

# Voice Learning of Reservoir Computing Architecture using Ternary Content Addressable Memory with Individuality

Sayaka Akiyama, Go Ajiki, Xiangbo Kong and Takeshi Kumaki  
 Dep. of electric and computer engineering, Ritsumeikan University  
 1-1-1, Noji-Higashi, Kusatsu, Shiga, 525-8577  
 {ri0087ex@ed, kumaki@fc}.ritsume.ac.jp

**Abstract**— With the rapid progress in artificial intelligence (AI) technology, the number of machines that have been designed to interact with human beings has been steadily increasing. However, the responses of such machines to human interactions are often excessively uniform. The purpose of our study is to incorporate the variations that occur during chip manufacturing into machine learning and give own individuality to AI-based robots. In this paper, reservoir computing architecture using Ternary Content Addressable Memory with Individuality is developed and learning is performed using voice data, which is a complicated waveform. It is found that the error of the average of all data between 10 chips is 140% at the maximum. Voice data learning results have individuality outputs.

## I. INTRODUCTION

Artificial intelligence (AI) technology, which has advanced rapidly, is expected to contribute to resolving a wide variety of social problems in fields such as human relationships, overtime work, and decline in production capacities. For example, AI-implemented robots have been developed to provide care for elderly people and serve guests in restaurants [1], [2]. However, the communications expressed by such robots in response to human inputs are often excessively uniform because they are designed to fulfill their roles precisely as programmed. Furthermore, AI-based robots can acquire individuality after learning process, however, teaching and learned data can be copied to other robots. It is difficult to seem to obtain personal like a human.

To overcome these problems, a Ternary Content Addressable Memory with Individuality (ITCAM) [3]-[5], has been developed for generating search result with LSI intrinsic manufacturing variations. In this study, a Reservoir Computing (RC) machine learning architecture [6] is combined with the ITCAM for realizing Individual robots. The RC is easy to make implement to hardware and not to be changed the weight in the middle layer. The proposed architecture is called the ITCAM-based Reservoir Computing Architecture (IRC) [7]. For verifying the IRC

capability, voice learning process implements on the IRC with 10 FPGA chips.

## II. RELATED WORK

This paper shows results some reservoir computing related works. The RC is proposed early 2000s [8]. This architecture can learn time series data, such as audio waveform. Recently, the RC gains exposure for improving security [9]. The use of sustainable and renewable energy has been emphasized worldwide. However, it is necessary to improve the security capability for the smart grid of wind power generation or solar power plants against cyber attacks. The smart grid is a power grid that can be controlled and optimized from both the supply side and the demand side [10]. Therefore, the RC, which has a simple training method and used random data for middle layer weight, is focused on to protect these smart grids. As a result, when the RC-based attack detection is used several attacks size, the attack response rate of the RC closes to 100%. Moreover, it is more resistant than the Multi Layer Perceptron (MLP) and the Support Vector Machines (SVM) methods in various attack detection methods.

The RC architecture can also be used as a climate prediction tool [11]. The Linear Inverse Modeling (LIM), which is a traditional predicting weather system, decides weather condition by checking to some important. This paper compares the performance of the LIM approach with the RC. In results, both the LIM approach and the RC have the same predictive ability when the large amount of training data is prepared, however the RC has better performance when the training data is limited. Thus, the RC is superior to the LIM by calculating the Normalized Root Mean Square Error (NRMSE), which the RC is a quarter of the LIM approach and the Anomaly Correlation Coefficient (ACC) of RC is doubled of the LIM approach. However, it is difficult to optimize the RC, and the pre-fixed weight of reservoir layer makes the prediction accuracy. It is needed to control the parameter.

From these studies, for utilizing the RC capability to prepare pre-fixed random data, we have proposed to ob-

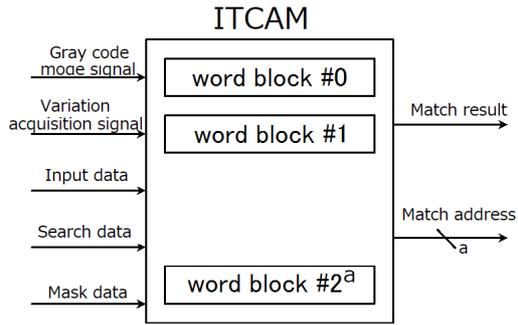


Fig. 1. Block diagram of the ITCAM

tain unique data from LSI chips. Because the data resulting from these chip variations are physically irreproducible and always unique, they can then be used to mimic the characteristics of human individuality.

