SICE Journal of Control, Measurement, and System Integration, Vol. 12, No. 2, pp. 056–064, March 2019

Implementation of Human Cognitive Bias on Neural Network and Its Application to Breast Cancer Diagnosis

Hidetaka Taniguchi *, Hiroshi Sato *, and Tomohiro Shirakawa *

Abstract : The neural network is one of the most successful machine learning models. However, the neural network often requires large amounts of well-balanced training data to ensure prediction accuracy. Meanwhile, human learners can generalize a new concept from even a small quantity of biased examples, simultaneously enlarging knowledge with an increase in experience. As a possible key factor in the ability to generalize, human beings have cognitive biases that effectively support concept acquisition. In this study, to narrow the gap between human and machine learning, we have implemented human cognitive biases into a neural network in an attempt to imitate human learning to enhance performance. Our model, named loosely symmetric neural network, has shown superior performance in a breast cancer classification task in comparison with other representative machine learning methods.

Key Words : neural network, cognitive biases, imbalanced sample data.

1. Introduction

The neural network (NN) is one of the most popular machine learning methods and has been studied extensively over the last few decades [1]. Many types of NN have already been developed, such as the convolutional NN [2], the recurrent NN [3], the auto-encoder [4], and the deep learning model [1]. These NNs have shown superior performance across a variety of tasks, such as breast cancer classification, using diagnosis data [5] and image data [6]. However, these NN-based models often require a large amount of well-balanced training data to ensure the prediction accuracy [7]. If the number of training data is insufficient or the distribution of the data is imbalanced, the performance of the NN will decrease [8]. NN-based models, which usually need a well-balanced large database, thus tend to be costly in terms of time and money.

Meanwhile, human beings can generalize a new concept from a small quantity of biased examples [9],[10]. For instance, human children can quickly learn a new animal from just one example, while machine learning requires uncountable amounts of data [11]. Also, human beings can quickly develop knowledge as the examples and experiences increase in number [12], whereas machine learning often faces an underfitting/overfitting problem in such a situation [13]. As exemplified above, there is still a gap between human learning and machine learning. To close that gap, a number of studies have attempted to develop a more human-like machine learning system, inspired by cognitive science and human nature [14]–[16]. A study by Shinohara et al. [17] demonstrated that two well-known cognitive biases–the symmetric bias [17],[18] and the mutually exclusive bias [19],[20] –could be effectively employed in machine learning tasks. The symmetric bias promotes a tendency of inferring *"if q then p"* after convincing that *"if p then q."* For example, if *p* represents *"the weather was rainy"* and *q* represents *"the ground was wet,"* the symmetric bias infers *"if the ground was wet (q), then the weather was rainy a while ago (p)"* from *"if the weather was rainy (p), then the ground was wet (q)"* [21]. Although this kind of inference can lead to systematic errors [17], this tendency in human nature is considered to contribute to faster decision making [21]. The mutually exclusive bias is another tendency in which *"if ¬p then ¬q"* is inferred after convincing that *"if p then q,"* where ¬*p* and ¬*q* are the negations of *p* and *q*. For example, suppose that a mother tells her son, *"if you don't clean up your room, then you will not be allowed to play."* In this sentence, *p* is interpreted as *"not cleaning up a room"* and *q* is interpreted as *"not being allowed to play."* In this case, her son may also interpret the sentence as *"If I clean up my room, then my mom will allow me to play"* (i.e., $\neg p \rightarrow \neg q$), and may thus clean up his room [21]. In this inference, *"if you don't clean up your room, then you will not be allowed to play"* should be interpreted as "punishment"; however, the son seems to misunderstand it as a "reward." Although their conversation involves logical errors, the communication between the mother and her son would seem to be successful because he would clean up his room and then be able to play. Shinohara et al. expected that including both of these biases in a model would yield more human-like inferences [17], as these two biases can lead to incorrect logic but yield faster decision-making. The resulting loosely symmetric (LS) model considers both symmetric and mutually exclusive biases. In cognitive experiments, the LS model has shown to exhibit a very high correlation with human inference [21].

In this study, to apply this human-like nature to machine learning tasks, we have implemented the LS model within an NN breast cancer classifier to learn from a small quantity of biased samples. Our model is an attempt to realize more flexible Hebbian learning [22] from the standpoint of cognitive biases. In Hebbian learning, when the firing of one neuron repeatedly or persistently fires another, the synaptic knob of the axon on

* Department of Computer Science, School of Electrical and Computer Engineering, National Defense Academy of Japan, 1-10-20 Hashirimizu, Yokosuka, Kanagawa 239-8686, Japan
E-mail: sirakawa@nda.ac.jp

(Received April 18, 2018)
(Revised October 31, 2018)

the first neuron is developed, or the axon enlarges existing syntactic knobs of connected neurons. Also, if two neurons do not fire for a certain period, the synaptic knob between them is reduced. The Hebbian learning rules are therefore interpreted as (i) *"if neuron x fires neuron y, then the synaptic knob of neuron x connected to neuron y is developed and enlarged"* and also (ii) *"if neuron x does not fire neuron y, then the syntactic knob of neuron x connected to neuron y is reduced."* The form of symmetric bias corresponds to (i), and the mutually exclusive bias corresponds to (ii). For example, if p represents *"neuron x sent a strong signal"* and q represents *"neuron y was activated,"* symmetric bias leads to *"if neuron y was activated* (q)*, then neuron x sent a strong signal* (p)*"* from *"if neuron x sent a strong signal* (p)*, then neuron y was activated* (q)*."* Also, mutually exclusive bias imparts the tendency that *"if neuron x did not send a strong signal, then neuron y was not activated."* Although the Hebbian learning rules are implemented in general neural networks, the forms of the above cognitive biases more faithfully duplicate the Hebbian forms. As subsequently described, these two biases are implemented in our neural network model with a weight-update rule based on the cognitive mechanism. Namely, the symmetric bias and the mutually exclusive bias imitate the forms of the Hebbian learning rules, and we, therefore, expect that they would contribute to faster decision making. We compared the performance of our model with the performance of five machine learning models including standard NN, support vector machine (SVM) [23], random forests (RF) [24], NN with dropout [25], and NN with batch normalization (BN) [26]. We selected these machine learning models for the following reasons: (i) NN is the base model for our proposed model, and we have attempted to investigate characteristic differences between them; (ii) SVM and RF are the most powerful machine learning models, and we have regarded these models as standard benchmarks of machine learning models; (iii) as subsequently described, our model has a high degree of similarity with the dropout algorithm, which contributes to enhancing the performance of the NN base model by adding noise to the neural network, especially when the quantity of training data is limited; (iv) BN is one of the most powerful techniques to improve the performance of NN and has a similarity to our model to some extent: these two models add process to layer inputs during training. The focus of this paper is to improve the accuracy of the machine learning model especially when a small number of biased examples were given. Our model showed the best performance among the six machine learning models mentioned above for the breast cancer classification task.

