ^{syn}	γ^{ref}	γ^{psp}	R	θ	γ^{wgt}	ξ^{wgt}
S	18	2	0.59	0.95	0.40	1	0.80	1.0	0.05
L	18	2	0.70	0.90	0.90	1	0.99	1.0	0.8

In the long-term data experiment, we use adapted parameters from the training experiments in order to estimate the test data. Table VI shows the experimental result for training datasets and test datasets. In this case, we calculate the average based on 10 simulation experiments for each data. For training data we also calculated the standard deviation. The number of total training data is 31680, with 2059 abnormal and 29621 normal data. The number of total test data is 2880, with 108 abnormal and 2772 normal data. The test data experiment is as follows. Figure 5 illustrates the input of long-term test data

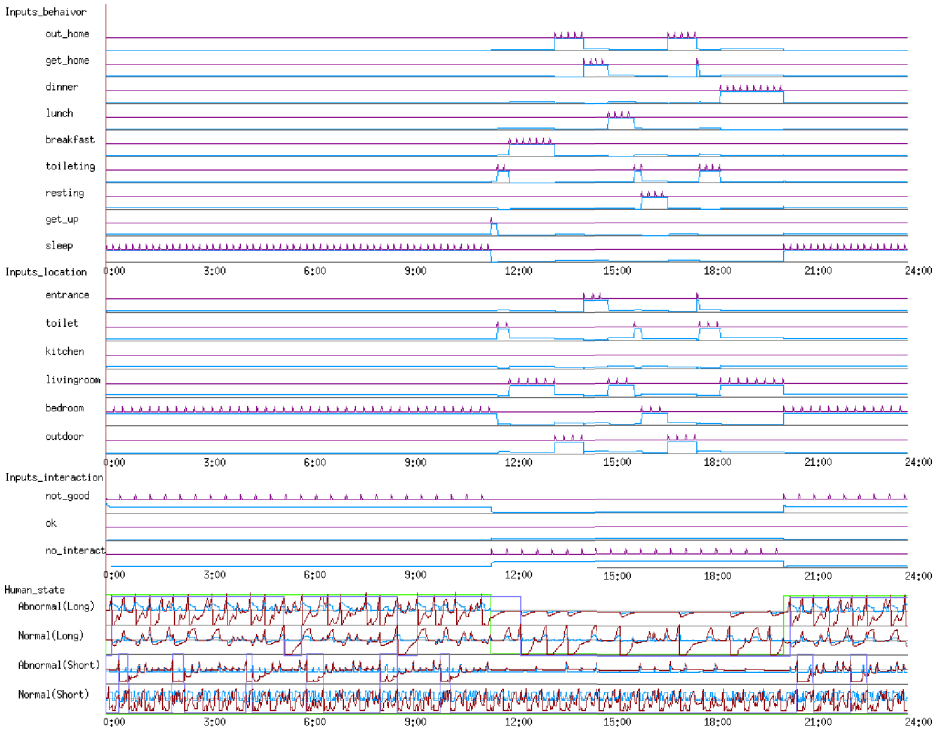

Fig. 4. Simulation window of the experiment

in one-day human activity. There is one abnormal state when the human spends quite a long time in the toilet after lunch, and during lunch time the interaction is "not good".

Figure 6 shows the estimated result by using SNN. We can understand, that the estimation result (purple line) is nearly at dose moreover, it does not match the teaching data (green line), and the F-measure, accuracy and fitting rate are not good as well. The reason is, that the feature of SNN's input data does not fit the output state. Figures 7 and 9 show the result by using GA to update the SNN parameters, where T is the number of generations and P is the population size. According to the many times experiment by applying GA, we knew that the experimental result does not change when $T = 20000$ and $P = 500$. Because of this, we present experiment result here when $T = 2000$ $P = 200$ (Figure 7) and $T = 20000$, $P = 500$ (Figure 9). We can see, that the estimation result (purple line) is almost matching the teaching data (green line), when the number of generations and the population size are increasing. The F-measure, accuracy and fitting rate became better than applying only SNN. With the use of GA, we are able to optimize the parameter of membership function for SNN's input data, and it follows, that the feature of SNN's input data fits the output state. Figures 8 and 10 show the application of GA and Hebbian learning. F-measure, accuracy and fitting rate have not improved that much and the result is not as better as applying SNN and GA. We will explain the reason in short-term result.

