Thesis for Master's Degree

Radio Frequency Fingerprinting System Based on Model Extension With Outlier Detection

Changyun Lee

School of Electrical Engineering and Computer Science

Gwangju Institute of Science and Technology

2020

Radio Frequency Fingerprinting System Based on Model Extension With Outlier Detection

비 학습 데이터 검출을 이용한 모델 확장 기반 라디오 주파수 핑거프린팅 시스템

Radio Frequency Fingerprinting System Based on Model Extension With Outlier Detection

Advisor : Professor Heung-No Lee

By

Changyun Lee

Graduate: School of Electrical Engineering and Computer Science
Gwangju Institute of Science and Technology

A thesis submitted to the faculty of the Gwangju Institute of Science and Technology in partial fulfillment of the requirements for the degree of Master of Science in the School of Electrical Engineering and Computer Science

Gwangju, Republic of Korea

2019. 12. 17

Approved by

Professor Heung-No Lee

Committee Chair

Radio Frequency Fingerprinting System

Based on Model Extension With Outlier Detection

Changyun Lee

Accepted in partial fulfillment of the requirements for

the degree of Master of science

December 17th 2019

Committee Chair ____________

Prof. Heung-No Lee

Committee Member ____________

Prof. Jong Won Shin

Committee Member ____________

Prof. Jonghyun Choi

MS/EC 20181015

Changyun Lee (이창윤). Radio Frequency Fingerprinting System Based on Model Extension With Outlier Detection (비 학습 데이터 검출을 이용한 모델 확장 기반 라디오 주파수 핑거프린팅 시스템). School of Electrical Engineering and Computer Science. 2020. 33p. Prof. Heung-No Lee

Abstract

Radio frequency (RF) fingerprinting is a way to find a transmitter using specific features in Radio Frequency signal. In recently, RF fingerprinting is getting more attention in electronic warfare or the internet of things network for security. It is an effective method to prevent malicious nodes' access or support allies' electronic devices by identifying if the signal is from the unauthorized transmitter. In the past, RF Fingerprinting research focused on classification problems based on machine learning or classification algorithms. Recently, as deep learning is developed, RF fingerprinting based on deep learning has been studied. However, many other studies still assume a classification of known signal data. In the real world, there are limitations to collecting all signal information for RF Fingerprinting. Therefore, there can be an outlier in performing RF fingerprinting, and we need to consider the outlier. In addition, there is no research on RF Fingerprinting system considering outlier detection with learning. In this paper, we suggest self-learning RF Fingerprinting system. The proposed system is to use incremental learning with Outlier detection in RF Fingerprinting system. It consists of three phases: initial training, outlier detection, and model extension. In the initial training, we train the Convolutional Neural Network (CNN) using existing RF signal. In the outlier detection phase, we use the mahalanobis distance method to detect the outliers and relabel the outliers to new class. In the model extension phase, we retrain the CNN using relabeled data based on incremental learning, with minimal loss of existing knowledge. Our proposed system acquire a 96% accuracy on outliers after model extension.

Contents

List of Figure

List of Tables

1. Introduction

Radio Frequency (RF) fingerprinting technique is finding specific features in RF data and classifying the transmitters using these features [1]. Due to the manufacturing process and device characteristics, same RF signals have different features. The commonly used features are I/Q data of signal, transient signal and spectrogram. Recently, RF fingerprinting research is undergoing in the field of IOT and electronic warfare.

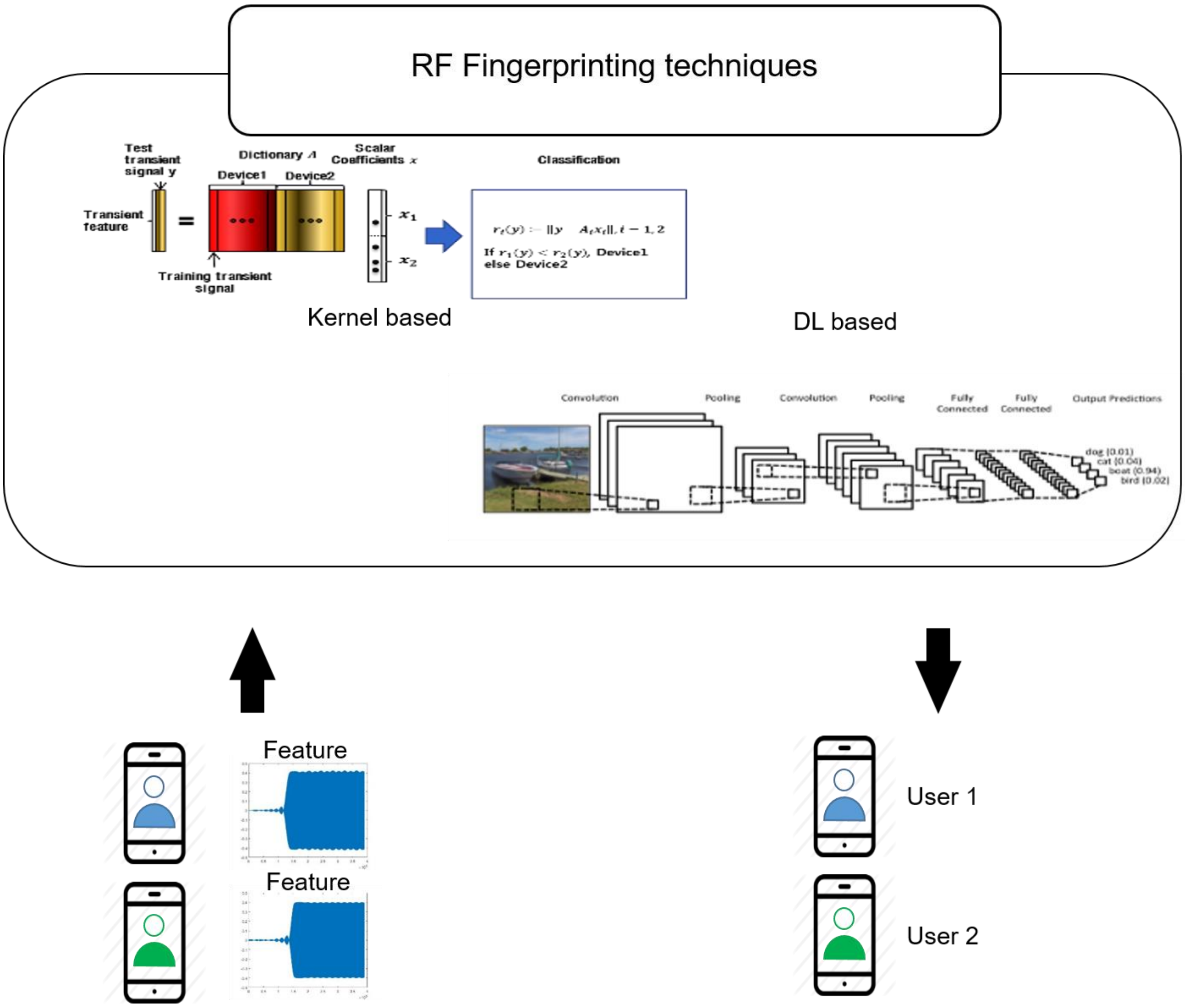

Fig. 1. Example of RF fingerprinting

There are two node having the same device. Even though two nodes are same model, they have different features in transient signal. Using these features, we can classify the two nodes through RF fingerprinting (Fig.1).

Previously, RF fingerprinting research have focused on the classification problem [2-5]: Kiwon *et al.* [2] used transient signal, sync signal, falling signal for features. They concatenate these features and used PCA for feature extraction. They classified the 8 devices by sparse presentation classifier and achieved 98.75% accuracy on 8 devices. As deep learning developed, deep learning also applied to RF fingerprinting. Jafari *et al.* [3] applied

Deep Neural Network (DNN), Convolutional Neural Network (CNN), and Recurrent Neural Network (RNN) to RF fingerprinting. They used I/Q data from six ZigBee devices and got 50 GB of RF data per class. They found out the complexity and overall test performance of every neural network. Merchant *et al.* [4] also applied the CNN to RF fingerprinting. They collected 7000 datum from seven devices. They experimented on outdoor and indoor environments at a different SNR about RF fingerprinting. They achieved 92.92% maximum correct identification rate. These RF fingerprinting algorithms are classification algorithm based on data they have. However, they cannot exactly classify on the data they do not have. These days, some researchers started to consider the data they do not have. Bassev *et al.* [5] considered the intrusion detection in RF fingerprinting. They define intrusion as data that they do not have. They proposed the model pipeline for intrusion detection in RF fingerprinting. They used the I/Q data from 6 ZigBee devices and CNN architecture. They performed the intrusion detection through density based clustering algorithm. They showed the performance metric: Adjusted Mutual Information, Rand Index. They proposed the model for intrusion detection in RF fingerprinting, however this model cannot solve the problem of incorrect classification due this intrusion. Incorrect classification problems caused by data we do not have. Due to this incorrect classification, even if RF fingerprinting algorithms were applied in IOT field, there would be access of malicious node if there is no prior data on malicious node (Fig. 2). If this access persists, the system is fatal. Therefore, the system is needed that learn and blocks them at the system level.

