International Journal of Innovative Computing, Information and Control
Volume **21**, Number **5**, October **2025**

ICIC International ©2025 ISSN 1349-4198
pp. 1353–1363

STUDY ON ESTIMATING THE OPERATING STATE OF NC MACHINE TOOLS USING CSI SENSING WITH AN ESP32 MODULE

Kozo Horiuchi[1], Hiroshi Shiratsuchi[2] and Takanori Matsuzaki[2]

[1]Gururi Co., Ltd.
MATSUNAGA bldg. 5F MIKAGE1881, 2-1-7 Uomachi, Kokurakita, Kitakyushu
Fukuoka 802-0006, Japan
kozo.horiuchi@gururi.co.jp

[2]Graduate School of Humanity-Oriented Science and Engineering
Kindai University
11-6 Kayanomori, Iizuka, Fukuoka 820-8555, Japan
{ sira; takanori }@fuk.kindai.ac.jp

Received March 2025; revised June 2025

Abstract. *In this paper, we propose a method for detecting machine tool failures using Channel State Information (CSI) data acquired with an ESP32 module. Previous research used the Linux 802.11n CSI Tool to obtain CSI data. However, the wireless communication cards compatible with this tool are no longer manufactured, necessitating the selection of a new device. To address this, we developed a new sensing system using the ESP32-based Atom S3 module. CSI data was acquired by transmitting Wi-Fi packets at 10 ms intervals and receiving them with a custom program. Using the collected data, the rotation speeds of a machine tool (0 rpm, 6000 rpm, and 12000 rpm) were classified with a Gradient Boosted Decision Tree (GBDT) model. When trained on 500 samples per class, the classification achieved a macro F1 score of 0.69, compared to 0.93 in previous research. Although the accuracy for 6000 rpm was lower, classification for 0 rpm and 12000 rpm remained high. These results confirm that the ESP32 module can acquire CSI data with similar characteristics to previous systems and is a viable alternative for non-contact sensing applications. Future work will address improvements in classification performance, such as utilizing phase information, introducing multiple receivers to reduce environmental noise, and exploring time-series-aware algorithms.*

Keywords: ESP32, Wi-Fi sensing, Channel State Information, Detection of machine tool failures, NC (Numerical Control) machine

1. **Introduction.** Recently, with the advancement of IoT (Internet of Things) technology, high-performance microcontrollers and sensors have become inexpensive and readily available, enabling the efficient accumulation of data sent from a large number of sensors [1]. In addition, the development of AI (Artificial Intelligence) technology has led to active research on forecasting and analysis dealing with large amounts of data [2, 3, 4, 5, 6, 7].

To measure the state of a target using sensors, various insights are required. In particular, in the case of vibration measurement, it is difficult to select the type, number, and installation position of sensors, and experience and understanding of the equipment are important factors.

Therefore, we have focused on a method for sensing vibrations in machine tools based on the effect of the propagation path of wireless communication devices during communication [8], and have conducted a series of studies on detecting equipment conditions and failures using non-contact vibration measurement based on CSI data [9, 10, 11, 12]. These

---

DOI: 10.24507/ijicic.21.05.1353

studies explored various aspects of CSI-based sensing, including visualizing vibrations, evaluating environmental influence, and performing real-time monitoring. In particular, our work in [11] demonstrated that the operational states of an NC machine tool (stopped, idling, cutting) and the types of materials being processed (plastic, aluminum, brass) could be classified using CSI amplitude data obtained via the Linux 802.11n CSI Tool and Intel 5300 NICs. Machine learning models such as Gradient Boosted Decision Trees (GBDT) and Long Short-Term Memory (LSTM) networks were applied, achieving macro F1 scores exceeding 0.93 in controlled environments. However, since this measurement environment relied on discontinued and bulky hardware, it became necessary to develop a more compact, maintainable, and widely deployable alternative CSI sensing system – An issue directly addressed in this research.

IEEE 802.11n and IEEE 802.11ac are currently commonly used Wi-Fi communication standards. These standards use OFDM (Orthogonal Frequency Division Multiplexing) to perform multiplexed communications with multiple subcarriers. Additionally, MIMO (Multiple Input Multiple Output) technology utilizes multiple transmitting and receiving antennas to enhance communication performance. To achieve advanced communications, detailed control that considers the effects of the transmission path is required.