In the experiment for short-term test data, we use adapted

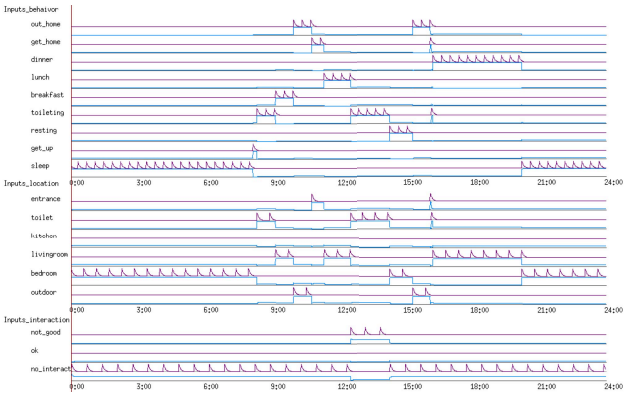

Fig. 5. Input for long-term test data

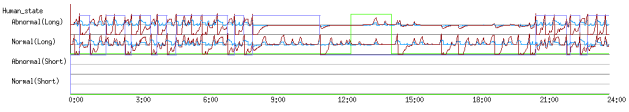

Fig. 6. Experimental result by using SNN for long-term test data

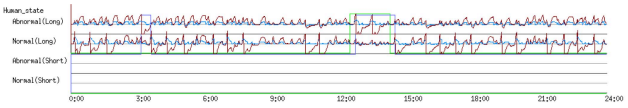

Fig. 7. Experimental result after using GA for long-term test data (T=2000, P=200)

TABLE VI
LONG-TERM EXPERIMENTAL RESULT

	F-Measure	Accuracy	Fitting rate (abnormal)	Fitting rate (normal)	F-Measure (Hebbian)	Accuracy (Hebbian)	Fitting rate of abnormal (Hebbian)	Fitting rate of normal (Hebbian)
Training data								
SNN only	0.251	0.694 (21987/31680)	0.835 (1719/2059)	0.685 (20285/29621)	0.115	0.081 (2567/31680)	0.970 (1997/2059)	0.023 (677/29621)
After GA T=2000 P=200	0.679 ±0.148	0.965 ±0.080 (30583/31680)	0.636 ±0.200 (1310/2059)	0.987 ±0.088 (2923/29621)	0.683 ±0.183	0.977 ±0.253 (30951/31680)	0.697 ±0.183 (1434/2059)	0.978 ±0.275 (28960/29621)
After GA T=8000 P=200	0.713 ±0.105	0.977 ±0.008 (30951/31680)	0.692 ±0.215 (1424/2059)	0.987 ±0.010 (29226/29621)	0.773 ±0.079	0.977 ±0.010 (30951/31680)	0.837 ±0.156 (1724/2059)	0.981 ±0.007 (29063/29621)
After GA T=20000 P=500	0.792 ±0.112	0.977 ±0.007 (30951/31680)	0.816 ±0.182 (1680/2059)	0.984 ±0.006 (29133/29621)	0.958 ±0.075	0.989 ±0.004 (30951/31680)	0.923 ±0.058 (1839/2059)	0.991 ±0.002 (29040/29621)
Test data								
SNN only	0	0.583 (1679/2880	0 (0/108)	0.606 (1679/2772)	0	0.583 (1679/2880	0 (0/108)	0.606 (1679/2772)
After GA T=2000 P=200	0.512	0.993 (2589/2880	0.861 (93/108)	0.996 (2760/2772)	0.868	0.990 (2852/2880	0.852 (92/108)	0.996 (2760/2772)
After GA T=8000 P=200	0.835	0.988 (2846/2880	0.796 (86/108)	0.996 (2760/2772)	0.874	0.991 (2853/2880	0.870 (94/108)	0.995 (2759/2772)
After GA T=20000 P=500	0.869	0.990 (2852/2880	0.861 (93/108)	0.995 (2759/2772)	0.883	0.991 (2855/2880	0.870 (94/108)	0.996 (2761/2772)