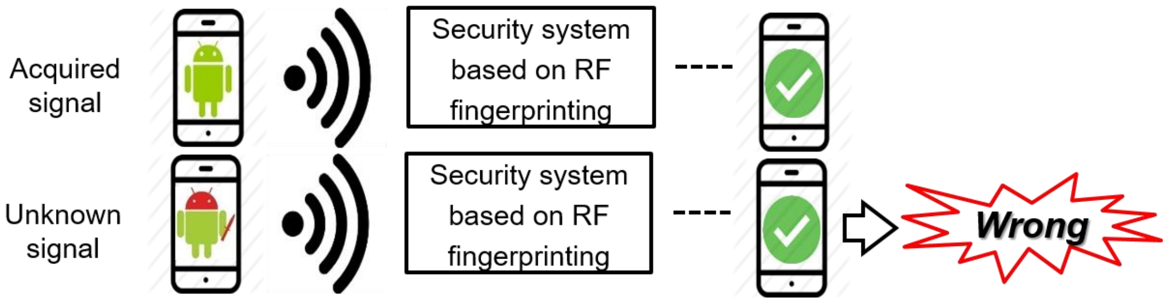

Fig. 2. Incorrect classification problem

In this paper, we propose a self-learning method for RF fingerprinting system. To solve incorrect classification problem, our system detect the unknown signal and learn by itself autonomously. This approach refers to infant's learning [6]. In [6], infants initially moves inefficiently to achieve the goal. However, they move in a way that minimizes inefficiency through repeated observation and action. As such, our system initially learn the data it has through training. After this training, our system performs classification and outlier detection until an unknown signal appears. When an unknown signal appears, our system finds it through outlier detection and relearns it as a new class (Fig. 3). We applied the outlier detection method and incremental learning in the proposed system and used the CNN classifier. We verified our system performance based on the MNIST data and show the good performance on MNIST data. In the end, we applied our proposed system to RF fingerprinting environment and confirmed the possibility.

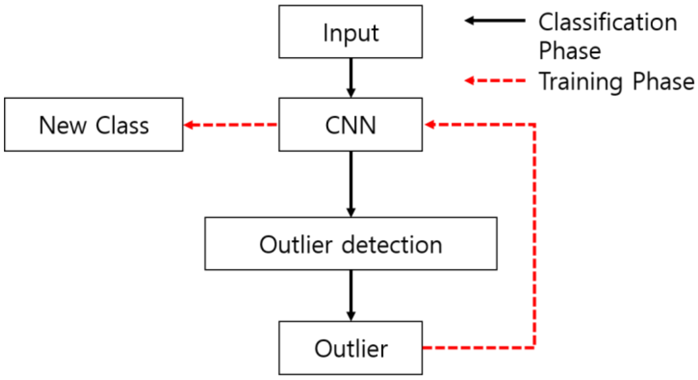

Fig. 3. Proposed system workflow

The remainder of this paper is organized as follows:

In chapter 2, prerequisite knowledge on the proposed system will be discussed for understanding. In this part, we will explain about outlier detection and incremental learning. Firstly, we will introduce the outlier detection method and related works about that. Then, we will explain the incremental learning method and related works.

In chapter 3, we will explain our research process by outlier detection and incremental learning part. In outlier detection part, we will explain how detect the outliers in our proposed system using specific detection rule. Following the result of outlier detection, we will demonstrate how retrain the classifier in the incremental learning part.

In chapter 4, we have two parts: experimental setup and result. In the experimental setup, we will explain about our experimental environments and database for test. In the experimental result, we will show our proposed system result based on MNIST and RF data. We will compare our system with 2class classifier and 3class classifier.

In chapter 5, we will give summary and future work of this paper.

2. Background

The proposed system is self-learning system that system detect the unknown and missing data and learn that by itself. Unknown and missing data mostly appears as outlier in the system. For this reason, we implemented the outlier detection method in our system. Then, we also applied incremental learning method for learning by itself. Before showing the details of our research process, we will explain the basic knowledge of these two things in this part.

2.1. Outlier detection

We use the outlier detection method for detecting unknown and missing data. Outlier detection is finding rare events or noticeably different values in data (Fig. 4). This method usually used in the field of data mining. Outlier detection is also called as anomaly detection and novelty detection.

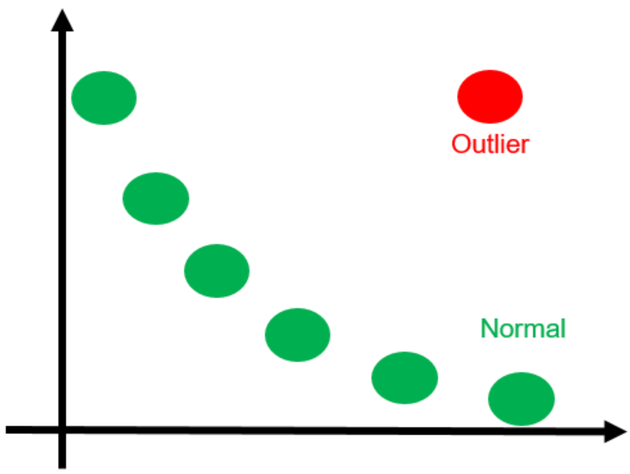

Fig. 4. Outlier detection

For the outlier detection, two methods are proposed. One thing is unsupervised learning. One of the unsupervised learning is clustering. There are several clustering algorithm: density based, depth based and so on [7]. These algorithms detect the data that is not in a cluster as an outlier. The other thing is semi-supervised learning. One of the semi-supervised learning is generative model. Generative adversarial networks (GAN) are examples of this generative model [8]. Recently, this GAN often used for the outlier detection [9-10]. [9] used GAN for learning feature map from normal image. When unknown images are input, anomaly detection is performed by calculating the difference from closet image. [10] also used GAN for outlier detection. Authors trained the generator using feature-matching loss. For the outlier detection, they expressed the normal and outliers as probability distribution ratio using GAN.

2.2. Incremental learning

Then, for learning these outliers, general supervised learning can be suggested. However, in supervised learning, we need to train classifier on whole dataset containing the outliers because of catastrophic inference. Catastrophic inference is forgetting the existing knowledge in neural network model [11]. It caused by hierarchical structure of neural network. It is inefficient way to train classifier. Therefore, we used the incremental learning method for learning specific data. Incremental learning is method for learning new data while minimizing the existing knowledge in model (Fig. 5). Incremental learning used for data analytics and big data processing, robotics, image processing, automated annotation and outlier detection [12].