The breast cancer classification task is studied for a long time in the field of machine learning using a variety of datasets [5],[6]. In this task, machine learning models often require a large amount of well-balanced training data to assure the prediction accuracy [27]. However, medical datasets, such as that of breast cancer, give some difficulties in obtaining large amounts of well-balanced data due to privacy issues [28]. We, therefore, considered there is a strong need for the machine learning model which can deal such a situation.

One of the most famous solutions for this problem is dropout [25], which omits each of the nodes with a certain probability. Dropout can prevent the overfitting/underfitting problem on NN [29] and has shown superior performance in a variety of tasks [25],[29]. Furthermore, the model can deal with the lack of data when the lack is moderate. However, dropout algorithm would be less effective when extremely few training examples were given [29]. There is still a question on whether dropout is a robust solution for the medical data classification or not.

Our method attempts to solve these problems. The purposes of our model are to improve classification accuracy with a small number of biased training examples and to create a new NN framework which utilizes human cognitive biases. Previous studies such as [17],[30],[31] showed the effectiveness of human cognitive biases for machine learning tasks, especially when the small and biased number of examples were given to the model. In our model, NN with LS can omit nodes and "revive" them according to the status of the network. This framework can provide more flexible representations of the network. In this paper, we utilized the LS model to enhance the performance of NN in the classification task using the small quantity of biased training data.

2. Materials and Methods

2.1 Neural Network

NN is a learning method based on the perceptron [32] and inspired by neuroscience [7]. NN has three kinds of layers, called input, hidden, and output layers. Each layer comprises one or more nodes. The number of nodes in a hidden layer can be variable [7]. Considering an m-layered feed-forward NN, scoring uses a logistic function as in (1)-(2), where x_i^k is the sum of input to i-th node in layer k, $w_{j,i}^{k-1,k}$ is a connective weight to i-th node in the layer k from j-th node in $k-1$ layer, and y_i^k is output of the unit.

$$y_i^k = \frac{1}{1+e^{-x_i^k}}, \tag{1}$$

$$x_i^k = \sum_j^n w_{j,i}^{k-1,k} y_j^{k-1}. \tag{2}$$

The distance between the output of NN y_i^m and true value t_i is calculated as in (3) using the sum-of-squares error function E, where δ_i^m is the difference between the output of NN and the true value.

$$E = \frac{1}{2}\sum_i (y_i^m - t_i)^2 = \frac{1}{2}(\delta_i^m)^2. \tag{3}$$

The aim of the backpropagation is to update the weights $w_{j,i}^{k-1,k}$ so as to decrease δ_i^m. The change in weight $\Delta w_{j,i}^{k-1,k}$ is as in (4), where α is a learning rate.

$$\Delta w_{j,i}^{k-1,k} \ = \ -\alpha \delta_i^k y_i^k (1 - y_i^k) y_j^{k-1}. \tag{4}$$

2.2 Dropout Neural Network

Dropout is a computationally inexpensive but powerful regularization method for NN [25],[29]. The dropout algorithm randomly omits each of the hidden nodes with a certain probability on each presentation of each training example [25],[29],[33],[34]. The dropped nodes do not participate in forward learning or backpropagation. NN with dropout is thus trained by a different network architecture on each presentation of each training example [34]. In a recent study, dropout was assumed to add noise to NN [34], thus preventing too much co-adaptation, especially when there were only a limited number of training data units [25],[33],[35].

2.3 Batch Normalization

BN is a technique for accelerating the training of NN [26]. BN standardizes the distribution of the input of each layer, as in (5):

$$\hat{x}^{(k)} = \frac{x^{(k)} - \mathrm{E}[x^{(k)}]}{\sqrt{\mathrm{Var}[x^{(k)}]}}. \tag{5}$$

Here, $x^{(k)}$ is the layer inputs, $\mathrm{E}[x^{(k)}]$ and $\sqrt{\mathrm{Var}[x^{(k)}]}$ are the mean and the standard deviation over mini-batch [36]. One of the purposes of BN is to avoid internal covariate shift; the phenomenon of the changes of parameters during training affects the distribution of network activations [26].

2.4 Loosely Symmetric Neural Network

We implemented the LS model into an NN framework, thus developing loosely symmetric neural network (LSNN). Table 1 shows the 2×2 contingency table of the LS model, where a, b, c, and d are the frequencies of co-occurrence of p, q, $\neg p$, and $\neg q$ [17].

Table 1 Contingency table of the LS model.

	q	$\neg q$
p	a	b
$\neg p$	c	d

The LS model estimates the strength of the relations between p, q, $\neg p$, and $\neg q$ as defined in equations (6)-(9):

$$LS(q|p) = \frac{a + \frac{bd}{b+d}}{a + b + \frac{ac}{a+c} + \frac{bd}{b+d}}, \tag{6}$$

$$LS(\neg q|p) = \frac{b + \frac{ac}{a+c}}{a + b + \frac{ac}{a+c} + \frac{bd}{b+d}}, \tag{7}$$

$$LS(p|q) = \frac{a + \frac{cd}{c+d}}{a + c + \frac{ab}{a+b} + \frac{cd}{c+d}}, \tag{8}$$

$$LS(\neg q|\neg p) = \frac{d + \frac{ac}{a+c}}{c + d + \frac{ac}{a+c} + \frac{bd}{b+d}}. \tag{9}$$

