DevoX: Deep Voice Detection MLOps Framework for Crime Prevention in the Core-Edge Cloud

Yuri Seo*
Department of Artificial Intelligence
Kyung Hee University
Yongin-si, South Korea
yuri0329@khu.ac.kr

Hyeon-ki Jo*
Department of Artificial Intelligence
Kyung Hee University
Yongin-si, South Korea
jhk1132@khu.ac.kr

Seol Roh
Department of Computer Science and Engineering
Kyung Hee University
Yongin-si, South Korea
seven800@khu.ac.kr

Teh-Jen Sun
Department of Artificial Intelligence
Kyung Hee University
Yongin-si, South Korea
dlwmnal1@khu.ac.kr

Seung-Woo Jeong
Department of Computer Science and Engineering
Kyung Hee University
Yongin-si, South Korea
jyng2227@khu.ac.kr

HakHo Kim
Department of Artificial Intelligence
Kyung Hee University
Yongin-si, South Korea
rkd2935@khu.ac.kr

Thien-Thu Ngo
Department of Computer Science and Engineering
Kyung Hee University
Yongin-si, South Korea
thu.ngo@khu.ac.kr

Eui-Nam Huh
Department of Computer Science and Engineering
Kyung Hee University
Yongin-si, South Korea
johnhuh@khu.ac.kr

***Abstract*—With the increase in criminal cases using deep voice, research related to deep voice detection is on the rise. However, the previous studies of deep voice detection are still insufficient when applied to actual crime prevention services. To address this issue, this paper proposes a new framework called "DevoX" that can apply deep voice detection to actual services. This framework continually trains on voice data from end devices (such as smartphones) for new types of deep voice crimes. Also, DevoX provides low-latency services through the core-edge cloud architecture. Edge cloud sends extracted features through encoding instead of call voice to prevent privacy leaks. It is possible to apply deep voice detection technology to actual services. Therefore, our study can be effectively utilized in preventing phishing based on deep voice.**

Keywords— core-edge cloud, deep learning, deep voice, mlops

I. INTRODUCTION

Recently, with the increasing use of AI (Artificial Intelligence) in applications, a number of cases where malicious use of AI technology is increasing. For example, crimes such as spreading disinformation [1], manipulating public opinion [2], and voice phishing by impersonating another person [3] are occurring through audio fake technology called deep voice. This has led to deep voice detection being needed to prevent crimes caused by deep voice technology.

Deep voice is a compound word of deep learning and fake voice. In particular, deep voice can be easily used as it can collect various voices through SNS (Social Network Services) such as YouTube, Facebook, and Instagram [4]. Also, the importance of deep voice detection is emerging due to the problem of difficulty in distinguishing between real and fake voices [5]. Additionally, research is increasing to improve deep voice detection accuracy using ML (Machine Learning) or deep learning [6]—[9]. This performance improvement is important from an academic perspective. However, these studies are limited to apply to actual services. First, continuous training for new crime types of deep voice detection was not considered. Second, low-latency inference for real-time service provision was not considered. Third, there is a vulnerability in privacy protection in that voice data is stored in the edge server.

Our study proposes an AI voice detection method and service framework (e.g., DevoX) based on MLOps to prevent deep voice crime. MLOps (Machine Learning Operations) is a paradigm that aims to deploy and maintain ML models reliably and efficiently [10], [11]. Therefore, we apply the MLOps framework for continuous training and efficient deployment of deep voice detection models as follows. Firstly, data ingested from applications are continuously trained with AI models in the core cloud to detect new crime types of deep voices. At this time, by transmitting encoded feature extraction instead of the user's voice call, the risk of privacy leakage can be mitigated. Then, the AI model trained in the core cloud is deployed to the edge cloud. In the edge cloud, deep voice is detected through AI model inference. When the result of inference feedback to the application in the end device, the end device notifies the user via an alert. The end device can provide services to users with low latency through nearby edge clouds. Consequently, by offering deep voice detection as a real-time service, it is expected that the damage from deep voice crime can be minimized.

Thus, our contributions are as follows

- Our study introduces an AI voice detection method and service framework (e.g., DevoX) based on MLOps to address the increasing malicious use of AI technology, particularly in crimes involving deep voice manipulation.
- Provides a reliable deep voice detection model by learning deep voice crime types that are updated through a continuous learning framework.