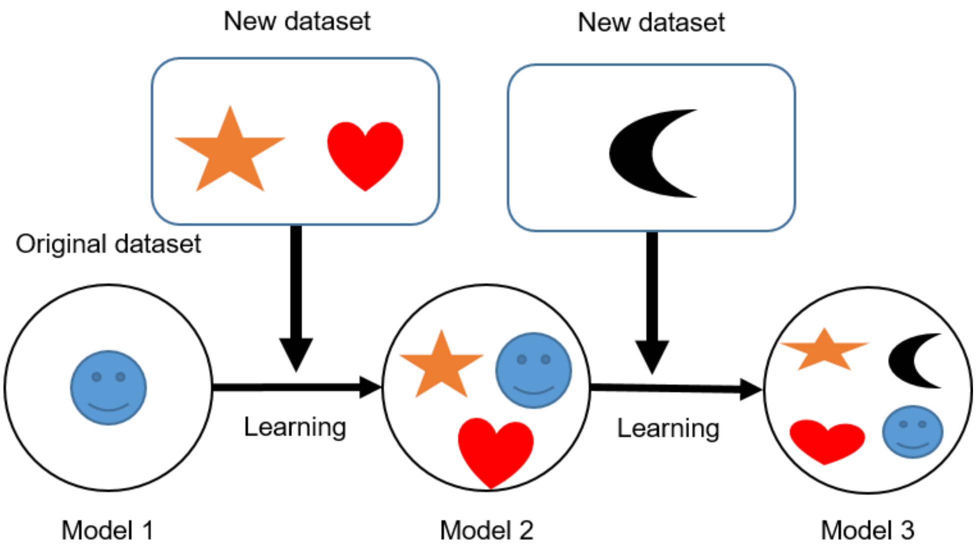

Fig. 5. Incremental learning example

For the incremental learning, fine tuning method is generally proposed (Fig. 6). Fine-tuning is to change the network architecture based on pre-trained network. Changed network's parameters are updated from pro-trained network. Changed network's parameters are updated from pre-trained network's parameters through training. [13-14] also used fine tuning method for incremental learning. In the [13], authors acquired the new parameters from new data using pre-trained CNN. Then, while randomly initializing the new parameters and fixing the old parameters in pre-trained CNN, they retrained the classifier. Once again, they conducted the training of classifier on whole parameters for incremental learning. Authors of [14] used the concept of fine-tuning for incremental learning. Especially, they used the reinforcement-learning model for incremental learning. Through their model, they divided the pre-train network into two-parts: core-part and edge-part. Core part is the connection that mainly consisted of existing knowledge. Edge-part is initialized connection that does not significantly affect knowledge. For the edge part, they also added nodes in layer according to result. As a result, they used two parts with different learning rate for incremental learning. Similarly, in [15], they also figured out the importance of connections between the layers through prototype field and pruned the connections. They proposed that they could perform incremental learning through this pruning.

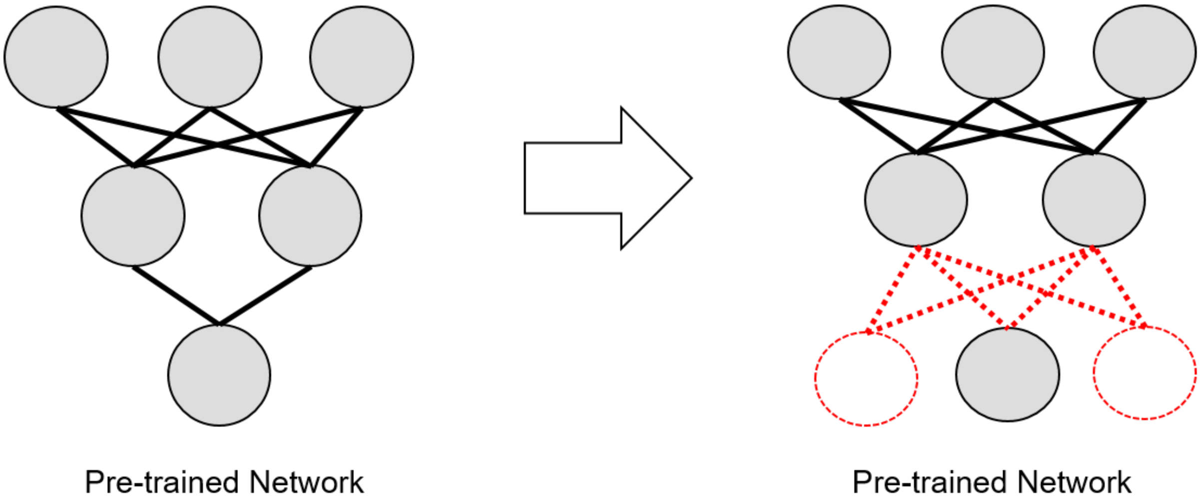

Fig 6. Fine-tuning example

3. Proposed method

In this part, we show the details of proposed system and research process. The details of our proposed system are shown in Fig 7. Our proposed system is divided into three parts: initial training, outlier detection, model extension. In the initial training part, we trained the classifier using data we have. After that, we set up the rule for outlier detection in the next part. In the last, we applied the incremental learning approach for learning outliers. We will explain about research process of these through subsection of this part.

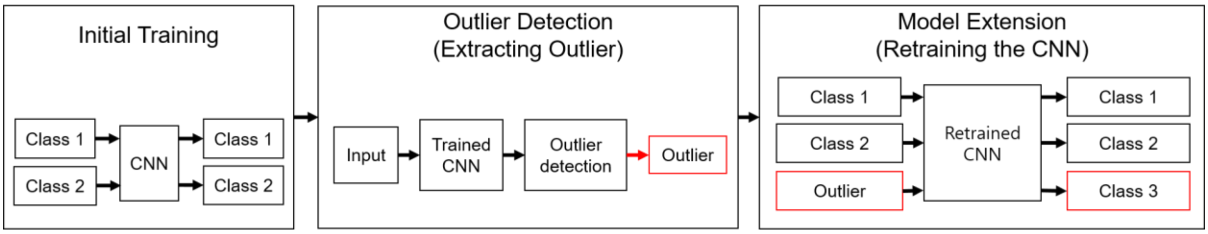

Fig. 7. Proposed system description

3.1. Initial training.

For the training, we acquired the signal from the same models. We used the Radio signal recording system for acquiring signal. The recording system acquired the transmitted signal in wired connection (Fig. 8). We acquired 15 signals from 3 military transmitters.

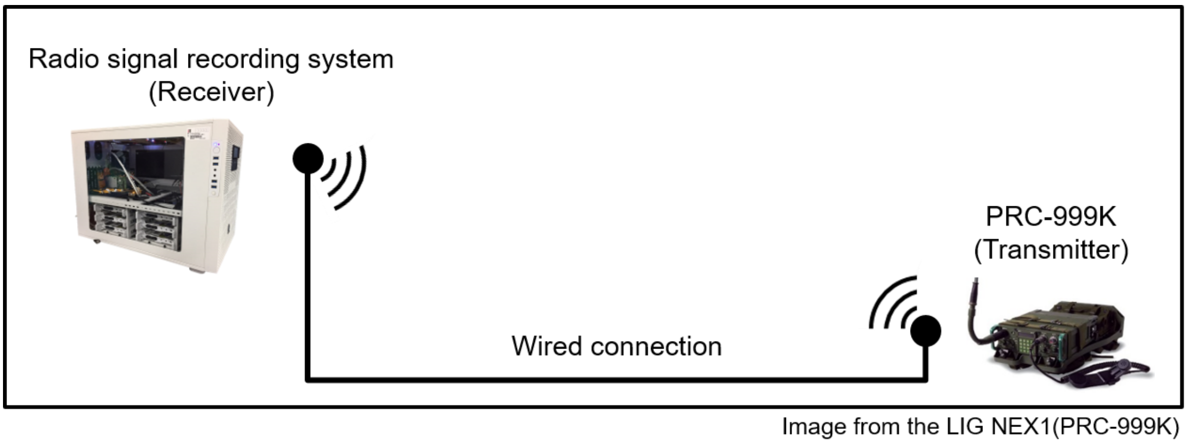

Fig. 8. Signal acquisition experiments

Before the initial training, we conducted the data pre-processing. Our received raw signal is shown in the Fig 9. The signal is frequency hopping signal in 30~88MHz. One signal is consisted of 34 burst and 134217728 samples. We used each hop for training. For extracting the each hop, we use a threshold method. Threshold method is performed through time windowing.

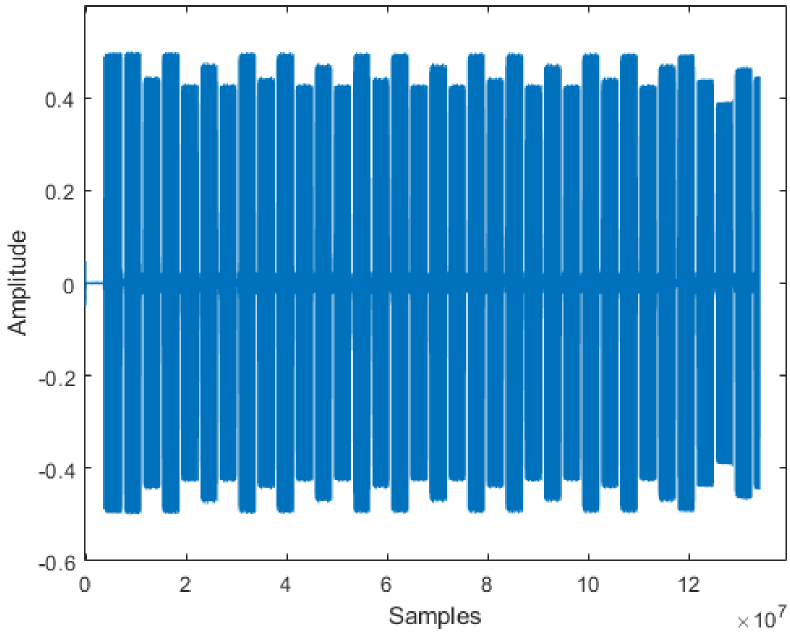

Fig. 9. First signal of first device

Fig. 10 shows the extracted first hop from the burst. We set the starting and ending point of time window based on the time to reach 10% of the maximum amplitude of burst. Then, each hop has 3900000 samples.