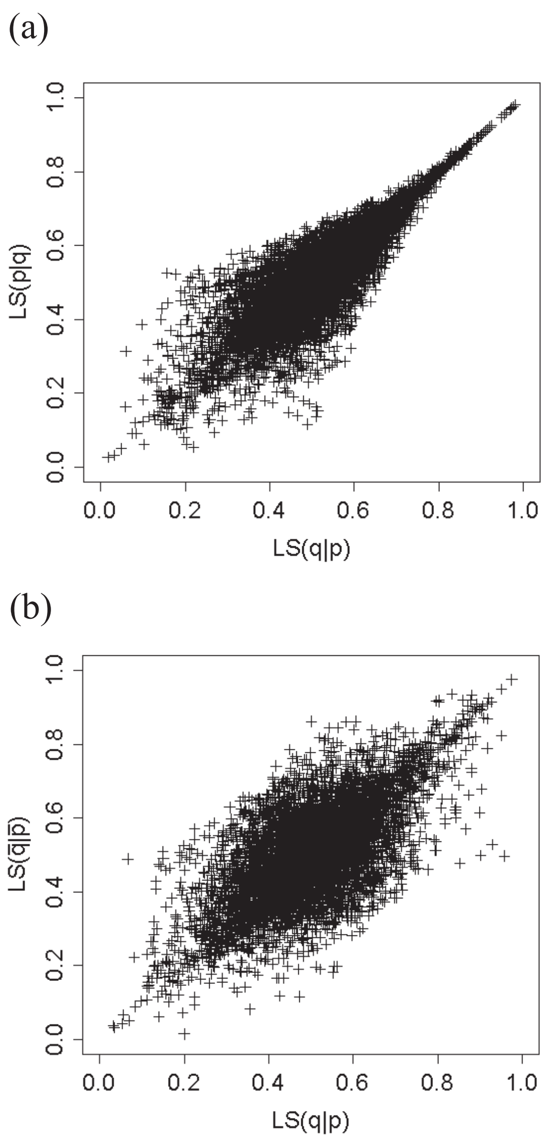

Fig. 1 (a) Relation between $LS(q|p)$ and $LS(p|q)$, and (b) relation between $LS(q|p)$ and $LS(\neg q|\neg p)$.

The LS model is a modification of conditional probability, which includes additional terms $ac/(a+c)$ and $bd/(b+d)$. If these two terms equal zero, LS becomes equivalent to the conditional probability. If $b = c$ is satisfied, (6) and (7) are equivalent, and the LS model satisfies symmetric bias completely. Also, if $a = d$ and $b = c$ are simultaneously satisfied, (6), (8), and (9) are equivalent, and the LS model acquires complete symmetric and mutually exclusive biases. Figure 1 shows the relation between $LS(q|p)$ and $LS(p|q)$ as well as the relation between $LS(q|p)$ and $LS(\neg q|\neg p)$. The data points in the figures are randomly generated by uniformly setting a, b, c, and d from [0, 1]. If $LS(q|p) = LS(p|q)$ holds, the symmetric bias is complete, and the graph in Fig. 1 (a) would show a positive and proportional relationship. Also, if $LS(q|p) = LS(\neg q|\neg p)$ holds, the mutually exclusive bias is complete and the graph in Fig. 1 (b) would show a positive and proportional relationship. If there is no bias, there is no correlation between $LS(q|p)$ and $LS(p|q)$, or between $LS(q|p)$ and $LS(\neg q|\neg p)$, and therefore, Figs. 1 (a) and 1 (b) would be random plots. The distributions of the plots in Fig. 1 show an intermediate shape; a hybrid of proportional and random distribution. If the model always represents a complete symmetric bias or mutually exclusive bias, this result would indicate that the model is too strongly illogical and does not show similarity to human inference. Namely, the LS model exists in an intermediate state between complete bias and no bias. The LS model exhibits the intermediate states of symmetry and mutual exclusivity as shown in Fig. 1. In cognitive experiments, the LS model showed a higher correlation to human inference [21] in comparison with other cognitive models, such as the ΔP [37] model which represents the relationship between response alternatives and outcomes, and dual factor heuristics (DH) models [38]. The reason that the LS model showed such high correlation to human inference is still under investigation. However, a number of studies have shown its effectiveness in machine learning tasks, such as spam classification and the N-armed bandit problem [17],[30],[31]. In [30],[31], the authors contributed to enhancing the prediction accuracy of naive Bayes (NB) using human cognitive biases. The machine learning model named loosely symmetric naive Bayes (LSNB) showed superior classification performances in comparison with conventional machine learning models, using a small and biased number of examples. In order to introduce this method into NN framework, we utilized the LS model to adjust the node values. LSNB and LSNN have some similarity; LSNB calculates its likelihood using the LS model, instead of the product of conditional probability as of NB. Similarly, LSNN adjusts the node values which can be interpreted as the probability distributions that range from 0 to 1 using the sigmoid function [39].

Our LSNN model adjusts the value of each node using LS during feed-forward learning and backpropagation. Our approach is inspired by studies in neuroscience, which imply the existence of the symmetric and mutually exclusive characteristics at the neuron level [40],[41]. For example, a neuronal characteristic of symmetry gives such a tendency that *"if neuron y activated, then neuron x activated."* from *"if neuron x activated, then neuron y activated"* [40]. Also, a neuronal characteristic of mutual exclusivity gives such a tendency that *"if neuron x did not activate, then neuron y did not activate"* [41]. Our LSNN approach is an attempt to reproduce these physiological characteristics in an NN framework from the standpoint of cognitive science. To reproduce the neuronal characteristics of symmetry and mutual exclusivity within the framework of NN, we have implemented a new architecture of NN using LS, which has a high correlation with results in physiology, neuroscience, and human inference [17],[38]. In order to apply this framework to NN, our LSNN is formulated as in (10)-(14):

$$a = y_i^{k-1}, \tag{10}$$

$$b = 1 - y_i^{k-1}, \tag{11}$$

$$c = 1 - x_j^k, \tag{12}$$

$$d = x_j^k, \tag{13}$$

$$LS\,(y_i^{k-1}) = \frac{a + \frac{bd}{b+d}}{a + b + \frac{ac}{a+c} + \frac{bd}{b+d}}. \tag{14}$$

Here, a is the value of node y_i^{k-1}, which is in the $(k-1)$th layer. Therefore, a reflects the degree to which a node has been activated. In addition, b reflects the degree to which a node has not been activated. In (12), (13), c and d reflect the degrees to which node x_j^k has not been activated or activated, respectively. As in (14), LSNN estimates the causal relationship between nodes y_i^{k-1} and x_j^k. If node y_i^{k-1} sends a signal to node x_j^k, then x_j^k is activated, and $LS\,(y_i^{k-1})$ outputs a greater value than y_i^{k-1} because LSNN predicts that y_i^{k-1} has contributed to activating x_j^k and that therefore y_i^{k-1} should be enhanced. Meanwhile, if a node y_i^{k-1} sends a signal to a node x_j^k, then x_j^k is not activated, and $LS\,(y_i^{k-1})$ outputs a lower value than y_i^{k-1} because y_i^{k-1} is considered to be a weak neuron. Also, the change in weight is given as in (15), where $\Delta_{LS}\,w_{ij}^{k-1,k}$ is the change of the weight between node y_i^{k-1} and node x_j^k, which is calculated using LS:

$$\Delta_{LS}\,w_{i,j}^{k-1,k} = -\alpha\delta_j^k y_j^k(1 - y_j^k)LS\,(y_j^{k-1}). \tag{15}$$