### III. ITCAM-BASED RESERVOIR COMPUTING ARCHITECTURE

In this section, the ITCAM-based Reservoir Computing Architecture (IRC) is described in detail.

#### A. ITCAM

The ITCAM is a novel type of content addressable memory that makes it possible to reflect diverse search results by manufacturing variations in LSI chips [3]-[5]. Variations are different from each chip. Furthermore, The ITCAM can store quantified manufacturing variations in masking register, and it can output different matching address in situations when same search data are input. A block diagram of the ITCAM package is shown in Fig. 1. When the Gray code mode signal, Delay variation acquisition signal, input data, search data, and mask data are input ports. The ITCAM performs operations such as searching for word blocks, which are internal storage area. The match result and matching address are output ports.

#### B. ITCAM-based Reservoir Computing

In our method, the ITCAM-based Reservoir Computing (IRC) is consisted of reservoir computing architecture and the ITCAM [7]. Searching result by variation data obtains from the ITCAM is used for pre-fixed weights in the reservoir layer. In this paper, the IRC has the ITCAM layer (8 bits  $\times$  8 entries),reservoir layer (M neurons), output layer, and input layer, which contains input data and teach data. A block diagram of the IRC is shown in Fig. 2.

Since we can select the number of neurons (M) in reservoir layer and input data size (insize), each neuron in

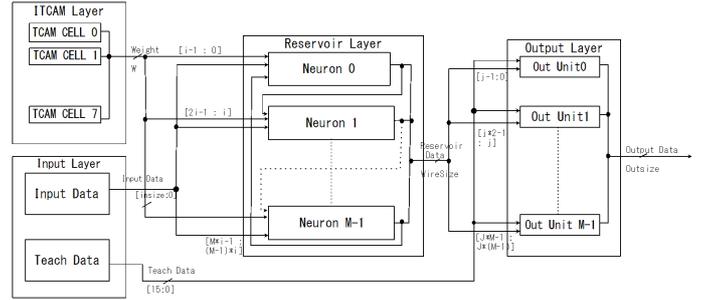


Fig. 2. Block diagram of the ITCAM-based RC architecture

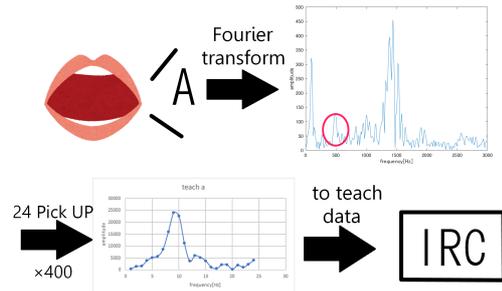


Fig. 3. Process of making voice data to teach data

reservoir layer can receive the ITCAM data for pre-fixed weights. All neurons must be connected to the other neurons. In this paper, we connect neurons like a ring, these neurons can receive all data from the other neurons. For verifying the IRC capability, the IRC is developed Verilog-HDL language and simulated by Xilinx ISE 14.7 design suite.

### IV. EXPERIMENTATION FOR VOICE LEARNING

In order to verify the operational capabilities of our proposed IRC, complex waveform and time-series data in the form of voice-recorded Japanese syllabary from “A” to “Nn” (except for “Wo”) were learned by the IRC. The recorded voice data were then modified to wave teach data by fast Fourier transform (FFT) using the MATLAB programming and numeric computing platform. In this process, each data was multiplied by about 400 for representing waveform in detail. We choose 24 sampling frequencies. The number of neurons was set to 8, and the input size was set to 8 bits. An overview of these processes is shown in Fig. 3.

Voice data waves for teach data are shown in Fig. 4. Input data is sine wave data shown in Fig. 5.

Table I shows reservoir layer weight values [7], obtained from 10 types of FPGA chips (8 bits  $\times$  8 entries) via ITCAM. The Xilinx Spartan6 XC6SCL160 FPGA chip us used in this study. In our experiments, the 8 output

TABLE I  
RESERVOIR LAYER WEIGHT VALUES.