TABLE VII
SHORT-TERM EXPERIMENTAL RESULT

	F-Measure	Accuracy	Fitting rate of abnormal	Fitting rate of normal	F-Measure (Hebbian)	Accuracy (Hebbian)	Fitting rate of abnormal (Hebbian)	Fitting rate of normal (Hebbian)
Training data								
SNN only	0.167	0.984 (31161/31680)	0.510 (52/102)	0.985 (31109/31578)	0.167	0.984 (31161/31680)	0.510 (52/102)	0.985 (31109/31578)
After GA T=2000 P=200	0.818 ±0.206	0.999 ±0.004 (31639/31680)	0.870 ±0.123 (89/102)	0.999 ±0.004 (31550/31578)	0.827 ±0.206	0.999 ±0.0004 (31646/31680)	0.844 ±0.166 (86/102)	0.999 ±0.004 (31560/31578)
After GA T=8000 P=200	0.847 ±0.053	0.999 ±0.001 (31646/31680)	0.888 ±0.026 (91/102)	0.999 ±0.001 (31556/31578)	0.869 ±0.206	0.999 ±0.004 (31653/31680)	0.892 ±0.123 (91/102)	0.999 ±0.004 (31562/31578)
After GA T=20000 P=500	0.855 ±0.044	0.999 ±0.000 (31649/31680)	0.883 ±0.017 (90/102)	0.999 ±0.000 (31559/31578)	0.877 ±0.011	0.999 ±0.000 (31654/31680)	0.912 ±0.020 (93/102)	0.999 ±0.000 (31561/31578)
Test data								
SNN only	0.051	0.974 (2806/2880)	0.143 (2/14)	0.978 (2804/2866)	0.051	0.974 (2806/2880)	0.143 (2/14)	0.978 (2804/2866)
After GA T=2000 P=200	0.373	0.987 (2843/2880)	0.786 (11/14)	0.988 (2832/2866)	0.595	0.995 (2865/2880)	0.786 (11/14)	0.996 (2854/2866)
After GA T=8000 P=200	0.512	0.993 (2859/2880)	0.786 (11/14)	0.994 (2848/2866)	0.629	0.995 (2867/2880)	0.786 (11/14)	0.997 (2856/2866)
After GA T=20000 P=500	0.647	0.996 (2868/2880)	0.786 (11/14)	0.997 (2857/2866)	0.846	0.999 (2876/2880)	0.786 (11/14)	1.000 (2865/2866)

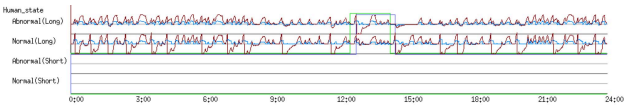

Fig. 8. Experimental result after using Hebbian learning and GA for long-term test data (T=2000, P=200)

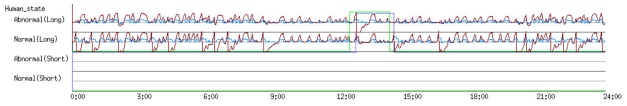

Fig. 9. Experimental result after using GA for long-term test data (T=20000, P=500)

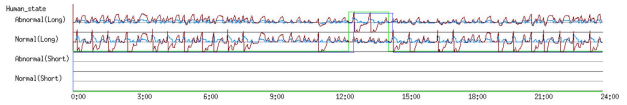

Fig. 10. Experimental result after using Hebbian learning and GA for long-term test data (T=20000, P=500)

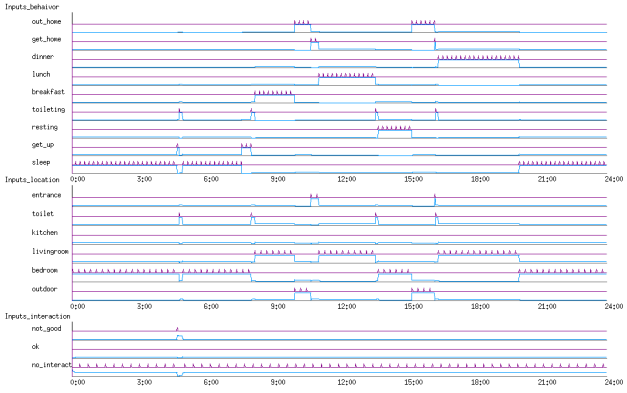

Fig. 11. Input for short-term test data

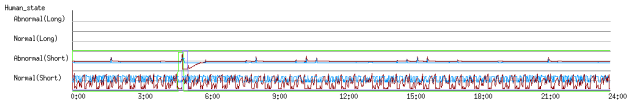

Fig. 12. Experimental result by using SNN for short-term test data

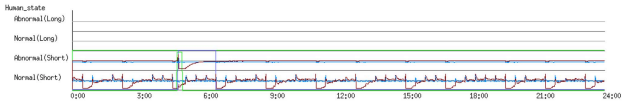

Fig. 13. Experimental result after using GA for short-term test data (T=2000, P=200)

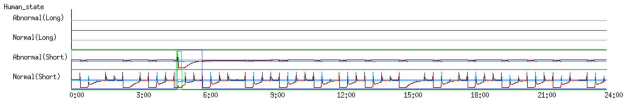

Fig. 14. Experimental result after using Hebbian learning and GA for short-term test data (T=2000, P=200)

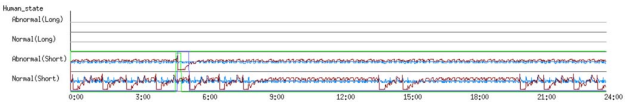

Fig. 15. Experimental result after using GA for short-term test data (T=20000, P=500)