*Co-author: equal contribution on the paper

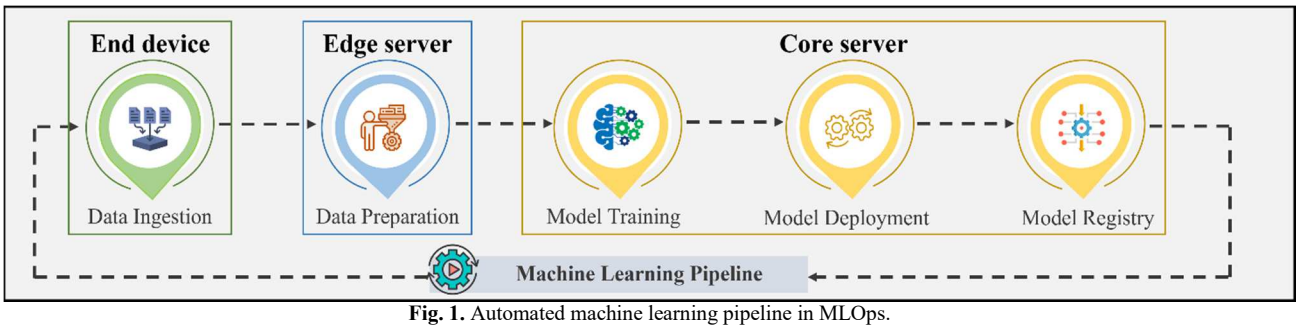

Fig. 1. Automated machine learning pipeline in MLOps.

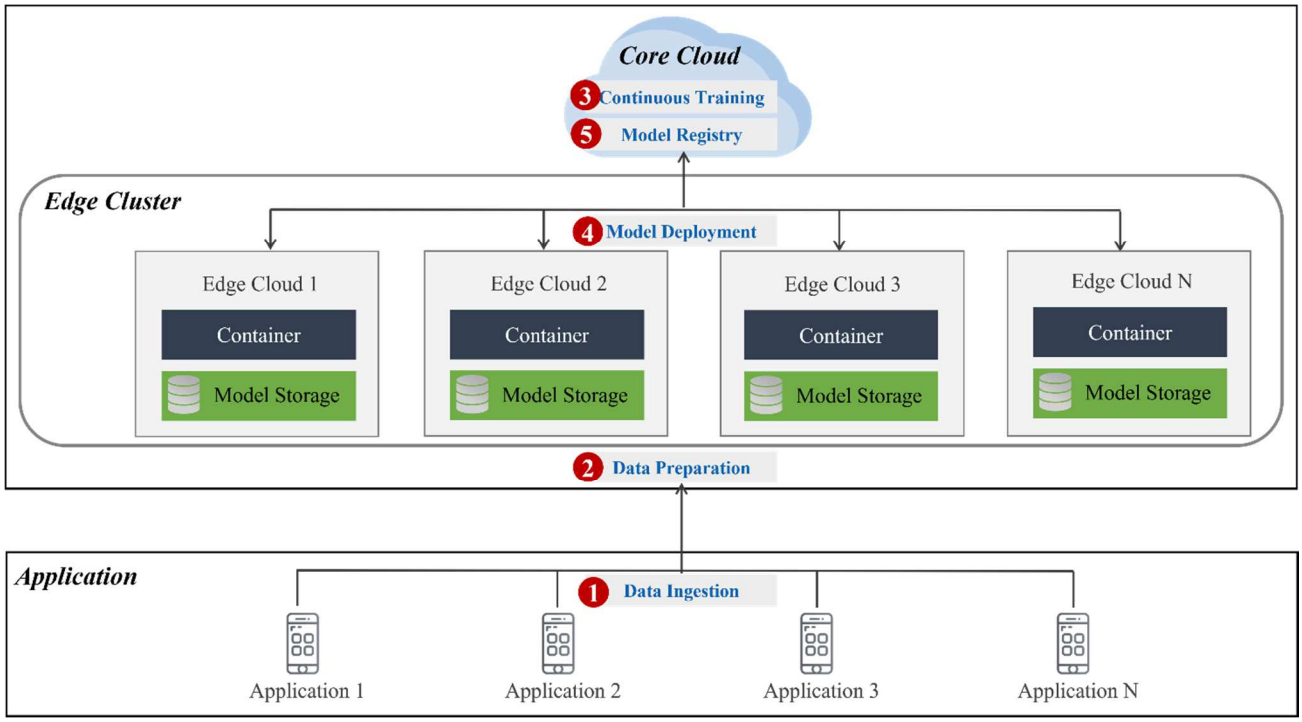

Fig. 2. Proposed MLOps framework architecture in core-edge cloud environment.