The procedure of the LSNN is as follows: (i) Calculate x_j^k using feed-forward learning; (ii) adjust the value of nodes in the $(k-1)$th layer; (iii) update weights using adjusted node values as in (15). Furthermore, LSNN has a different characteristic that is not involved in either standard NN or NN with dropout. In standard NN and NN with dropout, as shown in (4), if node y_i^{k-1} takes a value of 0, its weight $w_{ij}^{k-1,k}$ is not updated. Meanwhile, if x_j^k is activated, $LS\,(y_i^{k-1})$ outputs a value that is greater than 0. Namely, LSNN can update the connection weights between x_j^k and y_i^{k-1} even when y_i^{k-1} takes a value of 0. A significant difference between NN with dropout and LSNN is that the former randomly drops units from a layer, whereas the latter drops units according to the state of the network and also revives dropped units. We assume that this implementation of dropping and reviving nodes more precisely duplicates Hebbian learning, and thus contributes to faster decision making in comparison with standard backpropagation.

2.5 Experimental Settings

We used the Wisconsin Breast Cancer dataset [42] for the breast cancer classification task. The goal of this task was to classify data into one of two classes, *benign* and *malignant*. The Wisconsin Breast Cancer dataset consisted of 699 samples of data, comprising 458 *benign* data and 241 *malignant* data. The percentage of each class was *benign* = 65.5% and *malignant* = 34.5%. The number of features was 10. The features included "Sample code ID" and 9 other features, with values ranging from 1 to 10. The details of the Wisconsin Breast Cancer dataset are given in Table 2. Before the experiment, we eliminated "Sample code ID" from the feature vector and removed 17 samples that had missing data. Therefore, the total number of training data points was reduced to 682. We conducted four experiments with a different number of *benign* and *malignant* data in the learning phase, using six classification models in the task of breast cancer classification: NN, SVM, RF, NN with dropout (Drop-NN), NN with BN (NN-BN), and LSNN. For NN, Drop-NN, NN-BN, and LSNN, we used a three-layered NN with a sigmoid function, which is commonly used for binary classification. The number of nodes in a hidden layer was 30. For Drop-NN, the dropout rate was 50% for hidden units. For NN-BN, we set the mini-batch size as 32 for Exp. 1, 3 for Exp. 2, and 16 for Exp. 3 and Exp. 4, respectively. These numbers were chosen after trials of using 3, 6, 10, 16, 32, and 64. Training was done for 100 epochs for NN, Drop-NN, NN-BN, and LSNN. The SVM classifier was used with Gaussian kernel, which is common for binary classification. The SVM has the cost parameter that is used to determine the decision boundary. Furthermore, the model uses radial basis fuction (RBF) that takes gamma parameter. We set the cost parameter as 0.1 and the gamma parameter as 0.1. RF used 5 trees; this value showed the best performances after trials of using 3, 5, 10, and 30 trees. The parameters of each model were decided after some trials and the best values were chosen. In the following experiments, we used only biased and/or small numbers of training data. The numbers of training data in each experiment were as follows: *benign* = 150 and *malignant* = 150 for Exp. 1, *benign* = 6 and *malignant* = 6 for Exp. 2, *benign* = 150 and *malignant* = 6 for Exp. 3, and *benign* = 6 and *malignant* = 150 for Exp. 4, respectively. In Exp. 1, we used a relatively larger quantity of well-balanced data, while the number of data in Exp. 2 was severely limited. In Exp. 3, the data proportions were highly imbalanced as a re-

Table 2 Characteristic values of Wisconsin Breast Cancer dataset, obtained from [42], where STD represents standard deviation.

Attribute Features	Mean	STD	Mean (benign)	STD (benign)	Mean (malignant)	STD (malignant)
Clump Thickness	4.44	2.82	2.96	1.67	7.19	2.44
Uniformity of cell size	3.15	3.07	1.30	0.86	6.58	2.72
Uniformity of cell shape	3.22	2.99	1.41	0.96	6.56	2.57
Marginal adhesion	2.83	2.86	1.37	0.92	5.59	3.20
Single epithelial cell size	3.23	2.22	2.11	0.88	5.33	2.44
Bare nuclei	3.22	2.15	2.41	1.22	4.71	2.66
Bland chromatin	3.45	2.45	2.08	1.06	5.97	2.28
Normal nucleoli	2.87	3.05	1.26	0.95	5.86	3.35
Mitoses	1.60	1.73	1.07	0.51	2.60	2.56

sult of a very small number of malignant samples, and in Exp. 4, the number of benign samples was similarly limited. The number of test data was 100, with *benign* = 50 and *malignant* = 50. Scores were calculated from averages over 50 trials.

3. Results

The results of Exp. 1 to 4 are shown in Tables 3 to 6, respectively. In Exp. 1, we used a relatively higher quantity of training data in comparison with the other three experiments. In this experiment, LSNN, Drop-NN, NN-BN, SVM, NN, and RF showed better performance, in that order. Here, NN-BN and SVM had the same F-measure value. LSNN showed the best performance with almost perfect malignant classification accuracy; its error rate in benign classification was only approximately 5%; as a result, it showed the best F-measure value. Drop-NN had the best benign classification accuracy and relatively higher malignant classification accuracy, showing the second-highest F-measure. Similarly to Drop-NN, NN-BN had higher benign and malignant classification accuracies, and its F-measure value was the third-highest. Although NN had very high benign classification accuracy, with less than a 5% error rate, its malignant classification accuracy and the F-measure value were lower in comparison with the other three NN-based models mentioned above. Also, SVM showed similar results to LSNN with very high performances on both benign and malignant classifications, and its F-measure value was the third-highest, having the same rank as NN-BN. RF had relatively good scores on both classifications: approximately 0.88 benign classification accuracy, approximately 0.77 malignant classification accuracy, and approximately 0.81 F-measure value. However, in comparison with the other models, RF showed the lowest performance in terms of classification accuracy and F-measure value. In summary, in Exp. 1, LSNN had the best performance in terms of the F-measure values.