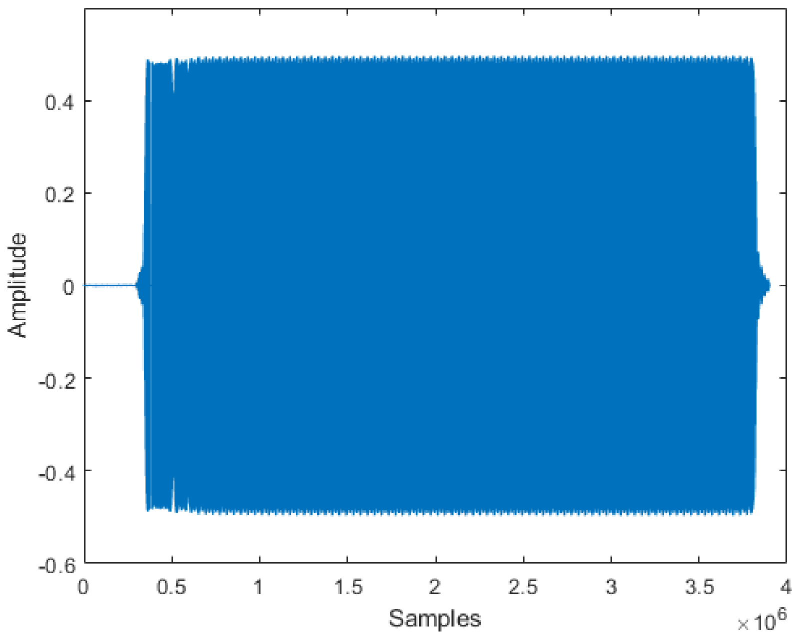

Fig. 10. Extracted first hop signal

After that, we applied Short-Time Fourier Transform (STFT) to each hop for acquiring spectrogram. The reason why we acquire the spectrogram is our classifier that CNN is well known for image classifier and spectrograms are often used in signal processing based deep learning research [16-17]. Through STFT, we acquired the 1028x1904 of spectrogram (Fig. 11). We took the absolute value of the spectrogram and converted it to a log scale. After, we applied image resizing to 1/4 based on bicubic interpolation. Through this data pre-processing, we acquired the 257x476 of spectrogram for training data (Fig. 12).

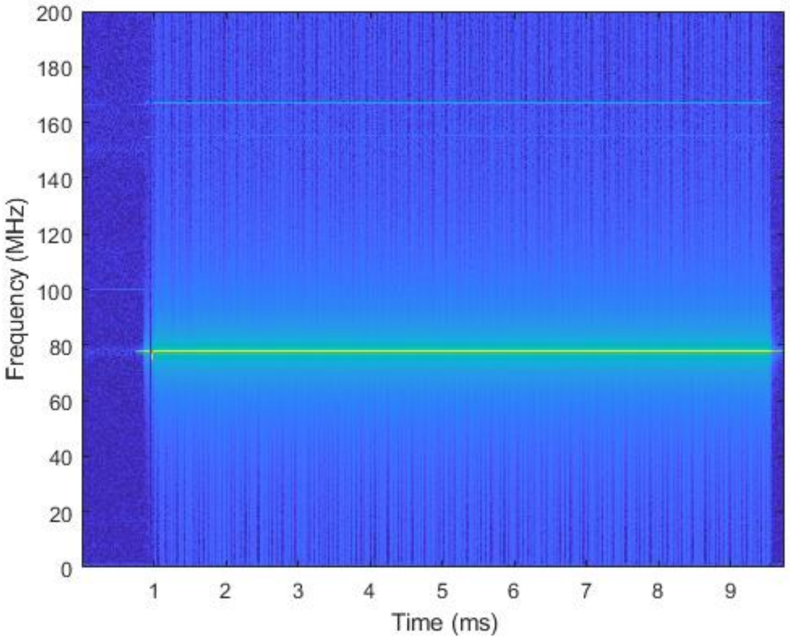

Fig. 11. Spectrogram of the 1st hop signal

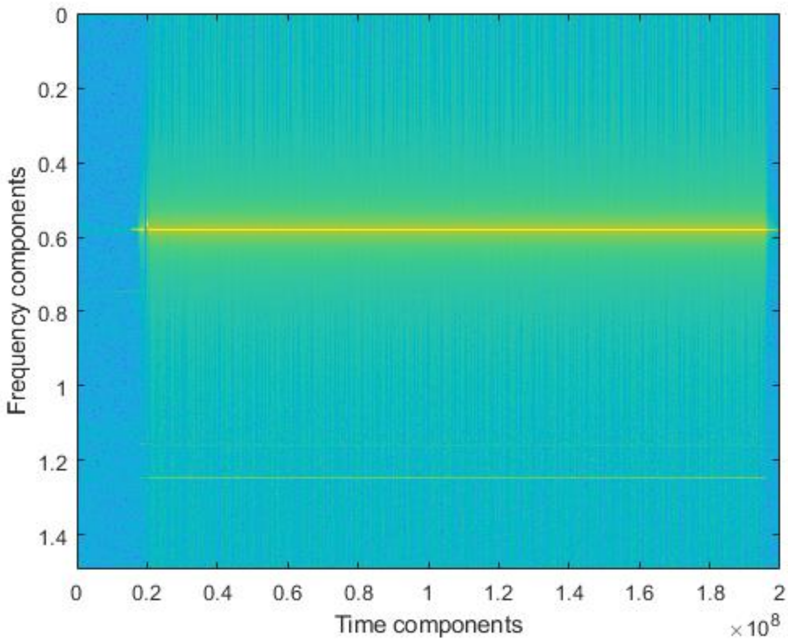

Fig. 12. Resized Spectrogram for training data

Using this spectrogram, we trained the CNN for 2-class classification. We left the third class as outlier data. Table 1 shows the our classifier property. After the initial training, we used the values of output layer for outlier detection (Fig. 13).

Layer	Kernel Size	Output Shape	Parameters	Activation
Input	-	1x257x476	-	-
Conv2D	3x3	64x128x237	640	RELU
Maxpool2D	2x2	64x64x118	-	-
Conv2D	3x3	32x31x58	18464	RELU
Maxpool2D	2x2	32x15x29	-	-
FC1	-	512	7127552	RELU
FC2	-	128	65664	RELU
Output	-	2	258	-

Table. 1. CNN Property

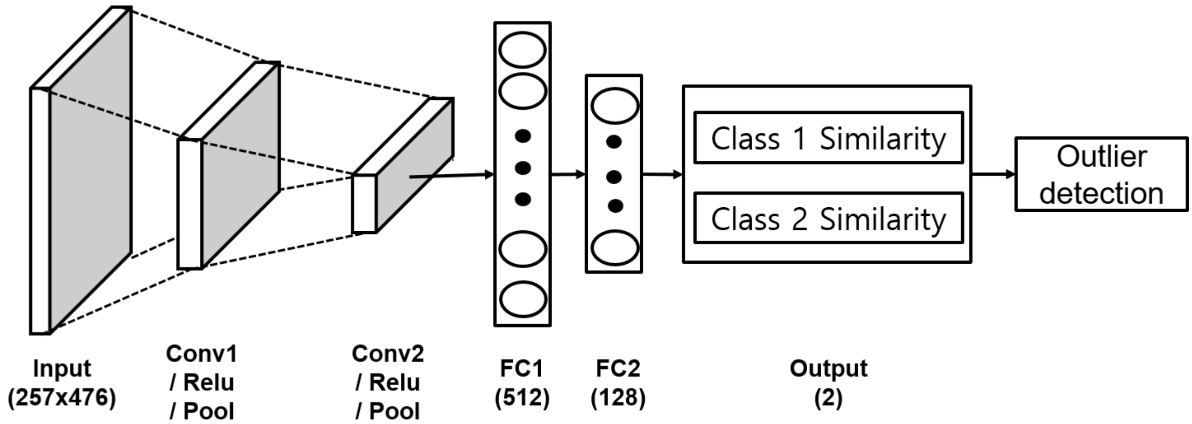

Fig. 13. CNN Architecture

3.2. Outlier detection

For outlier detection, the system needs a specific rule that could distinguish between normal and outlier. To define the specific rule for outlier detection, we first conducted the data analysis in this part. Through the data analysis, we set a rule for outlier detection and have implemented to system algorithmically. We will explain the detail in a subsection.