- Reduces the risk of personal information leakage by extracting and transmitting only features from voice call data.
- Provides deep voice detection applications to users by training a deep voice detection model and providing inference services.

The remainder of this paper is organized as follows. Section 2 introduces research related to deep voice detection and MLOps. Section 3 explains the proposed framework, and its experimental results are provided in Section 4. Finally, Section 5 concludes and presents directions for future research.

II. Related Work

A. Deep Voice Detection

Deep voice detection refers to detecting manipulated fake voices using deep learning [6]. As the number of cases of deep voice abuse in crime increases, improving deep voice detection accuracy has become a major concern [6]. Accordingly, research focusing on improving accuracy using various deep learning or ML models, including SVM-Linear [7], XGBoost [8], and VGG16 [9], is increasing. Meanwhile, recent research is attempting to apply it to actual service applications. Lim et al. [12] applied explainable AI to deep voice to provide reliability through analysis of the deep voice detection model. Additionally, Kang et al. [5] proposed a new system (i.e., Deep Detection) that can detect deep voice and authenticate users without exposing voice data to the server. These methods can be helpful when applied to actual services related to deep voice detection. However, our study differs from previous studies in that it focuses not only on model accuracy but also on overall services for deep voice crime prevention by considering various factors (continuous training, low-latency services, privacy-enhanced, etc.).

B. MLOps (Machine Learning Operations)

MLOps is a compound word for Machine Learning (ML) and Operations (Ops). MLOps uses an automated pipeline from data ingestion for training machine learning to deployment of the trained model [13]. Additionally, existing studies adapt MLOps to core-edge cloud architecture [14]. MLOps is constructed in such a way that the training process, which requires a lot of computing costs, is carried out in the data center, and the inference process, which requires relatively low costs and fast service, is carried out in the edge server. Then, service costs can be optimized and fast service can be provided to users. In this paper, we provide a deep voice detection service by continuously learning a deep voice detection model in an HPC environment and distributing the trained model to an edge server for inference. Moreover,

recent studies have integrated MLOps into core-edge cloud architecture [14]. MLOps optimizes both the resource-intensive training process in the data center and the service-intensive inference process in the edge server. This approach enables cost optimization for services and provides low-latency service to users. This paper introduces a deep voice detection MLOps framework, achieving continuous learning of a deep voice detection model within an HPC environment and deploying the trained model to an edge server for inference. Fig. 1 shows the automated pipeline in our suggested MLOps framework. At the Data Ingestion stage, the application ingests voice data from the end device. This voice data is recorded at one-second intervals and transmitted. The ingested data is sent to the edge server for Data Preparation. The edge server stores this preparated data and executes deep voice detection models. Also, the prepared data is forwarded to the core server for the Model Training step at specific intervals. In the Model Deployment, the trained model is redeployed to the edge server and managed through a Model Registry.

III. The Proposed Systems

A. Conceptual Framework

Fig. 2. is the proposed framework architecture for MLOps for "DevoX" application services. The overall framework process is automated through the MLOps pipeline and consists of five stages including Data Ingestion, Data Preparation, Model Training, Model Deployment, and Model Registering. Voice data collected from the application are moved to storage in the edge cloud (Data Ingestion, see ①). Data preprocessing and inference are performed in containers in the edge cloud (Data Preparation, see ②). Additionally, the core cloud performs model training using preprocessed data (Model Training, see ③). The core cloud deploys the trained model to the edge cloud (Model Deployment, see ④). Also, the trained models are registered to manage the version of models (Model Registering, see ⑤). A significant advantage of this framework lies in its ability to efficiently distribute tasks through modularization among the three parts of the service, thereby leveraging the unique strengths of each. The framework we propose in this study has three main layers. First, the core cloud trains the model using preprocessed data and deploys the trained model to the edge cloud. Second, the edge cloud focuses on data collection, preprocessing, and inference processes in containers. Third, applications communicate with the edge cloud to send voice data and receive feedback to alert the result to the user.