In Exp. 2, only a very small number of training data was used to train the machine learning models (*benign* = 6 and *malignant* = 6). LSNN had the best malignant classification accuracy, with approximately a 5% error rate in benign classification. Also, in this experiment, LSNN showed similar results to those in Exp. 1, despite the quantity of training data having been dramatically decreased. Namely, LSNN successfully optimized proper connection weights from a limited quantity of training data and showed the best performance in terms of the F-measure values. Drop-NN had similar benign classification accuracy to that of LSNN. However, the malignant classification accuracy of the former decreased by approximately 0.15 points from its score in Exp. 1 and was much lower than the latter. The F-measure value of Drop-NN in Exp. 2 thus decreased in comparison with its score in Exp. 1. NN-BN had similar tendencies and results to Drop-NN, and its malignant classification accuracy was decreased by approximately 0.15 points from Exp. 1, while its benign classification was relatively high. Namely, Drop-NN and NN-BN were affected by the restricted number of training data in a similar way. NN also showed similar results to Drop-NN; however, its benign classification accuracy was lower than that of Drop-NN. In the Exp. 2, NN decreased its benign classification accuracy by 0.04, its malignant classification accuracy by 0.1, and its F-measure value by 0.08, in comparison with its results in Exp. 1. The difference in performance between Drop-NN and NN seemed to have been reduced in comparison with that in Exp. 1 as a result of the dropout algorithm having become less effective. SVM showed a different tendency from the other five machine learning models, increasing its benign accuracy performance from Exp. 1, and showed the best benign classification accuracy in this experiment. However, its malignant classification accuracy dramatically decreased from that in Exp. 1. Namely, SVM showed a relatively greater difference in performance between its benign and malignant classification accuracies as a result of changes in the quantity of training data. RF decreased its benign and malignant classification accuracies and its F-measure value from Exp. 1 and showed the worst performance. The performance decrements of SVM and RF were greater than those of the NN-base models as SVM decreased its malignant classification accuracy, approximately by 0.2 points from Exp. 1 to Exp. 2. Also, RF substantially decreased its benign and malignant classification accuracies in comparison with the NN-base models. The benign classification accuracy of RF decreased by approximately 0.09 points, from Exp. 1 to Exp. 2. Furthermore, the malignant classification accuracy of RF decreased by approximately 0.21 points, which was almost at the same rate as SVM. Meanwhile, standard NN, Drop-NN, NN-BN, and LSNN did not show such a wide decrease from Exp. 1 to Exp. 2. These four NN-base models were considered to be tolerant of skewed quantities of data. Namely, the NN-base models showed stable performance in comparison with SVM and RF.

In Exp. 3, all machine learning models except LSNN showed higher benign classification performance in comparison with the other three experiments. Although the benign classification accuracy of LSNN slightly decreased from Exp. 1 and 2, in Exp. 3, LSNN showed the best malignant classification performance and F-measure value. Also, Drop-NN did not show a substantial performance decrease from Exp. 1, and showed similar results in comparison with its scores in Exp. 2. Although NN-BN showed the highest benign classification accuracy, its malignant classification accuracy was decreased by 0.64 points in comparison with its scores in Exp. 1, and 0.50 points from Exp. 2. The F-measure score of NN-BN was thus dramatically decreased and was the fourth highest. NN greatly increased its benign classification accuracy in comparison with the results of the two previous experiments described, while greatly decreasing its malignant classification performance and its F-measure value. Namely, NN and NN-BN showed strong sensitivity to imbalanced sample data distributions. NN and NN-BN increased its benign classification accuracy while simultaneously decreasing its malignant classification accuracy and the F-measure value. SVM also decreased its malignant classification performance from Exp. 1 and 2, while its benign classification accuracy was nearly perfect and the highest. However, the F-measure value of SVM decreased in comparison with Exp. 1 and 2. Similarly to NN, NN-BN, and SVM, RF dramatically increased its benign classification performance from Exp. 1 and 2, whereas its malignant classification performance simultaneously became very poor. RF, therefore, decreased its F-measure value, showing the worst performance among the machine learning models. In Exp. 3, in comparison with Exp. 1 and 2, conventional machine learning algorithms NN, SVM, and RF increased their benign classification accuracy and decreased their malignant classification accuracy at the same time. Also, NN-BN showed strong sensitivity to imbalanced data distributions. The malignant classification accuracy of NN-BN became poor in this experiment, and its F-measure score was rather decreased from the conventional NN. Although Drop-NN did not decrease its benign classification accuracy in Exp. 3, its malignant classification accuracy decreased from Exp. 1, while remaining approximately the same as that in Exp. 2. Meanwhile, such decreases in malignant classification accuracy were not observed in LSNN. Although LSNN decreased its benign classification accuracy slightly, it showed more stable learning from imbalanced sample data distributions than the other machine learning models and demonstrated the best performance in terms of the F-measure value.

In Exp. 4, conversely to performance in Exp. 3, most models showed superior results in malignant classification, and their scores in benign classification accuracy were lower than those in the other three experiments. The relative proportions for different types of data between Exp. 3 and 4 were symmetrically opposite. Therefore, most machine learning models showed almost symmetrically opposite scores for corresponding classification accuracies between Exp. 3 and 4, except LSNN. Although LSNN decreased its benign classification performance by approximately 0.06 points, this score decrease was much lower than that of the other machine learning models, as subsequently described. Also, LSNN showed nearly perfect malignant classification accuracy at the same time. The F-measure value of LSNN was still high in Exp. 4 and did not decrease substantially in comparison with Exp. 1 to 3. Meanwhile, the benign classification performance of Drop-NN decreased, surprisingly, in comparison with its value in the other three experiments, and the performance became inferior relative to the benign classification performance and F-measure value of standard NN. Drop-NN, therefore, showed some sensitivity to biased sample data distributions that was not observed in the other three experiments. The F-measure value of Drop-NN thus decreased dramatically from its value in Exp. 1 to 3. NN-BN had perfect malignant classification accuracy, while its benign classification accuracy was relatively low. The benign classification accuracy of NN-BN was the worst among the four NN-base models. Meanwhile, in Exp. 4, NN showed higher classification accuracies and a higher F-measure value in comparison with Drop-NN and NN-BN; NN did not decrease its benign classification performance from Exp. 1 to 3 to the extent of Drop-NN and NN-BN.

Table 3 Results of experiment 1. Classification accuracies for the benign and malignant samples and F-measure values are indicated.