3.2.1. Data analysis

As we mentioned, we used the values of output layer for outlier detection after initial training. However, in the values of the output layer, there was an overlap of distribution between outlier and learned class (Fig. 14). This overlap causes the system to misjudge outliers as learned class.

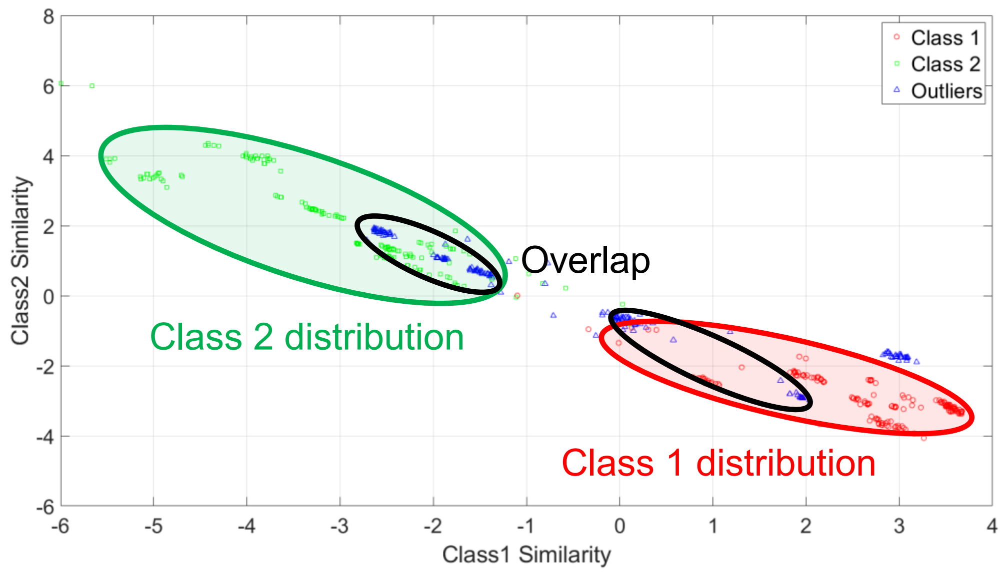

Fig. 14. Overlap in output values

To solve this problem, we used t-stochastic neighbor (TSNE) embedding method. TSNE is data reduction and visualizing method for vector [18]. TSNE mainly used for data analysis. Through the TSNE, we projected the components of learned class and outliers to 2-D domain. Therefore, we confirmed that there is an overlap between learned class and outliers (Fig. 15). To reduce this overlap, we considered feature map in FC2 layer. The reason why use feature map is to increase the amount of information by increasing the feature dimensions. When we also applied TSNE to the feature map, we could confirm that there is no overlap in feature map (Fig. 16).

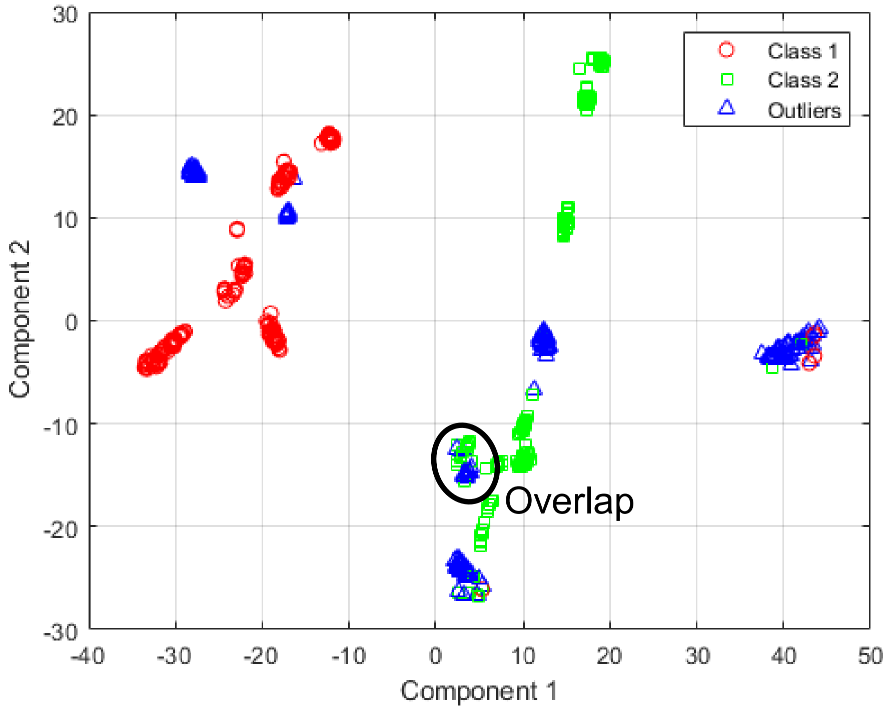

Fig. 15. The value in output layer applied TSNE

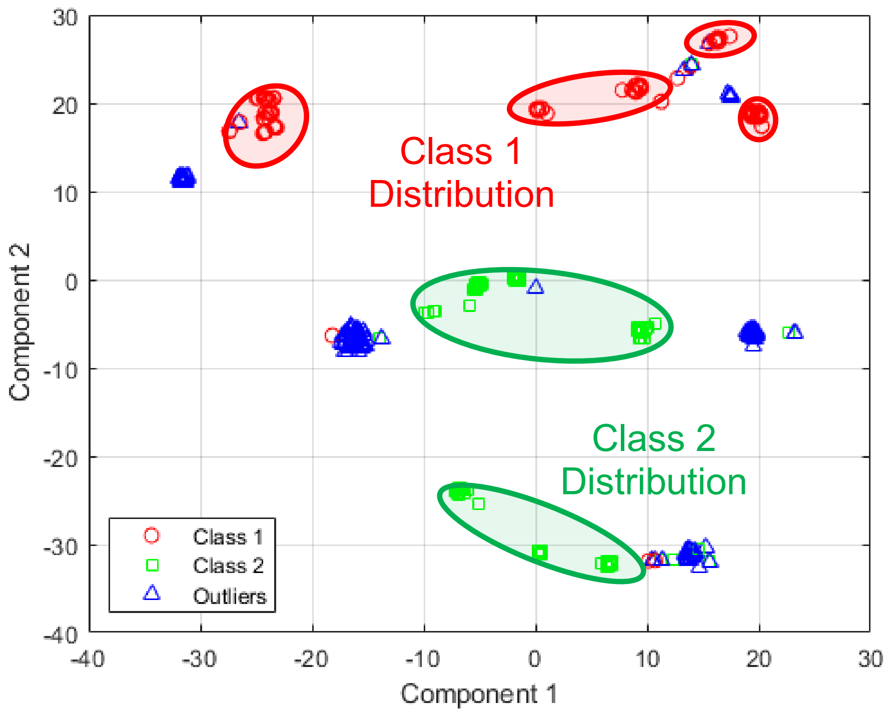

Fig. 16. The feature map in FC2 layer applied TSNE

For better examples, we also used MNIST data. There is a more pronounced difference between learned class and outliers in feature map values (Fig. 17, 18).