In wireless communications, it is known that transmitted signals are affected by the presence or absence of machines or people in the same space, as well as by physical effects such as reflection, scattering, and attenuation during propagation through space [13]. Information on changes in signals affected by these influences can be obtained as CSI (Channel State Information) from the communication device. CSI provides detailed information about the propagation characteristics of a known signal as it travels from the transmitter to the receiver, and it is essential for faster and more stable Wi-Fi communication [14].

The previous research used the Linux 802.11n CSI Tool [15, 16] to obtain CSI as shown in Figure 1. CSI is a detailed representation of the propagation characteristics of wireless signals, providing information about the effects of physical phenomena such as reflection, diffraction, and multipath attenuation on transmitted signals. These characteristics make CSI an effective tool for vibration measurement and equipment monitoring, as changes in CSI data can be used to infer the vibrations and movements of equipment.

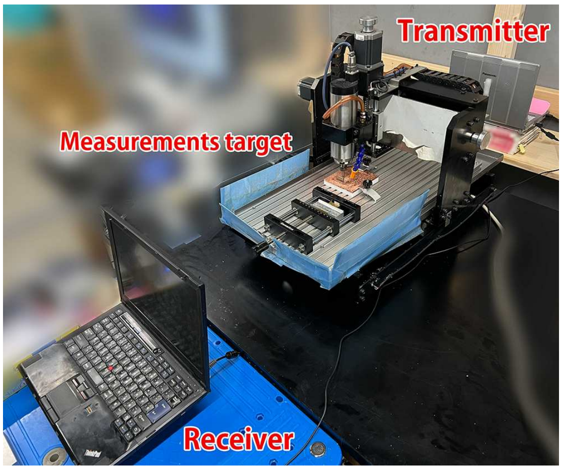

Figure 1. Appearance of current environment

To use this tool, a compatible wireless communication card is required, specifically a PC (Personal Computer) equipped with an Intel Wi-Fi Link 5300 NIC (Network Interface Card). However, this product has been discontinued, and manufacturer support for the

Intel Wi-Fi Link 5300 NIC ended on June 1, 2016 [17]. The measurement tool used in the previous research was adapted to this device, and it is urgent to select a replacement machine and software. Additionally, these devices are bulky laptops that require significant space around the equipment and interfere with machinery operation. Addressing these limitations, alternative wireless communication devices have been explored.

This research focuses on the ESP32 module as a viable alternative due to its cost-effectiveness, compact size, and modern capabilities. Miniaturization of sensing devices not only reduces constraints on installation locations but also makes it easier to deploy multiple devices. Existing research indicates that combining multiple CSI data can estimate spatial information [26]. Moreover, CSI data changes caused by human movement and the operation of other devices can be mitigated by integrating data from multiple devices, thereby improving measurement accuracy.

In this paper, we investigated a new measurement environment and its device that can obtain CSI without relying on the Linux 802.11n CSI Tool. Two Atom S3 devices equipped with ESP32-S3 modules were used: one device transmitted packets at 10 ms intervals, while the other received CSI data. We compared the obtained data to ensure it is comparable to data used in previous research. This paper reports the findings of this comparison and verification.

The remaining part of the paper is organized as follows. Section 2 introduces the challenges in previous measurement environments and explains the selection process for a new device. Section 3 describes the software used for obtaining CSI and its measurement accuracy. Section 4 evaluates the data actually obtained and compares it with data from the previous research. Finally, Section 5 concludes the paper.

2. Challenges and Selection of a New Device.

In our previous research, CSI data was acquired using two laptops equipped with Intel Wi-Fi Link 5300 NICs. While this setup enabled effective data collection, it presented several challenges.

1) The required hardware is discontinued and no longer supported.
2) The laptops are bulky and require significant space for installation.
3) Their presence around machinery can interfere with normal operation.

To address these limitations, we established the following criteria for a new measurement device.

1) It must be a currently supported product.
2) It should be low-cost and readily available.
3) It must be compact and easy to install near operating equipment.

Among the candidates examined, the Raspberry Pi series and ESP32 modules were identified as potential solutions. Each device was evaluated based on its compatibility with the outlined criteria and its ability to meet the demands of modern vibration sensing systems.