	1	2	3	4	5	6	7	8	9	10
1	183	247	234	229	2	16	0	232	14	10
2	205	5	1	252	25	32	17	245	22	28
3	207	7	1	249	25	31	21	246	30	31
4	161	215	208	206	233	236	233	202	238	235
5	200	6	250	247	60	34	58	245	59	27
6	226	36	18	14	42	50	44	10	48	43
7	22	84	72	67	96	108	96	62	111	102
8	186	242	241	232	6	14	3	231	13	13

TABLE II  
RELATIVE ERRORS OF “Ta” AND “Ri” [%]

	CHIP 1	CHIP 2	CHIP 3	CHIP 4	CHIP 5	CHIP 6	CHIP 7	CHIP 8	CHIP 9	CHIP 10
Ta	358	200	199	600	135	119	361	331	161	262
Ri	1764	363	515	1463	209	680	588	592	202	363

data from the output layer are averaged after one learning process.

### V. EXPERIMENTAL RESULTS

From teaching process with all voice data, 10 chips results are shown in Fig. 6. In order to evaluate each result, the absolute relative error was calculated using the following equation.

$$Relative\_error = \left| \frac{1 - result}{teach\_data} \times 100 \right| [%]$$

#### A. All Chips Verification

The verification results for all chips show larger values than the teach data, because the values of 2 of the 8 neurons are twice those of the other neurons. Therefore, it is considered that the obtained value is multiplied by 1.25. Thus, we multiply all averages by 0.8 for changing neuron values.

Moreover, such as “Ta” and “Ri” from all chips are shown in Table II. All of these looks very different from the teach data.

Chip 1 result of “Ri” is 1764%, which is difficult to represent as “Ri”. This seems to be due to input data and teach data incompatibilities.

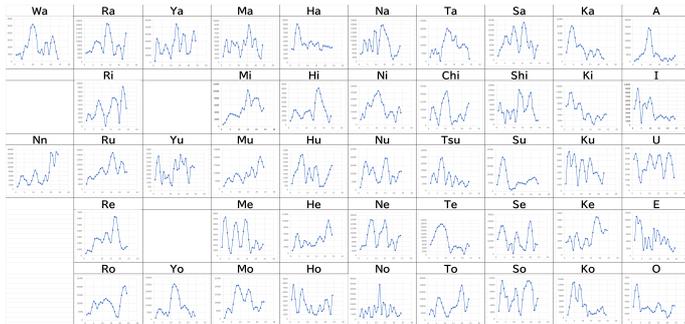


Fig. 4. Voice data wave

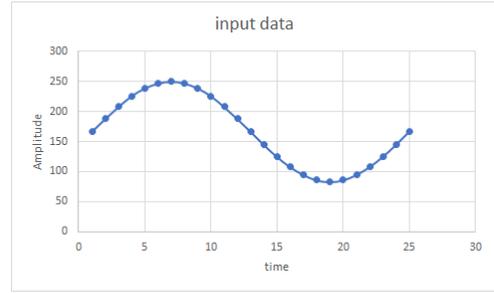


Fig. 5. Inputted sine wave

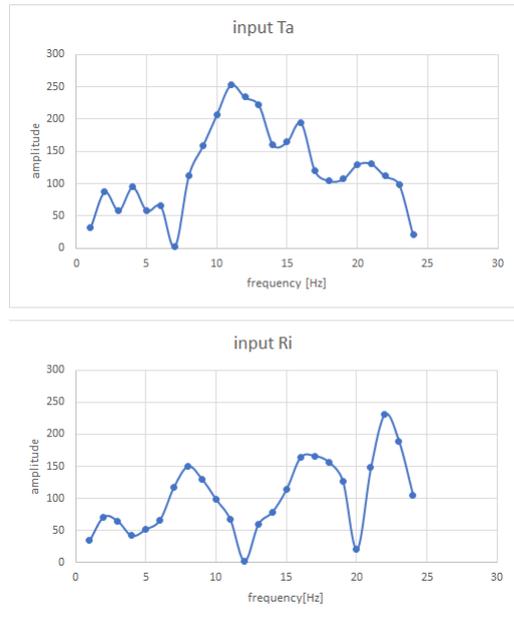


Fig. 7. Up: input data of “Ta”, Under: input data of “Ri”

As a countermeasure, the IRC learning for “Ta” and “Ri” was performed again using the input data shown in Fig. 7. Although these data were made from the same voice data, the results were smaller than their teach data.

From the obtained result, we found that our proposed IRC could improve the relative errors, as shown in Table III.

The relative error of this second learning is 8%, thereby indicating that taking input data and teach data from the same data drastically improves the IRC learning ability.