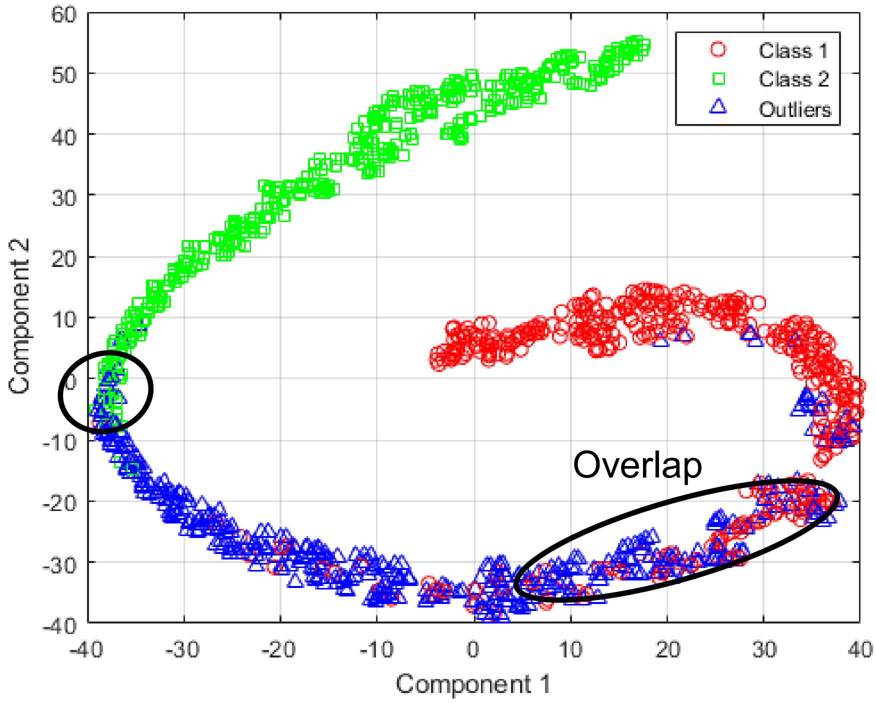

Fig. 17. Output value based on MNIST data

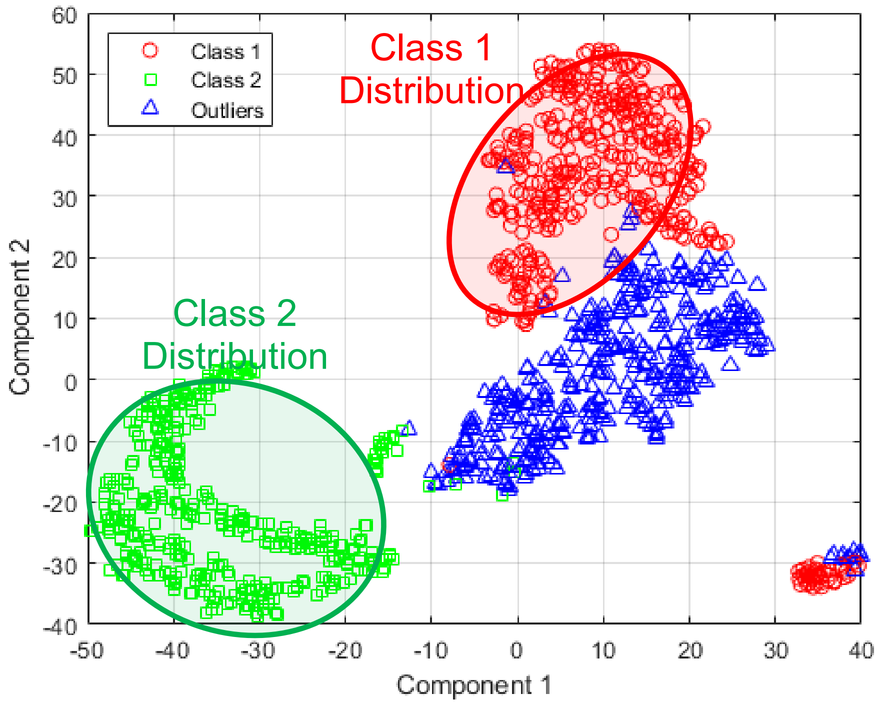

Fig. 18. Feature map based on MNIST data

Therefore, we also adapt to use feature map for outlier detection (Fig.19).

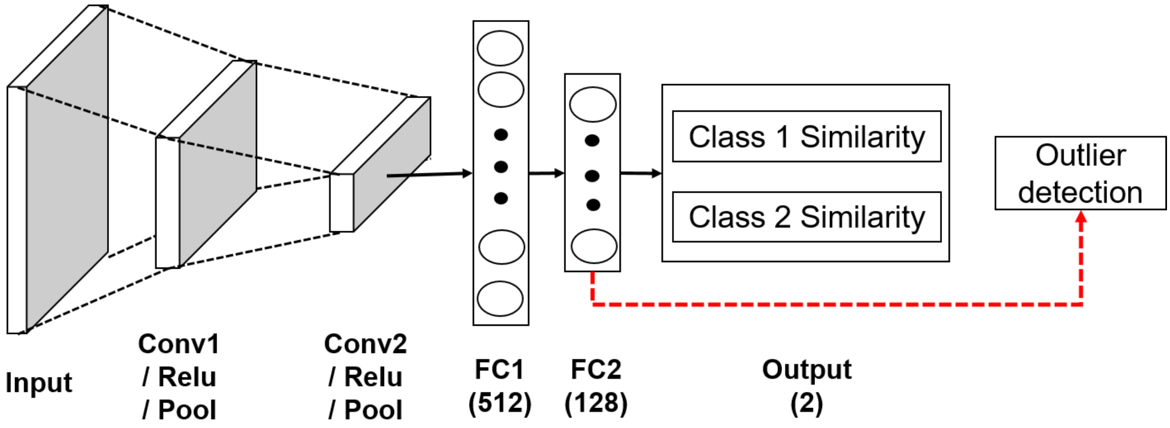

Fig. 19. CNN Architecture for outlier detection

3.2.2. Detection rule

Through the data analysis, we set a rule for the outlier detection. Learned class and outliers have specific each distribution. However, outliers does not follow the learned class distribution. Therefore, when a signal comes in, if the signal does not follow the distribution of the learned class, we classify these signals as outliers (Fig. 20). In this rule, we assumed that the data follow a Gaussian distribution based on the central limit theorem, assuming that the number of distributions with mean and variance is sufficient.

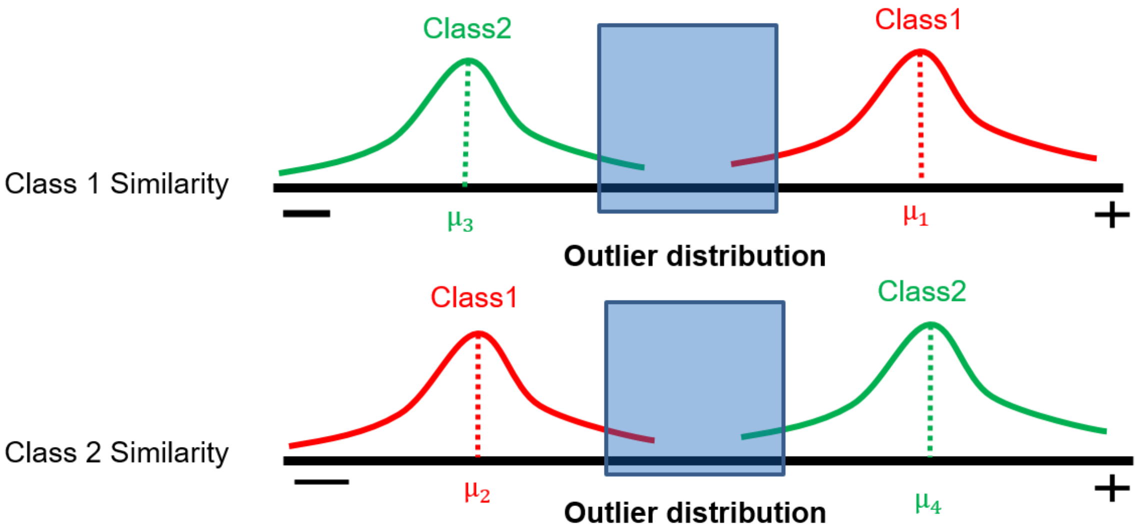

Fig. 20. Learned class and outlier distribution

For the detection, we used mahalanobis distance (MD). MD is a method of measuring a distance between distribution and point (Fig. 21). A distribution has mean and standard deviation. MD calculate a distance between point and mean of distribution. Then, MD express this distance as a multiple of standard deviation.