	Malignant	Benign	F-measure
NN	0.928	0.952	0.939
SVM	0.994	0.929	0.963
RF	0.767	0.882	0.814
Drop-NN	0.979	0.953	0.966
NN-BN	0.955	0.972	0.963
LSNN	0.995	0.941	0.969

Table 4 Results of experiment 2. Classification accuracies for the benign and malignant samples and F-measure values are indicated.

	Malignant	Benign	F-measure
NN	0.823	0.912	0.861
SVM	0.797	0.970	0.872
RF	0.556	0.793	0.631
Drop-NN	0.820	0.947	0.876
NN-BN	0.808	0.960	0.868
LSNN	0.977	0.946	0.962

Table 5 Results of experiment 3. Classification accuracies for the benign and malignant samples and F-measure values are indicated.

	Malignant	Benign	F-measure
NN	0.363	0.993	0.530
SVM	0.527	0.994	0.688
RF	0.170	0.962	0.281
Drop-NN	0.820	0.949	0.877
NN-BN	0.312	0.998	0.459
LSNN	0.997	0.901	0.951

Table 6 Results of experiment 4. Classification accuracies for the benign and malignant samples and F-measure values are indicated.

	Malignant	Benign	F-measure
NN	0.998	0.827	0.919
SVM	1.0	0.473	0.791
RF	0.949	0.316	0.721
Drop-NN	0.996	0.678	0.859
NN-BN	1.0	0.511	0.813
LSNN	0.996	0.886	0.944

NN thus showed a higher F-measure value in comparison with its score in Exp. 2 and 3. SVM showed perfect malignant classification accuracy. At the same time, its benign classification accuracy greatly decreased in comparison with its scores in the other three experiments. SVM consistently demonstrated a benign classification accuracy of greater than 0.9 throughout Exp. 1 to 3. However, its benign classification accuracy performance suddenly became poor in Exp. 4. RF showed tendencies similar to those of SVM, NN-BN, and Drop-NN; RF greatly improved its malignant classification accuracy and dramatically decreased its benign classification accuracy performance in comparison with the other three experiments. Conversely to the results in Exp. 3, NN, NN-BN, SVM, and RF sacrificed benign classification accuracy to ensure malignant classification performance. This tendency was also observed for Drop-NN, despite its having demonstrated stable learning in Exp. 3. Therefore, as observed in Exp. 3 and 4, NN, SVM, RF, NN-BN, and Drop-NN decreased their stable learning performance in comparison with Exp. 1 and 2. Meanwhile, LSNN was the only NN-base model that did not exhibit such a tendency and overcame its sensitivity to small biased sample data distributions. Across all four experiments, LSNN did not exhibit sensitivity to small biased examples, and its performance was the best of any of the machine learning models in terms of the F-measure value.

4. Discussion

We conducted four experiments using the most representative machine learning models–NN, SVM, RF, NN-BN, and Drop-NN–and compared their performance with our LSNN in a breast cancer classification task. In Exp. 1, we used a relatively greater quantity of well-balanced sample data in the training phase for the machine learning models. Exp. 1 was conducted to confirm the performance of the machine learning models when a greater quantity of well-balanced sample data had been given to them. As expected, all models showed good performance in this experiment, demonstrating, in particular, the highest performance of LSNN.

In Exp. 2, we used only a very limited number of training examples. Obviously, the performance of the machine learning models decreased in comparison with that shown in Exp. 1, except for that of LSNN. The performance decreases of NN, SVM, RF, NN-BN, and Drop-NN were undoubtedly triggered by the lack of data. However, LSNN, NN-BN, and Drop-NN still exhibited high performance with a relatively small number of data. The reason is that we assume, for a low quantity of data in a dataset, smaller neural network size is more advantageous for concept acquisition. Drop-NN drops its nodes randomly and makes shrinkage on its network; thus, the algorithm starts off with an advantage in adaptation for the lack of data. NN-BN showed a similar tendency to Drop-NN, and its ability of normalization contributed to stable learning, as implied in [26]. The study in [36] mentioned that BN is not suited when the size of mini-batch is small. However, in this experiment, NN-BN showed certain performance with only three mini-batches. NN-BN with a small number of mini-batch is considered to be suitable for a small number of training data in our experiment. LSNN has a mechanism for adjustment of its network size; namely, nodes in LSNN are dropped and revived according to the learning data. This may have created a stronger tolerance to lack of data for LSNN.

In Exp. 3, we used highly imbalanced training examples. NN, SVM, and RF showed differences in performance in this experiment, with very high benign classification accuracy performance and very low malignant classification accuracy performance. Such conventional machine learning methods as NN, SVM, and RF often require large amounts of well-balanced training data to ensure prediction accuracy. Therefore, if the training data are highly imbalanced, conventional models fail to learn properly; e.g., one reason is that these models may mistakenly adjust to noisy distributions. In actuality, in our experiment, NN, SVM, and RF showed high performances in benign classification accuracy. We assume that this is a result of overfitting [13], and these three models may have had difficulties in finding a correct decision boundary, resulting in biased pseudo-negative (pseudo-benign) decisions. Also, NN-BN showed a similar tendency to these conventional machine learning models. We considered the BN technique was less effective under imbalanced data distributions. The performance of BN would be strongly dependent on the activation of layer inputs in the same mini-batch [43]. We considered NN-BN could not normalize layer inputs because of the unstable transitions of E[$x^{(k)}$] and Var[$x^{(k)}$] since the distribution of layer inputs was too biased. We considered this phenomenon triggered too much internal covariate shift [26]. Meanwhile, Drop-NN and LSNN did not exhibit such performance differences between malignant and benign classification accuracies and showing higher accuracies; however, for Drop-NN, there was a slight decrease in malignant classification accuracy in comparison with Exp. 1. Drop-NN creates a different neural network structure at each turn of the learning phase by dropping a number of nodes randomly, and the resulting acquired concept is given by the average of learning by multiple network structures. This property of Drop-NN prevents excessive co-adaptation on noisy distributions; in fact, Drop-NN demonstrated superior results in comparison with the conventional models: NN, SVM, and RF. The performance of LSNN can be explained similarly. LSNN adjusts its network structure according to the data distribution, and this may impart greater tolerance to the imbalance in the dataset than that of Drop-NN.