B. System Process

Fig. 3. shows the architecture of receiving AI services from an application. Applications communicate through a socket container (container A) located at a nearby edge cloud (see ①). Voice data generated during communication is detected in real-time whether it is a real or fake voice through the AI inference server (container B, see ②). If the detected voice is fake, the result is sent to the application (see ③).

C. Dataflow of the use case

To design a scalable MLOps framework for Devox applications, we performed iterative training and evaluation.

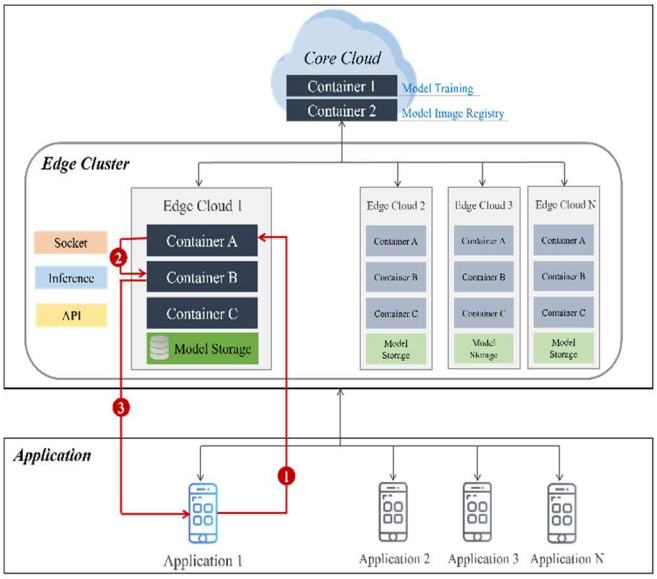

Fig. 3. Procedural architecture for providing AI services to applications.

From the user's perspective, the data flow is observed in three

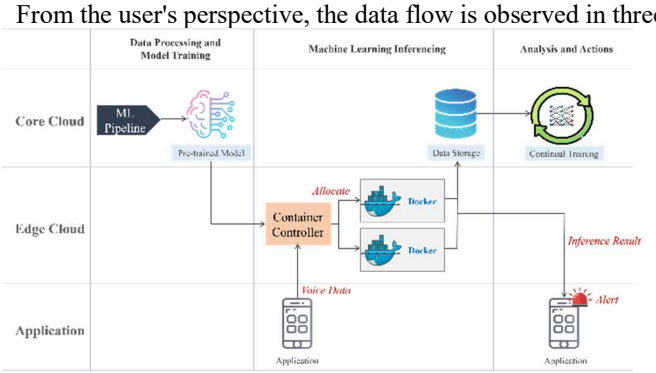

Fig. 4. The dataflow of the use case.

stages, including data processing and training, inference, analysis, and action (Fig. 4). First, the model is trained through the ML pipeline in the core cloud. Second, the trained model is deployed to the container controller in the edge cloud, and an inference container is created to analyze voice data transmitted from the application. Third, the edge container preprocesses data and infers and analyzes deep voice through deployed models. The data used for inference is stored in the data storage of the core cloud, and the deep voice analysis results are provided to users through notifications through the application. Additionally, the core cloud performs continuous training using data stored in the data storage. Continuous training on various types of data can help detect new types of deep voices.

D. Deep voice detection data analysis and model training

A total of 21 hours of voice dataset was segmented into real and fake voices and applied to a deep voice detection model. AIHub emotion classification Korean speech [15] and LJSpeech [16] data used as actual speech.

The fake speech dataset includes WaveFake [17] and the deep voice dataset was generated by the Bark model, which is the text-to-audio model based on the transformer. Bark can produce voices in multiple languages, including Korean, supporting pre-trained voice presets for generating deep voices with varied speakers. It also generates paralinguistic

sounds like diverse emotional expressions, laughter, and sighs. This dataset was labeled and employed to train the deep voice detection model.