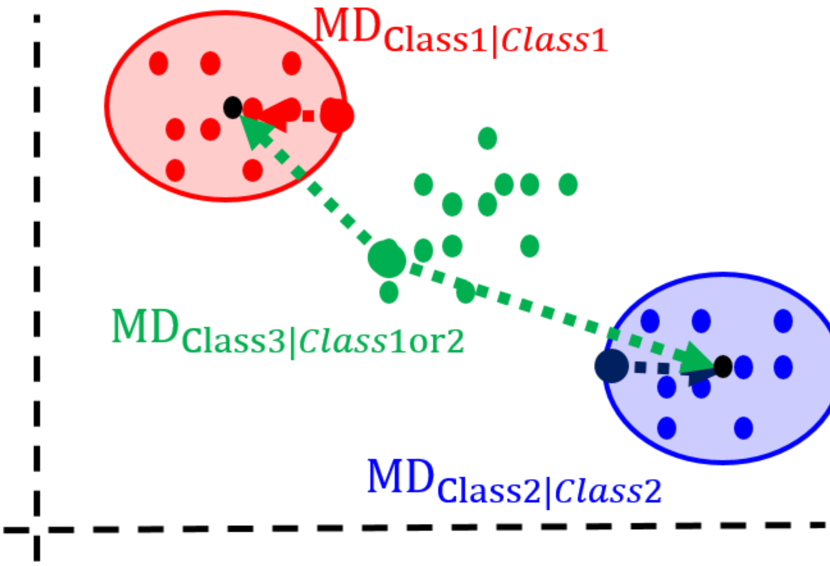

Fig. 21. Mahalanobis distance

$$D^2 = (x-m)^T \cdot C^{-1} \cdot (x-m)$$

- D^2 = Square of the MD
- x = Vector of the obsevation , (1)
- m = Vector of mean values of independent variables
- C^{-1} = Inverse covariance matrix of independent variables.

Our overall algorithm for the outlier detection is shown in the Fig. 22. Using the elliptic envelope, system could estimate the distribution of learned data. After that, system use the MD to determine if the input signal is outlier or not.

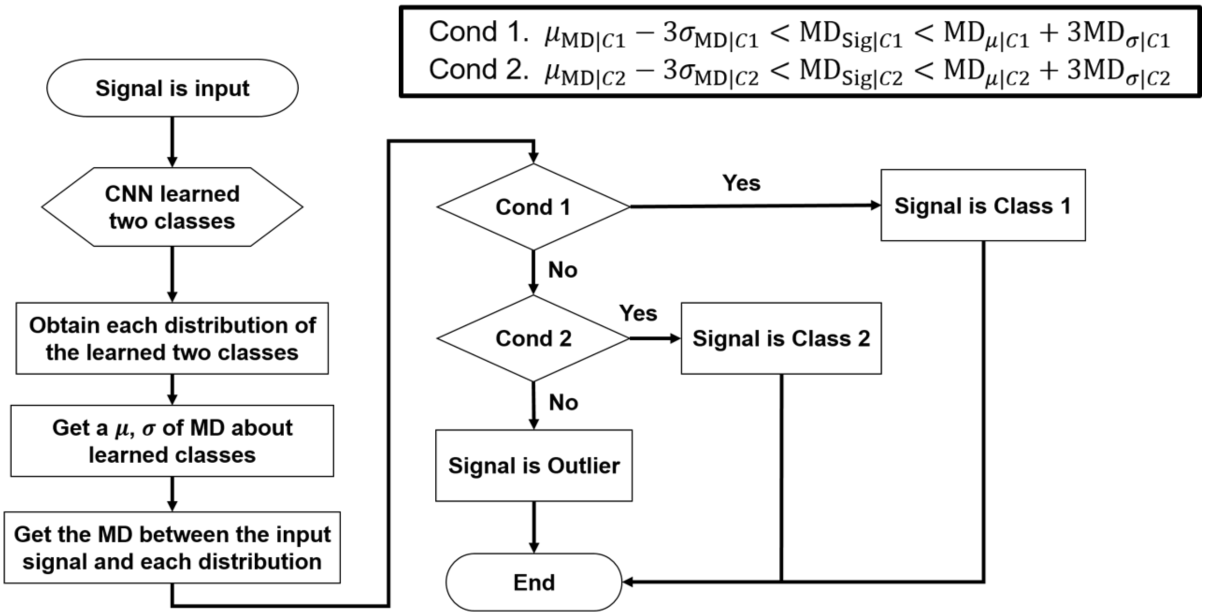

Fig. 22. Algorithm for outlier detection.

3.3. Model Extension

After the outlier detection, there would be model extension part in the system. Using the outlier detection result, the proposed system can learn the outliers for new class. For this learning, we applied incremental learning to system using fine tuning method. We changed the size of output layer and initialized. Then we retrained the classifier while fixing the parameters of network except output layer (Fig 23). Table 2 shows the our retrained CNN property.

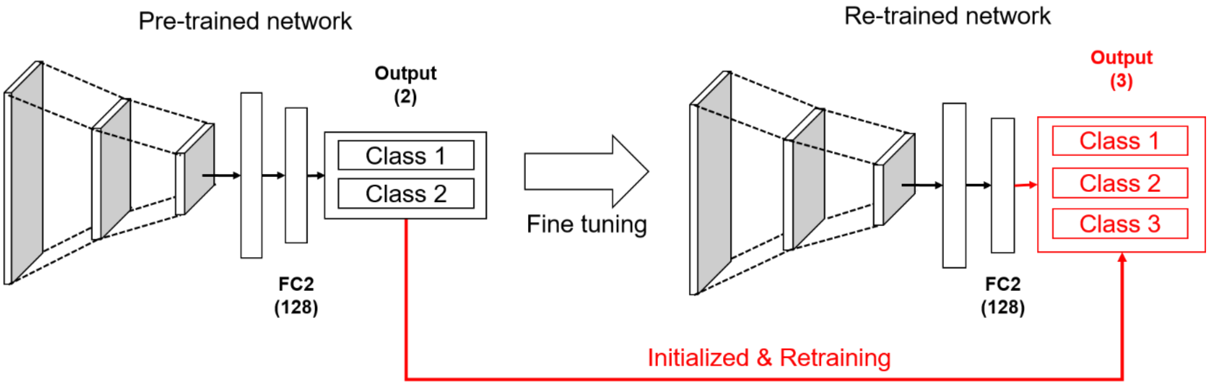

Fig. 23. Fine tuning for incremental learning

Layer	Kernel Size	Output Shape	Parameters	Activation
Input	-	1x257x476	-	-
Conv2D	3x3	64x128x237	640	RELU
Maxpool2D	2x2	64x64x118	-	-
Conv2D	3x3	32x31x58	18464	RELU
Maxpool2D	2x2	32x15x29	-	-
FC1	-	512	7127552	RELU
FC2	-	128	65664	RELU
Output	-	2→3	258→387	-

Table. 2. Retrained CNN Property

4. Experiment

4.1. Experimental system

As we mentioned, we acquired the signal from the 3 military transmitter using radio signal recording system (Fig. 24). As the *push-to-talk* function of a transmitter executed, recording system acquired through wired connection. We sampled the signals as 400MHz. The sampled signals saved to a computer and loaded to MATLAB 2016a.

Fig. 24. Experiment equipment: Radio recording system, PRC-999K

In our experiment, we conducted the test with two datasets: RF data and MNIST data. RF data consists of 170 spectrograms per class (Fig. 25). However, due to the small dataset of RF data, we also used MNIST data (Fig. 26). MNIST data is grey scale hand written digit image with 28x28. Every class in MNIST data has 2000 images. We divided these two datasets by 6:2:2 ratio for each part. We verified our system using MNIST data, also applied in RF data. Our computational environment is based on GTX Titan and Pycharm with Pytorch 1.0.1.

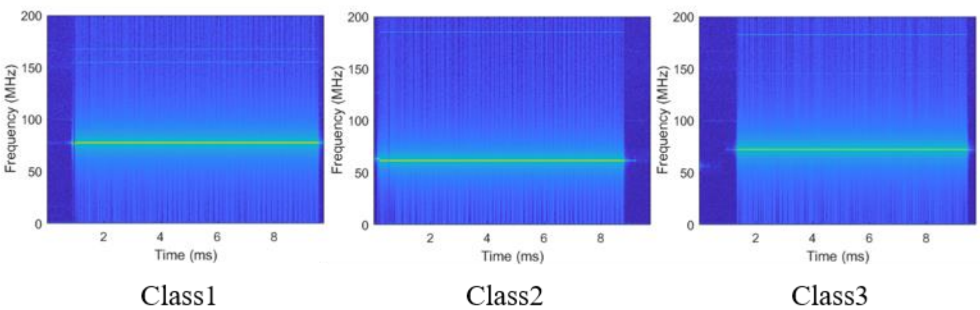

Fig. 25. RF data per each class

Class1 Class2 Class3

Fig 26. MNIST data per each class

4.2. Experimental result

First, we checked the detection ratio in outlier detection part. Detection ratio means that it is classified as the corresponding class. We input the 20% of data for test in this part. Each test class has 34 in RF data and has 400 images in MNIST data.