TABLE III  
ERRORS OF “Ta” AND “Ri” [%]

	CHIP 1	CHIP 2	CHIP 3	CHIP 4	CHIP 5	CHIP 6	CHIP 7	CHIP 8	CHIP 9	CHIP 10
Ta	5	6	7	6	4	6	8	6	6	6
Ri	3	7	8	3	4	2	3	8	2	2

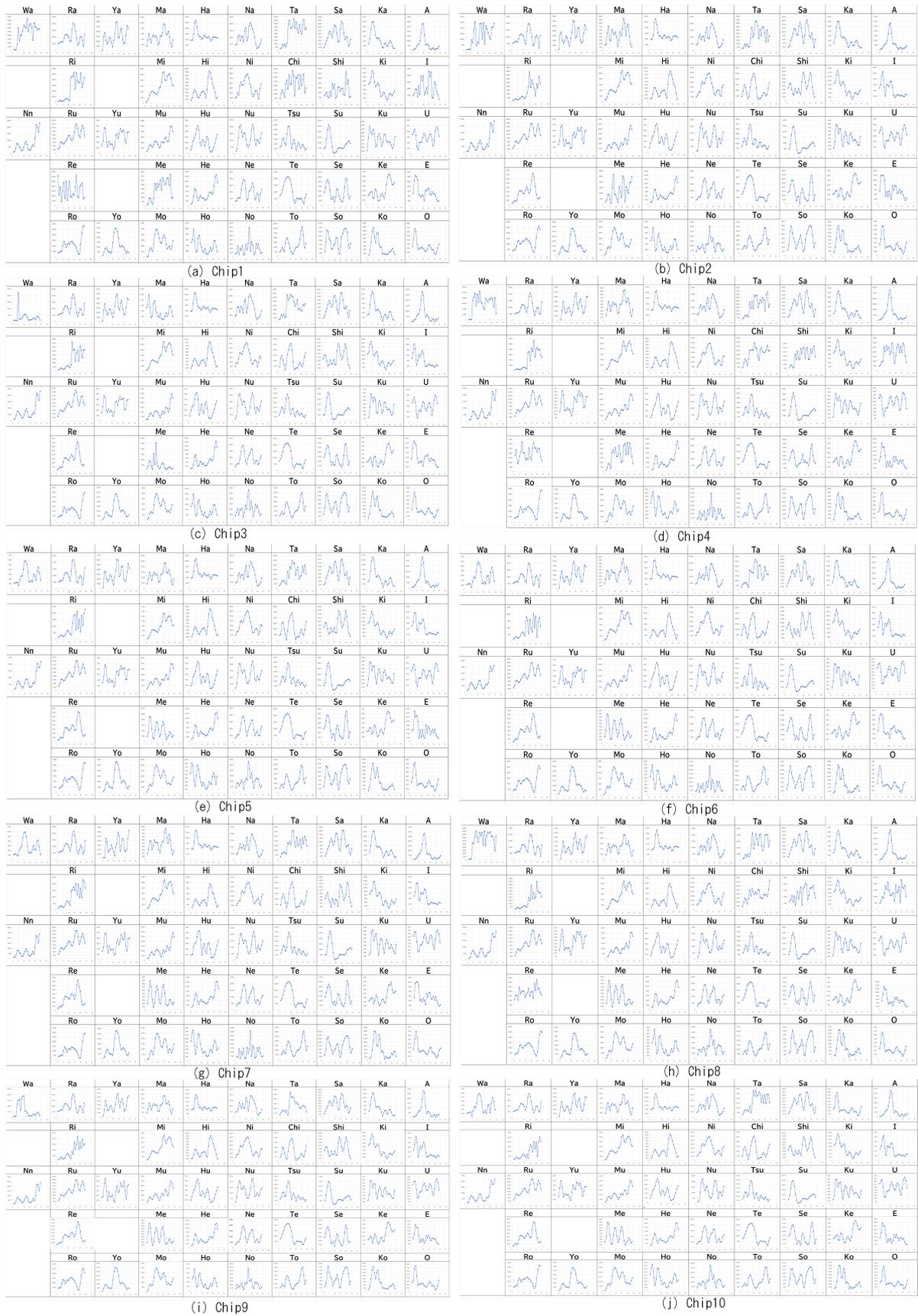


Fig. 6. Voice data learning results of all chips. The vertical and horizontal axes show the amplitude and frequency [Hz], respectively.