As a candidate for a new device, we first consider the Raspberry Pi series. The Raspberry Pi is a small and low-cost single-board computer developed by the Raspberry Pi Foundation in the United Kingdom [19]. It is widely used for various purposes such as educational use and prototype development. As of 2024, the latest model is the Raspberry Pi 5. There are several requirements for obtaining CSI with a Raspberry Pi. Firstly, the standard OS (Operating System) for using a Raspberry Pi is Raspbian, but in its default configuration, it cannot obtain CSI information. Therefore, a patch called Nexmon [20] needs to be applied. Nexmon requires a Broadcom chip, and the following main Raspberry Pi series models meet this requirement:

- Raspberry Pi 3
- Raspberry Pi Zero W
- Raspberry Pi Pico W
- Raspberry Pi B3+
- Raspberry Pi B4
- Raspberry Pi 5
- Raspberry Pi Zero 2 W

Even within these series, the ability to obtain CSI can vary depending on the OS and kernel version, so caution is necessary.

Secondly, the network interface must be switchable to monitor mode. According to the Nexmon documentation, the Raspberry Pi 5 and Zero 2 W cannot be switched to monitor mode. Therefore, based on these considerations, we excluded the Raspberry Pi from our candidates for this research.

Next, we consider the ESP32 module. The ESP32 module is a low-cost microcontroller module with Wi-Fi and Bluetooth capabilities developed by Espressif Systems [21]. It is specifically designed for developing IoT (Internet of Things) devices and, like the Raspberry Pi, is widely used. There are two different libraries that can be used to obtain CSI with the ESP32: the ESP32 CSI Toolkit [22, 23] and ESP-CSI [24]. In this paper, we chose the ESP-CSI library provided by Espressif Systems, the developer of the ESP32 module. Additionally, the ESP-CSI library is used based on the ESP-IDF (Espressif IoT Development Framework) [25]. We selected ESP-CSI over the ESP32 CSI Toolkit because it is officially supported by Espressif, well-documented, and actively maintained. Moreover, ESP-CSI is fully integrated with the ESP-IDF framework, which ensured better compatibility, stability, and ease of development for this research.

The ESP32 module is divided into several series, and the series supported by ESP-IDF are as follows:

- ESP32 Series
- ESP32-S2 Series
- ESP32-S3 Series
- ESP32-C3 Series

Among these, the ESP32-S3 series is considered to deliver the best performance in terms of processing power. There are many kits equipped with these ESP32 modules, but as of 2024, the M5Stack series is a product that is readily available in Japan and offers high expandability. As shown in Figure 2, the M5Stack series is very compact in size and meets the three conditions mentioned earlier for a measurement device. Therefore, for this

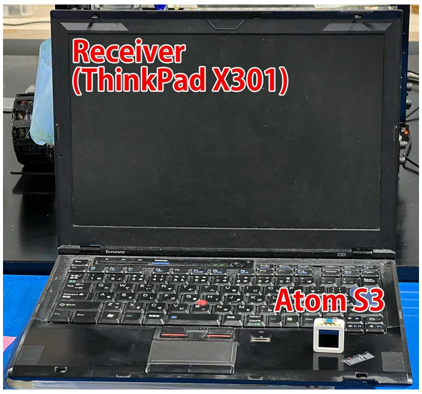

Figure 2. Current device and Atom S3

paper, we will adopt the Atom S3, which is equipped with an ESP32-S3 series chip from this product line.

3. **Data Obtaining Method.** In this paper, we use two Atom S3 units as shown in Figure 3. One unit is programmed with "csi_send" from ESP-CSI, acting as an access point and broadcasting packets at set intervals. The other unit is programmed with "csi_recv" to verify that CSI information can be received.

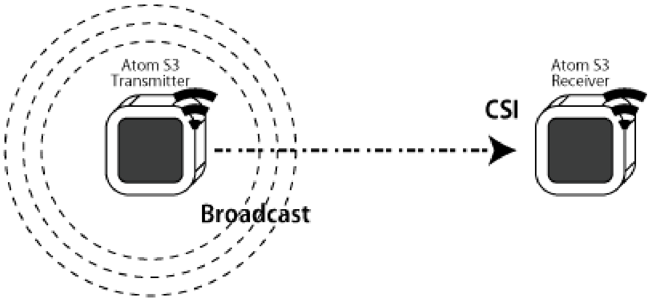

FIGURE 3. Getting CSI using device-to-device communication

Based on the previous research [11], to verify machine vibrations, data must be recorded with a minimum sampling interval of 10 [ms]. Therefore, the transmission interval is set to 10 [ms] in the compile options of "csi_send" to broadcast communication packets. When data was acquired under these conditions for approximately 9.28 seconds, 681 data points were obtained. Given that data was collected every 10 [ms] for about 9.28 seconds, there should have been 928 data points, but only 681 data points were actually obtained. This indicates that approximately 26.61% of the data was lost.