	RF	MNIST
Class 1	0.70	0.9025
Class 2	0.076	0.8825
Outlier	0.38	0.715

Table. 3. Detection ratio based on output value

	RF	MNIST
Class 1	0.0	0.8675
Class 2	0.0	0.89
Outlier	1.0	0.9525

Table. 4. Detection ratio based on FC2 feature map

	MNIST
Class 1	0.856
Class 2	0.878
Outliers	0.798

Table. 5. Average of detection ratio

Table 3, 4 show the result of outlier detection part when outliers is class 3. The system can detect the outliers by maximum 95% using feature map data. It can be seen that the detection ratio of using feature map is higher than the output value. However, there is bad performance in RF data. This will be explained in result analysis part. Table 5 show the average including when the outliers is class 1 and class 2 based on class rotation in MNIST data.

Using this detection ratio, our system learn the outliers in model extension part. Therefore, we checked the accuracy of the system based on outliers in this part. (Table 6-9). Our system cannot classify the outliers at 2class classifier. Through the model extension, our system get a maximum 96% accuracy on outliers based on MNIST data. However, due to the outlier detection result in RF data, there is also bad performance in model extension part. Table 10 also shows the average of result based on class rotation in MNIST data.

	2 Class Classifier	Proposed System	3 Class Classifier
ACC_{class1}	0.9705	0.9399	0.9411
ACC_{class2}	1.0	0.8874	0.9411
$ACC_{outliers}$	N/A	0.7349	0.9705

Table. 6. Proposed system based on output value of MNIST data

	2 Class Classifier	Proposed System	3 Class Classifier
ACC_{class1}	0.9975	0.8574	0.9874
ACC_{class2}	1.0	0.9224	0.9874
$ACC_{outliers}$	N/A	0.9624	0.9925

Table. 7. Proposed system based on feature map of MNIST data

	2 Class Classifier	Proposed System	3 Class Classifier
ACC_{class1}	0.9705	0.3529	0.9411
ACC_{class2}	1.0	0.7647	0.9411
$ACC_{outliers}$	N/A	0.3529	0.9705

Table. 8. Proposed system based on output value of RF data

	2 Class Classifier	Proposed System	3 Class Classifier
ACC_{class1}	0.9705	0.0	0.9411
ACC_{class2}	1.0	0.0	0.9411
$ACC_{outliers}$	N/A	1.0	0.9705

Table. 9. Proposed system based on feature map of RF data

	2 Class Classifier	Proposed System	3 Class Classifier
ACC_{class1}	0.9849	0.9482	0.9874
ACC_{class2}	0.995	0.9241	0.9874
$ACC_{outliers}$	N/A	0.7149	0.9925

Table. 10. Average of accuracy based on feature map in MNIST data

4.3 Result analysis

There is bad performance in the experiment with RF data. To solve this bad performance, we conducted the one more experiment to check overfitting in this section. RF dataset is so small that can cause the overfitting problem. When the overfitting problem occurs, classification results are good for training data, but are not good for the test data [19]. Considering this problem, we conducted the experiment with training data (Table. 11).

	RF
Class 1	0.8705
Class 2	0.8882
Outliers	0.9823

Table. 11. Detection ratio based on training data

	2 Class Classifier	Proposed System
ACC_{class1}	0.9941	0.8941
ACC_{class2}	0.9882	0.9117
$ACC_{outliers}$	N/A	0.9823

Table. 12. Proposed system based on training data

Through this experiment, we can confirm that proposed system show the good performance on training data. As a result, because the number of samples is not enough, the system judges the test data as outliers in the experiment using test data. Then, we applied early stopping method to solve this overfitting problem (Table 13).

Epoch	1	10	20	40	60	80
Class 1	0.0	0.0	0.0	0.0	0.0	0.0
Class 2	0.0	0.0	0.0	0.0	0.0	0.0
Outliers	1.0	1.0	1.0	1.0	1.0	1.0

Table. 13. Detection ratio in early stopping

Table 13 shows the early stopping method has no effect on training. Therefore, to solve this problem, we will collect more RF data for experiment as a future work.

5. Conclusion

In this paper, we propose the self-learning system in which its aim is to detect outliers and learn by itself without human assistance. To achieve this aim, the proposed system operates by following three steps as follows. First, it uses a learning method to classify given data into pre-defined classes. Second, it uses the proposed detection rule, based on the mahalanobis distance as we have introduced in section 3, to detect outliers when new data is given. Last, the system learns these detected outliers as new class without human assistance.

Through the experience, we confirmed that the system is increasing number of classes without human assistance. In the experiment based on the MNIST data, our proposed rule can detect the outliers by 95%. After this detection, the system learn the outliers and can classify to the new class by 96%. Experiments based on MNIST data show that our system is fully feasible. However, in the experiment based on the RF data, there was a deterioration in performance. This deterioration caused by overfitting problem. We confirmed that our proposed system is overfitting through the experiment based on training data. We also applied early stopping method to solve overfitting problem. However, there is no effect of early stopping method to the system. Therefore, we will improve system performance based on RF data by collecting more RF data samples to solve overfitting problem as a future work.

6. Reference

[1] Y. Jia, S. Zhu, et.al, “Specific emitter identification based on the natural measure,” *Entropy*, vol. 19, pp. 117, 2017.

[2] K. Yang, et.al, "Multimodal Sparse Representation-Based Classification Scheme for RF Fingerprinting," in IEEE Communications Letters, vol. 23, no. 5, May 2019, pp. 867-870.

[3] H. Jafari, et.al, "IoT Devices Fingerprinting Using Deep Learning," MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM), Los Angeles, CA, 2018, pp. 1-9.

[4] K. Merchant, et.al, “Deep learning for Rf device fingerprinting in cognitive communication network,” IEEE, J.Sel.Topics Signal Process., vol. 12, no. 1, Feb. 2018, pp. 160-167.

[5] J. Bassey, et.al, "Intrusion Detection for IoT Devices based on RF Fingerprinting using Deep Learning," 2019 Fourth International Conference on Fog and Mobile Edge Computing (FMEC), Rome, Italy, 2019, pp. 98-104.

[6] S. Liu, et.al, "Six-month-old infants expect agents to minimize the cost of their actions", Cognition, vol. 160, pp. 35-42, 2017. Available: 10.1016/j.cognition.2016.12.007.

[7] H. C. Mandhare, et.al, "A comparative study of cluster based outlier detection, distance based outlier detection and density based outlier detection techniques," 2017 International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, 2017, pp. 931-935.

[8] Ian J. Goodfellow, et.al, Generative adversarial nets. In Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'14), Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (Eds.), Vol. 2. MIT Press, Cambridge, MA, USA, 2014, 2672-2680.

[9] Thomas Schlegl, et.al, ‘Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discover’, IPMI 2017, pp. 146 157.

[10] Mark Kliger, et.al, ‘Novelty Detection with GAN’, ICLR 2018.

[11] Michael McCloskey, et.al, ‘Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem’, Psychology of Learning and Motivation, 1989, Volume 24, pp. 109-165.

[12] Gepperth, Alexander, et.al, “Incremental learning algorithms and applications.” ESANN 2016.

[13] Zhizhong Li, et.al, “Learning without Forgetting" in IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 40, No. 12, 2018.12

[14] Haonan Guo, et.al, ‘Learning Automata Based Incremental Learning Method for Deep Neural Networks’, IEEE Access Special Section on Mission Critical Sensors and Sensor Networks (MC SSN), 2019.05

[15] Youlu Xing, et.al, ‘Perception Evolution Network Based on Cognition Deepening Model adapting to the Emergence of New Sensory Receptor’, IEEE Trans. On Neural Networks and Learning Systems, 2016.03

[16] Yann LeCun, et.al, 1998. Convolutional networks for images, speech, and time series. In The handbook of brain theory and neural networks, Michael A. Arbib (Ed.). MIT Press, Cambridge, MA, USA 255-258.

[17] S. Hershey, et al, "CNN architectures for large-scale audio classification," 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, 2017, pp. 131-135.

[18] L. Maaten, "Visualizing Data using t-SNE", Journal of Machine Learning Research (JMLR), vol. 9, pp. 2579-2605, 2008.

[19] R. Caruana, "Overfitting in neural nets: backpropagation, conjugate gradient, and early stopping", Neural

Information Processing Systems (NIPS), pp.381-387, 2001.