Maximum	1764
Minimum	19
Average	156.1
Median	28
100 and over	I, Shi, Ta, No, Me, Ri, Re, Wa
10 or less	Ha
Exemplary Wave	Ka, Ko, Ne, Ha, Mu, Ma
Personal Wave	I, E, hi, Ta, Chi, Me, Ri, Re, Wa

(a) Chip 1

Maximum	515
Minimum	7
Average	47.1
Median	28
100 and over	Ta, Ri
10 or less	Ha, Hu, Ru
Exemplary Wave	Ki, Chi, Ne, Hu
Personal Wave	E, Ta, Ma, Me, Ri, Wa

(c) Chip 3

Maximum	209
Minimum	8
Average	37.1
Median	27
100 and over	Se, Ta, Ri
10 or less	Ha, Mi, Ra
Exemplary Wave	Su, Ma, Ya, Ra
Personal Wave	Ta, Ri

(e) Chip 5

Maximum	588
Minimum	7
Average	51.6
Median	29
100 and over	Ta, No, Ri
10 or less	Ka, Ha
Exemplary Wave	Chi, Me
Personal Wave	I, Ta, Ya, Ri

(g) Chip 7

Maximum	234
Minimum	3
Average	39.7
Median	27
100 and over	Ta, Ri, Wa
10 or less	Ku, Mu, Ro
Exemplary Wave	I, Su, Hu, Mu, Ro
Personal Wave	Ri, Wa

(i) Chip 9

Maximum	630
Minimum	10
Average	66.1
Median	30
100 and over	Se, Ta, Me, Ri, Wa
10 or less	Ro
Exemplary Wave	Su, Ya, Ro
Personal Wave	Ta, No, Ho, Me, Ri, Wa

(b) Chip 2

Maximum	1471
Minimum	8
Average	177.7
Median	28
100 and over	I, Shi, Ta, No, Me, Ri, Re, Wa
10 or less	Ku, Ko, Ha, Mu
Exemplary Wave	Ko
Personal Wave	I, Shi, Ta, Chi, Me, Ri, Re, Wa

(d) Chip 4

Maximum	680
Minimum	4
Average	47.0
Median	27
100 and over	Ta, No, Ri
10 or less	Ne, Ha, Hi, Hu, Mi
Exemplary Wave	Chi, Ne, Hi, Hu
Personal Wave	E, Ta, Ri

(f) Chip 6

Maximum	1625
Minimum	8
Average	122.9
Median	30
100 and over	I, Shi, Ta, Chi, Ri, Re, Wa
10 or less	Hi
Exemplary Wave	Ka, Ne, Hi, Hu, Ra
Personal Wave	I, Shi, Ta, Chi, Ri, Re, Wa

(h) Chip 8

Maximum	363
Minimum	8
Average	43.0
Median	26
100 and over	Ta, Ri
10 or less	Mu
Exemplary Wave	Ku, Su, Chi, Mu
Personal Wave	Ta, Ri

(j) Chip 10

Fig. 8. Maximum [%], minimum [%], average[%], median[%], errors of 100% and over, errors of 10% or less, and the exemplary and personal results for each chip.

## B. Individual Chip Verification

Next, we will give detailed explanations of relative errors for each chip. Table 8 shows the maximum, minimum, average, and median values. Note that errors of 100% and over, 10% or less, and exemplary and personal waves are included.

### Chip 1

The Chip 1 result is shown in Table 8-(a), where it can be seen that Chip 1 has the second-largest error and only one result that is less than 10%. However, the median data value is relatively small. The results for “Ka”, “Ko”, “Ne”, “Ha”, “Mu”, and “Ma” show good wave shapes.

### Chip 2

The Chip 2 result is shown in Table 8-(b). Although the average value is less than 100%, the median value, which is 30, is higher than all the other chips. In addition, “Se”, “Ta”, “Me”, “Ri”, and “Wa” are higher than 100%.

### Chip 3

The Chip 3 result is shown in Table 8-(c), where it can be seen that “Ta” and “Ri” are over 100% and the average is below 50. Additionally, “Ki”, “Chi”, “Ne”, and “Hu” show exemplary waves, thereby indicating excellent learning results for this chip.

### Chip 4

The Chip 4 result is shown in Table 8-(d). In this case, the maximum error is 1471%, which is the third-largest value. This chip has eight results that are over 100% (“I”, “Shi”, “Ta”, “No”, “Me”, “Ri”, “Re”, and “Wa”) but it also has four results (“Ko”, “Ha”, “Mu”, and “Ku”) that are less than 10%. Additionally, the difference between the average and median values is large.

### Chip 5

The Chip 5 result is shown in Table 8-(e), where it can be seen that this chip has the smallest maximum error data and an average and median value difference of just 10%. Additionally, while results of less than 10% are noted for “Ga”, “Mi”, and “Ra”, only “Ta” and “Ri” are the personal waves.