The deep voice detection model was implemented based on the Wav2Vec2.0 model, which is mainly used in speech recognition and speech processing tasks [18]. The Wav2Vec model includes feature extraction module and encoder module. The feature extraction module is located in the user application to extract voice features, and the extracted data is transmitted to the edge cloud for inference by the encoder module. This separated design allows the utilization of privacy-enhanced information, applicable to both the core and edge cloud infrastructure. The deep voice detection model samples voices at 8000 Hz for both genuine and synthetic voices. Tokenization is applied to the sampled voices, followed by feature extraction through a CNN-based module. Recognizing that speech often contains non-essential information like noise, using all data might be computationally inefficient. To address this, a feature quantization layer in the model selects and masks crucial information from the extracted feature vectors. This masking involves learning against the transformer block output, creating a dictionary that denotes the importance of information. Ultimately, the trained quantization layer focuses on extracting key information from all features. The feature vector, processed in this manner, proceeds through a transformer-based encoding module and a classifier layer, determining whether it constitutes a deep voice or not.

IV. Experimental Evaluation

A. Deep Voice Detection Model Performance

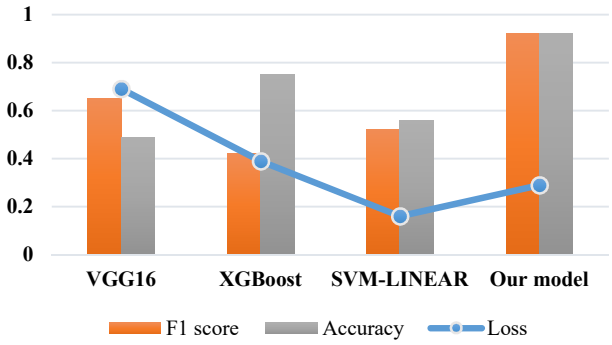

Fig. 5. Deep voice detection model performance comparison.

Our main goal is to apply a deep voice detection model that achieves high accuracy to the service. Fig. 5 shows the results of comparing the accuracy performance between standard models (VGG-16, XGBoost, SVM-LINEAR) and our proposed model. According to the results in Fig. 5, it can be observed that the deep voice detection model achieves the highest performance with an accuracy of 92% when trained with 50 epochs.

B. MLOps

a) Hardware: The hardware specifications used in the core cloud and edge cloud for this experiment are as follows (Table 1).

Table 1. Core cloud and edge cloud hardware spec.

	Core cloud	Edge cloud
CPU	Intel(R) Xeon(R) Gold 6140 CPU @2.30GHz	Intel Core Processor (Broadwell, IBRS)-1 core
GPU	Tesla V100-virtual	-
RAM	160GB	8GB
DISK	1TB	200GB
OS	Ubuntu 18.04.6	Ubuntu 20.04

b) Model training and deployment time in core-edge cloud environment: The total time taken to train and deploy a model using 302 voice data transmitted from the edge server is shown in Table 2. The voice data upload step refers to obtaining training data from the edge server via an API. Since the number of data is fixed at 302, Table 2 shows that all three attempts were processed in a similar time frame. The Data pre-processing step denotes the phase of labeling and preprocessing the data for training. Model training refers to the step of training the model, and in this experiment, the model is trained for 30 epochs. Model deploying denotes the step of deploying the model to the edge server. To deploy the model to all edge servers, it is dispatched in the form of a container image, with an average time of 22 seconds being observed.

Table 2. The data preprocessing, model training, and deployment time measurement in a core-edge cloud environment.

Experiment	API download (s)	Data pre-processing (s)	Model training (s)	Model deploying (s)
1	3.91	0.78	43.78	14.04
2	3.98	0.77	28.28	16.01
3	3.95	0.78	28.73	36.24
Average	3.95	0.78	33.60	22.10

C. Application

Table 3 shows the communication time from the application (end device) to the socket container (edge server) and the AI inference time at the edge server. The voice data sent from the application to the socket container is transmitted in an average of 968 ms. Also, since the size of the voice data is fixed, there is relatively little variation in time. Conversely, the inference time for the deep voice detection model on the edge server varies between 320 ms and 1,717 ms depending on the voice input. The total service time for deep voice feedback is expected to be around 2 seconds on average. However, the edge server on which the experiment was conducted does not have a GPU (Graphics Processing Unit). Thus, if the inference is done on a server equipped with a GPU, the AI inference time could become more consistent and possibly faster.

Table 3. The communication time and the inference time.