In Exp. 4, the data proportion was set to be symmetrically opposite to that of Exp. 3. As shown in Tables 5 and 6, all machine learning models showed almost symmetrically opposite scores between Exp. 3 and 4. Namely, malignant classification accuracies increased, and benign classification accuracies, conversely, decreased. Although the drop-NN was proposed to prevent the data imbalance problem in NN, it also showed a performance decrease in benign classification accuracy. As shown in Table 6, the performance of Drop-NN decreased in comparison with the performance of NN. In other words, Drop-NN had strong data sensitivity to imbalanced sample distributions. We predicted that the effectiveness of the dropout algorithm would be limited in some cases. The studies in [25],[34] showed that the dropout algorithm worked successfully in the tasks of handwritten number classification, image-classification, and email categorization. These tasks often use over hundreds of features, and the dropout algorithm is considered to be useful for tasks that address a large number of features. However, in this study, we used only a small number of features for the breast cancer classification task. Furthermore, as shown in Table 2, the variation in values in data was larger in the malignant data, causing the malignant features to be more unclear. Also, although dropout is said to prevent the overfitting problem of NN with a small training set [33],[35], Goodfellow et al. mentioned the dropout algorithm would be less effective when extremely few training examples had been given [29]. Another study mentioned dropout would be less effective when a sufficient number of data had been given [44]. Therefore, its adaptation is rather difficult to perform under certain conditions, such as the experimental settings in Exp. 4. The performance of NN-BN was also decreased in comparison with the performance of NN. The benign classification accuracy of NN-BN decreased more than 0.4 points, and this score was much lower than the benign classification accuracies of NN, Drop-NN, and LSNN. Similarly to the results of Exp. 3, NN-BN was strongly affected by imbalanced data distributions, and it might cause too much internal covariate shift [26]. NN-BN, therefore, showed strong sensitivity to imbalanced data, and its performance was rather decreased from the conventional NN. Meanwhile, our LSNN did not show such sensitivity and showed the best results across all experiments. LSNN showed similar characteristics to LSNB which has strong durability to small and biased data [30],[31]. These two models utilized the LS model to adjust the representation of feature space and showed superior performances using the small and imbalanced examples.

As a result, LSNN overcame the sensitivity to the data-imbalance observed in NN, NN-BN, and Drop-NN by using human cognitive biases. As in Exp. 3 and 4, this is also a result of the adjustment to the neural network structure. Furthermore, LS is a mechanism that learns what is "not A" by observing the presence of "A," and this explains the reason that LSNN is tolerant of the imbalance in data. In conclusion, LSNN, which is an LS-implemented neural network, showed the best performance in all of the experiments, showing tolerance and adaptability to a small, imbalanced dataset.

5. Conclusion

In this study, we implemented human cognitive biases into NN, attempting to imitate human learning to enhance the performance of NN. We conducted four types of experiments using different numbers of training data: a relatively large number of well-balanced training examples in Exp. 1, a restricted number of training examples in Exp. 2, and a biased number of training examples in Exp. 3 and 4. In Exp. 1, LSNN, Drop-NN, NN-BN, SVM, NN, and RF showed high performance, in that order. In Exp. 2, NN, SVM, RF, NN-BN, and Drop-NN decreased their scores in comparison with their results in Exp. 1, whereas LSNN did not show a substantial performance decrease. Also, the performance reductions of NN, SVM, RF, NN-BN, and Drop-NN worsened in Exp. 3 and 4. Meanwhile, our LSNN model did not show much performance reductions as a result of biased sample data distributions, and maintained a high performance, using human cognitive biases. Namely, our model seemed to simulate human learning with respect to some content and overcame the weaknesses of machine learning models that have the sensitivity to small, imbalanced training data. In future research, we will further investigate the relationship between NN, human cognitive bias, and Hebbian learning and how these factors interact in the learning process in order to realize human-level concept learning.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton: Deep learning, *Nature*, Vol. 521, No. 7553, pp. 436–444, 2015.

[2] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner: Gradient-based learning applied to document recognition, *Proc. the IEEE*, pp. 2278–2324, 1998.

[3] J. Hopfield: Neural networks and physical systems with emergent collective computational abilities, *Proc. the National Academy of Sciences*, pp. 2554–2558, 1982.

[4] G. Hinton and R. Salakhutdinov: Reducing the dimensionality of data with neural networks, *Science*, Vol. 313, No. 5786, pp. 504–507, 2006.

[5] A. Marcano-Cedeno, J. Quintanilla-Dominguez, and D. Andina: WBCD breast cancer database classification applying artificial metaplasticity neural network, *Expert Systems with Applications*, Vol. 38, No. 8, pp. 9573–9579, 2011.

[6] Z. Han, B. Wei, Y. Zheng, Y. Yin, K. Li, and S. Li: Breast cancer multi-classification from histopathological images with structured deep learning model, *Scientific Reports*, Vol. 7, No. 4172, 2017.

[7] T. Mitchell: *Machine Learning*, McGraw Hill, 1997.

[8] N. Japkowicz and S. Stephen: The class imbalance problem: A systematic study, *Intelligent Data Analysis*, Vol. 6, No. 5, pp. 429–449, 2002.

[9] B. Lake, R. Salakhutdinov, and J. Tenenbaum: Human-level concept learning through probabilistic program induction, *Science*, Vol. 350, pp. 1332–1338, 2015.

[10] B. Lake, T. Ullman, J. Tenenbaum, and S. Gershman: Building machines that learn and think like people, *Behavioral and Brain Sciences*, Vol. 40, e253, 2017.

[11] L. Gerken, C. Dawson, R. Chatila, and J. Tenenbaum: Surprise! Infants consider possible bases of generalization for a single input example, *Developmental Science*, Vol. 18, No. 1, pp. 80–89, 2015.

[12] U. Goswami: Principles of learning, implications for teaching: A cognitive neuroscience perspective, *Journal of Philosophy of Education*, Vol. 42, pp. 381–399, 2008.

[13] E. Alpaydin: *Introduction to Machine Learning*, MIT Press, 2014.

[14] B. Lake, R. Salakhutdinov, J. Gross, and J. Tenenbaum: One shot learning of simple visual concepts, *Proceedings of the 33rd Annual Meeting of the Cognitive Science Society (CogSci 2011)*, pp. 2568–2573, 2011.

[15] R. Salakhutdinov, J. Tenenbaum, and A. Torralba: One-shot learning with a hierarchical nonparametric Bayesian model, *ICML Workshop on Unsupervised and Transfer Learning 2012*, pp. 195–206, 2012.

[16] D. Lin, E. Dechter, K. Ellis, J. Tenenbaum, and S. Muggleton: Bias reformulation for one-shot function induction, *Twenty-First ECAI*, pp. 525–530, 2014.