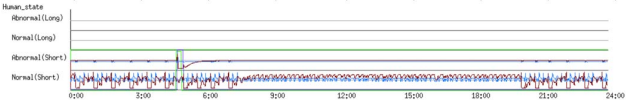

Fig. 16. Experimental result after using Hebbian learning and GA for short-term test data (T=20000, P=500)

parameters from training experiments in order to estimate the test data. Table VII shows the experimental result for training and test datasets. In this case, we calculate the average based on 10 simulation experiments for each data, and for training data we also calculated the standard deviation. The number of total training data is 31680, with 102 abnormal and 31578 normal data. The number of total test data is 2880, with 14 abnormal and 2866 normal data. Figure 11 illustrates the input of short-term test data in one-day human activity. The test data experiment is as follows. Figure 12 shows the estimated result by using SNN. Estimation result is similar to long-term result, which means that, the F-measure, accuracy and fitting rate are not good. The reason is, that feature of SNN's input data does not fit the output state. Figures 13 and 15 show the estimation result by using GA in order to update the SNN parameters. Estimation result is similar to the long-term result, we can see, that the estimation result (purple line) is nearly matching the teaching data (green line), when the number of generations and the size of population are increasing. The F-measure, accuracy and fitting rate became better than applying only SNN. To optimize the parameter of membership function for SNN's input data we use GA, and the feature of SNN's input data is going to fit the output state. Furthermore, the estimation result converges when $T = 20000$ and $P = 500$. Figures 14 and 16 show the application of Hebbian learning and GA. Estimation result has a different value with long-term result. In this case we can see, that Hebbian learning is effective. Estimation result (purple line) is nearly matching the teaching data (green line), and the F-measure, accuracy and fitting rate became better than applying GA and SNN, when the number of generations and the population size are increasing. It is because, we use Hebbian learning in order to influence the learning mechanism of neurons in a short period of time. The Hebbian learning is only effective in a short period of time, but as the time has passed PSP value was forgotten. This is why Hebbian learning is effective in short-term and not so effective in long-term.

IV. Conclusion

In this work, we proposed an evolutionary computation approach to optimize spiking neural network for detecting

abnormal activities in the elderly people's daily life. The initial experimental results showed that the proposed method is able to estimate abnormal activities based on human behavior, human location and human interaction data.

ACKNOWLEDGMENTS

This work was partially supported by MEXT Regional Innovation Strategy Support Program: Greater Tokyo Smart QOL (Quality of Life) Technology Development Region.

REFERENCES

[1] Ministry of Internal Affairs and Communications, "Population estimates," *Statistics Bureau*, 2012.

[2] N. Kubota and A. Yorita, "Topological environment reconstruction in informationally structured space for pocket robot partners," in *Proc. of the 2009 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Dec 2009, pp. 165–170.

[3] N. Kubota, H. Sotobayashi, and T. Obo, "Human interaction and behavior understanding based on sensor network with iphone for rehabilitation," in *Proc. of the International Workshop on Advanced Computational Intelligence and Intelligent Informatics*, 2009.

[4] K. van Laerhoven, D. Kilian, and B. Schiele, "Using rhythm awareness in long-term activity recognition," in *Proc. of the 12th IEEE International Symposium on Wearable Computers (ISWC 2008)*, 2008, pp. 63–66.

[5] J. Kemp, E. Gaura, R. Rednic, and J. Brusey, "Long-term behavioural change detection through pervasive sensing," in *Proc. of the 14th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, 2013, pp. 629–634.

[6] J. A. Anderson and E. Rosenfeld, *Neurocomputing*. MIT press, 1993, vol. 2.

[7] W. Maass and C. M. Bishop, *Pulsed neural networks*. MIT press, 2001.

[8] W. Gerstner and W. M. Kistler, *Spiking Neuron Models*. New York, USA: Cambridge University Press, 2002.

[9] D. Tang and N. Kubota, "Human localization by fuzzy spiking neural network based on informationally structured space," in *Neural Information Processing. Theory and Algorithms*. Springer, 2010, pp. 25–32.

[10] D. Tang, J. Botzheim, N. Kubota, and T. Yamaguchi, "Estimation of human transport modes by fuzzy spiking neural network and evolution strategy in informationally structured space," in *Proc. of the 2013 IEEE International Workshop on Genetic and Evolutionary Fuzzy Systems*. IEEE, 2013, pp. 36–43.

[11] J. Botzheim, D. Tang, B. Yusuf, T. Obo, N. Kubota, and T. Yamaguchi, "Extraction of daily life log measured by smart phone sensors using neural computing," *Procedia Computer Science*, vol. 22, pp. 883–892, 2013.

[12] D. O. Hebb, *The Organization of Behavior*. New York, USA: Wiley and Sons, 1949.

[13] H.-P. Schwefel, *Numerical Optimization of Computer Models*. New York, NY, USA: John Wiley and Sons, Inc., 1981.

[14] G. Syswerda, "A study of reproduction in generational and steady-state genetic algorithms," *Foundation of Genetic Algorithms*, pp. 94–101, 1991.