Experiment	Application send Time (*)	Socket container receive time (*)	Time difference (ms)	Inference time (ms)
1	07:02:53:431	07:02:54:330	899	1,717
2	07:02:54:480	07:02:55:414	934	320
3	07:02:55:491	07:02:56:485	1,070	1,412
Average	-	-	968	1,150

* Hour : Minute: Second : Millisecond

CONCLUSION

In this study, we propose a new MLOps-based framework that can apply deep voice detection service with low latency called DevoX. The DevoX framework helps to prevent deep voice-related crimes through continuous model training and deployment for various types of crimes. We provide specific descriptions and analysis for each role to highlight their importance in the core-edge cloud architecture. We also provide specifics on the total time taken during the deep voice detection process, including communication of voice data results. The proposed method provides deep voice detection risk alerts to users with lower latency. In future work, we will study not only deep voice but also the overall voice phishing crime detection framework.

ACKNOWLEDGMENT

This work was supported by Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No.2202- 0-00047, Development of Microservices Development/Operation Platform Technology that Supports Application Service Operation Intelligence) and the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2023-RS-2023-00258649) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation).

REFERENCES

[1] J. Lee, and S. Y. Shin, "Something that they never said: Multimodal disinformation and source vividness in understanding the power of AI-enabled deepfake news", Media Psychology, vol.25(4), pp.531–546, 2022.

[2] N. Diakopoulos, and D. Johnson, "Anticipating and addressing the ethical implications of deepfakes in the context of elections", New Media & Society, vol.23(7), pp.2072–2098, 2021.

[3] C. Stupp, "Fraudsters used AI to mimic CEO's voice in unusual cybercrime case", The Wall Street Journal, vol.30(08), 2019.

[4] Y. Rodríguez-Ortega, D. M. Ballesteros, and D. Renza, "A machine learning model to detect fake voice", International Conference on Applied Informatics, Cham: Springer International Publishing, pp. 3–13, 2020.

[5] Y. Kang, W. Kim, S. Lim, H. Kim, and H. Seo, "DeepDetection: Privacy-Enhanced Deep Voice Detection and User Authentication for Preventing Voice Phishing", Applied Sciences, vol.12(21):11109, 2022.

[6] Z. Almutairi, and H. Elgibreen, "A review of modern audio deepfake detection methods: challenges and future directions", Algorithms, vol.15(5):155, 2022.

[7] M. Lataifeh, A. Elnagar, I. Shahin, and A. B. Nassif, "Arabic audio clips: Identification and discrimination of authentic Cantillations from imitations", vol. 418, pp. 162–177, 2020.

[8] J. Khochare, C. Joshi, B. Yenarkar, S. Suratkar, and F. Kazi, "A deep learning framework for audio deepfake detection", Arabian Journal for Science and Engineering, pp. 1–12, 2021.

[9] H. Khalid, M. Kim, S. Tariq, and S. S. Woo, "Evaluation of an audio-video multimodal deepfake dataset using unimodal and multimodal detectors", Proceedings of the 1st workshop on synthetic multimedia-audiovisual deepfake generation and detection, pp. 7–15, 2021.

[10] G. Symeonidis, E. Nerantzis, A. E. Kazakis, and G. A. Papakostas, "Mlops-definitions, tools and challenges", In 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC), pp. 0453–0460, 2022.

[11] S. Roh, K. M. Jeong, H. Y. Cho, and E. N. Huh, "An Efficient Microservices Architecture for MLOps", 2023 Fourteenth International Conference on Ubiquitous and Future Networks (ICUFN), pp. 652–654, 2023.

[12] S. Y. Lim, D. K. Chae, and S. C. Lee, "Detecting deepfake voice using explainable deep learning techniques", Applied Sciences, vol.12(8):3926, 2022.

[13] E. Raj, D. Buffoni, M. Westerlund, and K. Ahola, "Edge mlops: An automation framework for aiot applications", 2021 IEEE International Conference on Cloud Engineering (IC2E), pp. 191–200, 2021.

[14] J. Moon, S. Kum, and S. Lee, "A heterogeneous IoT data analysis framework with collaboration of edge-cloud computing: Focusing on indoor PM10 and PM2. 5 status prediction", Sensors, vol.19(14):3038, 2019.

[15] AIHub, "Korean emotion classification", https://aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&dataSetSn=259.

[16] Keith Ito, "The lj speech dataset," https://keithito.com/LJ-Speech-Dataset/, 2017.

[17] J. Frank, and L. Schönherr, "Wavefake: A data set to facilitate audio deepfake detection", arXiv preprint arXiv:2111, 02813, 2021.

[18] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations", Advances in Neural Information Processing Systems vol. 33, pp. 12449–12460, 2020.