[17] S. Shinohara, R. Taguchi, K. Katsurada, and T. Nitta: A model of belief formation based on causality and application to n-armed bandit problem, *T. Jpn. Soc. A. I.*, Vol. 22, No. 1, pp. 58–68, 2007.

[18] M. Sidman, R. Rauzin, R. Lazar, S. Cunningham, W. Tailby, and P. Carrigan: A search for symmetry in the conditional discriminations of rhesus monkeys, baboons, and children, *Experimental Analysis of Behavior*, Vol. 37, No. 1, pp. 23–44, 1982.

[19] E. Markman and G. Wachtel: Children's use of mutual exclusivity to constrain the meanings of words, *Cognitive Psychology*, Vol. 20, No. 2, pp. 121–157, 1988.

[20] W. Merriman, L. Bowman, and B. MacWhinney: The mutual exclusivity bias in children's word learning, *Monographs of the Society for Research in Child Development*, No. 54, pp. i+iii+v+1–129, 1989.

[21] T. Takahashi, M. Nakano, and S. Shinohara: Cognitive symmetry: illogical but rational biases, *Symmetry: Culture and science*, No. 21, pp. 275–294, 2010.

[22] D. Hebb: *The Organization of Behavior: A neuropsychological theory*, John Wiley and Sons, 1949.

[23] C. Cortes and V. Vapnik: Support-vector networks, *Machine Learning*, Vol. 20, No. 3, pp. 273–297, 1995.

[24] L. Breiman: Random forests, *Machine Learning*, Vol. 45, No. 1, pp. 5–32, 2001.

[25] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov: Dropout: A simple way to prevent neural networks from overfitting, *Machine Learning Research*, Vol. 15, No. 1, pp. 1929–1958, 2014.

[26] S. Ioffe and C. Szegedy: Batch normalization: Accelerating deep network training by reducing internal covariate shift, arXiv preprint arXiv:1502.03167, 2015.

[27] G. Weiss and F. Provost: Learning when training data are costly: The effect of class distribution on tree induction, *Journal of Artificial Intelligence Research*, Vol. 19, No. 1, pp. 315–354, 2003.

[28] G. Hrovat, G. Stiglic, P. Kokol, and M. Ojstersek: Contrasting temporal trend discovery for large healthcare databases, *Computer Methods and Programs in Biomedicine*, Vol. 113, No. 1, pp. 251–257, 2014.

[29] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio: *Deep*

Learning, MIT Press, 2016.

[30] H. Taniguchi, T. Shirakawa, and T. Takahashi: Implementation of human cognitive bias on naive Bayes, *EAI Endorsed Transactions on Creative Technologies*, Vol. 16, No. 7, e3, 2016.

[31] H. Taniguchi, H. Sato, and T. Shirakawa: A machine learning model with human cognitive biases capable of learning from small and biased datasets, *Scientific Reports*, Vol. 8, No. 7397, 2018.

[32] F. Rosenblatt: The perceptron: a probabilistic model for information storage and organization in the brain, *Psychological Review*, Vol. 65, No. 6, 1958.

[33] G. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov: Improving neural networks by preventing co-adaptation of feature detectors, arXiv preprint arXiv:1207.0580, 2012.

[34] G. Dahl, T. Sainath, and G. Hinton: Improving deep neural networks for LVCSR using rectified linear units and dropout, *Acoustics, Speech and Signal Processing (ICASSP 2013)*, pp. 8609–8613, 2013.

[35] W. Sun, S. Shao, R. Zhao, R. Yan, X. Zhang, and X. Chen: Sparse auto-encoder-based deep neural network approach for induction motor faults classification, *Measurement*, Vol. 89, pp. 171–178, 2016.

[36] I. Gitman and B. Ginsburg: Comparison of batch normalization and weight normalization algorithms for the large-scale image classification, arXiv preprint arXiv:1709.08145, 2017.

[37] H. Jenkins and W. Ward: Judgment of contingency between responses and outcomes, *Psychological Monographs: General and applied*, Vol. 79, No. 1, pp. 1–17, 1965.

[38] M. Hattori and M. Oaksford: Adaptive non-interventional heuristics for covariation detection in causal induction: Model comparison and rational analysis, *Cognitive Science*, Vol. 31, pp. 765–814, 2007.

[39] F. Chollet: *Deep Learning with Python*, Manning Publications, 2017.

[40] M. Reigl, U. Alon, and D. Chklovskii: Search for computational modules in the C. elegans brain, *BMC Biology*, Vol. 2, No. 1, 25, 2004.

[41] M. Sommer and R. Wurtz: Composition and topographic organization of signals sent from the frontal eye field to the superior colliculus, *Neurophysiology*, Vol. 83, No. 4, pp. 1979–2001, 2000.

[42] Breast Cancer Wisconsin (Original) Data Set, https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(original), accessed 20 February 2019.

[43] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen: Improved techniques for training gans, *Advances in Neural Information Processing Systems 2016*, pp. 2234–2242, 2016.

[44] H. Zhang, Y. Miao, and F. Metze: Regularizing DNN acoustic models with Gaussian stochastic neurons, Acoustics, *Speech and Signal Processing (ICASSP 2015)*, pp. 4964–4968, 2015.

Hidetaka Taniguchi

He received his B.S., and M.S., degrees from Tokyo Denki University, Japan, in 2014 and 2016, respectively. In 2016, he joined National Defense Academy of Japan, where he is currently a Ph.D. candidate. His research interests include artificial intelligence, machine learning, and cognitive science.

Hiroshi Sato (Member)

He is an Associate Professor of Department of Computer Science, School of Electrical and Computer Engineering at National Defense Academy of Japan. He was previously a Research Associate at Department of Mathematics and Information Sciences at Osaka Prefecture University in Japan. He holds Bachelor's degree of physics from Keio University in Japan, and Master and Doctor of Engineering from Tokyo Institute of Technology in Japan. His research interests include agent-based simulation, evolutionary computation, and artificial intelligence. He is a member of JSAI, IEICE, etc.

Tomohiro Shirakawa (Member)

He received his Ph.D. from the Department of Earth and Planetary Systems Science, Kobe University, Hyogo, Japan in 2007. He is presently working as a research associate at the School of Electrical and Computer Engineering, National Defense Academy of Japan. Prior to the National Defense Academy, he worked for the Department of Computational Intelligence and Systems Science, Tokyo Institute of Technology, and for three years as a postdoctoral fellow of the Japan Society for the Promotion of Science. His research interests include biophysics, living systems theory, and bio-inspired computing.