### Chip 6

The Chip 6 result is shown in Table 8-(f). This chip can achieve stable results in cases nor involving weak learning because it has just three errors over 100% and five errors below 10%. However, exemplary waves were only observed for “Chi”, “Ne”, “Hi”, and “Hu”.

### Chip 7

The Chip 7 result is shown in Table 8-(g). For this chip, there are only three syllables over 100%. On the other hand, the average percentage is 51, which is the third-largest value observed. The chip’s exemplary waves are “Chi” and “Me”, and its personal waves are “I”, “Ta”, “Ri”, and “Ya”.

### Chip 8

The Chip 8 result is shown in Table 8-(h). This chip has seven results over 100% and just one result that is less than 10%. In addition, the average percentage is over 100%. From these results, we can say that Chip 8 showed the most effective learning of all the chips tested.

## Chip 9

The Chip 9 result is shown in Table 8-(i), where it can be seen that its maximum percentage is the second to the lowest of all chips tested. On the other hand, with a minimum of 3%, which is smaller than all the others, it can be said that Chip 9 has exemplary learning characteristics.

## Chip 10

Result of chip 10 is Table 8-(j). The Chip 10 result is shown in Table 8-(j). In this case, the maximum percentage is smaller than the others except for Chips 5 and 9. Furthermore, only one result (“Mu”) is less than 10%.

From these results, we can conclude that each chip are implemented with unique characteristics. In addition, while all median percentages are about 30%, larger averages often have some larger errors. In conclusion, weight variations in the middle layer produce different results in each chip and thus create output individual.

## VI. CONCLUSIONS

This paper shows machine learning results achieved by using the individualistic values of LSI chips for reservoir layer weight, we found that even if the same teach data are used, exemplary waveforms can be achieved for each chip by changing the weight. In addition, we found that even though the shape of the obtained personal waveforms differs depending on the chip, each chip will provide some exemplary results. In the future, we will continue to study learning data and strive to clarify individual chip characteristics.

## REFERENCES

- [1] N. Savage, “Robots rise to meet the challenge of caring for old people”, *Nature* 601, S8-S10, 2022
- [2] R. Kawaba, “Japan’s Skylark rolls out robo-waiters for contactless dining”, *Nikkei Asia*, 2021
- [3] S.Kimura, T.Kumaki, T.Fujita, and T.Ogura, “Evaluation of Ternary Content Addressable Memory with Individuality on FPGA”, *Nonlinear Circuits, Communications and Signal Processing (NCSP)*, pp.1-4, 2016
- [4] T.Naito, S.Nakahara, and T.Kumaki, “Proposal of Reservoir Computing Architecture with ITCAM”, *RISP International Workshop on Nonlinear Circuits, Communications and Signal Processing (NCSP)*, pp. 578-581, 2020
- [5] S.Kimura, T.Watanabe, R.Yukawa, T.Kumaki, T.Fujita, and T.Ogura, “FPGA implementation and evaluation of ternary content addressable memory with individuality”, *Journal of Signal Processing*, Volume: 20, No. 4, pp.137-140. 2016
- [6] G.Tanaka, “Reservoir Computings High-speed machine learning theory and hardware for time series pattern recognition”, *Morikita Publishing*, 2021
- [7] T.Naito, S.Nakahara, and T.Kumaki, “Proposal of reservoir computing, architecture with ITCAM”, *RISP International Workshop on Nonlinear Circuits, Communications and Signal Processing (NCSP)*, pp. 578 -581, 2020
- [8] M.Lukojević and H.Jaeger “Reservoir computing approaches to recurrent neural network training”, *Computer Science Review*, Vol. 3, issue 3, pp. 127-149, 2009
- [9] K.Hamedani, L.Liu, R.Atat, J.Wu, and Y.Yi, “Reservoir Computing Meets Smart Grids: Attack Detection Using Delayed Feedback Networks”, *IEEE Transactions on Industrial Informatics*, Vol. 14, No. 2, 2018
- [10] “The Smart Grid”, *Smart Grid.gov*
- [11] B.T.Nadiga, “Reservoir Computing as a Tool for Climate Predictability Studies”, *American Geophysical Union*, 2021
- [12] G. Ajiki and T. Kumaki, “FPGA implementations of reservoir computing with ITCAM,” *RISP International workshop on Nonlinear Circuit, computer and Signal Processing (NCSP)*, 2021.