Investigating the cause of the high data loss, it was found that the default "csi_recv" program provided as a sample in the ESP-CSI library was writing data to the serial console each time a packet was received. This behavior became a performance bottleneck that made it difficult to achieve the 10 [ms] data acquisition interval required to match the conditions of our previous research [11]. To address this, we modified the program to buffer 1000 packets internally and output them in a single batch once the buffer was full. This significantly reduced the frequency of serial transmissions and improved throughput. As a result, 1000 packets were successfully obtained in approximately 10.12 seconds, compared to only 681 in 9.28 seconds under the default behavior. Considering the expected 1012 packets over that duration, the data loss was reduced to approximately 1.18%.

In the previous research [11] using the Linux 802.11n CSI Tool, a data loss of a few percent was also observed. Therefore, it was confirmed that the newly selected measurement device can obtain data with a similar level of accuracy.

This batch-output approach not only enhanced stability but also enabled reliable CSI acquisition at the sampling rate required to evaluate the ESP32-based system as a viable alternative to the previous environment. This concept is analogous to combining multiple rows into a single SQL INSERT statement – rather than issuing them individually – to reduce processing overhead and improve throughput.

4. Review and Discussion of the Obtained Data.

4.1. **Adjustment of data.** Having confirmed that the Atom S3 can obtain data with characteristics comparable to those reported in our previous research [11], measurements will be conducted in a new environment using the Atom S3 module, replicating the measurement layout and conditions reported in our previous research [11], as shown in Figure

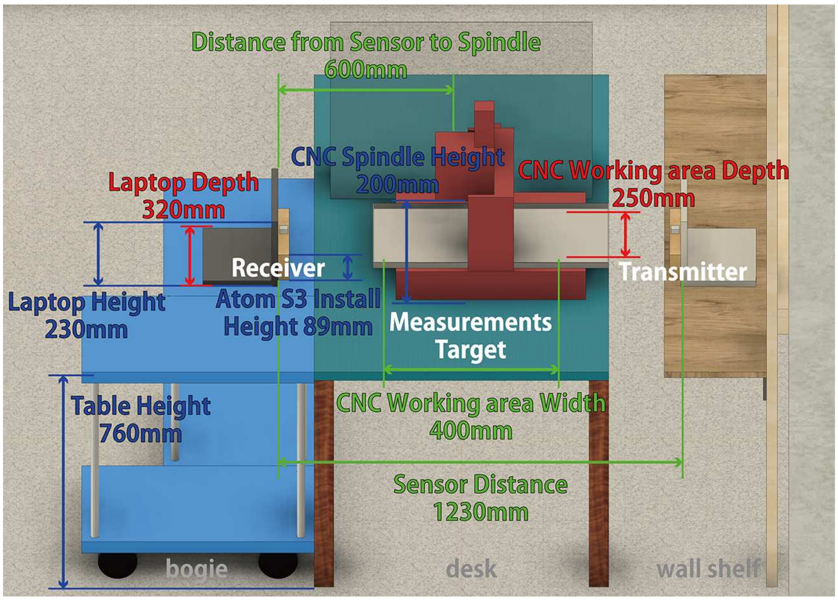

Figure 4. Measurement environment using Atom S3

4, where the height direction is represented in blue, the depth direction in red, and the width direction in green. The room where the measurements are taken is a laboratory in a reinforced concrete building, with dimensions of 8500 [mm] in width, 4120 [mm] in depth, and 3400 [mm] in height. The NC machine tool is installed on a desk in the room, which has dimensions of 760 [mm] in height, 900 [mm] in depth, and 1800 [mm] in width. The distance between the wireless communication devices is set to 1230 [mm], consistent with previous research. To ensure the 1230 [mm] spacing, additional stands are prepared in front of and behind the desk, on which the wireless communication devices are placed. Both Atom S3 devices are installed at a height of 89 [mm] from the desk, and the spindle of the NC machine tool rotates at a position 200 [mm] from the desk and 600 [mm] from the receiver.

The measurement patterns will follow three patterns as described in the previous research [11]. Table 1 shows a description of these patterns. Additionally, only data from 30 subcarriers will be used. This allows us to confirm whether evaluations can be performed similarly to the previous research.

Table 1. Measurement patterns

Operating condition	NC machine tool condition
0 [rpm] (No rotation)	NC machine tool is turned on but not rotating. In idling state.
6000 [rpm]	The spindle is rotating at 6000 rpm. It is not cutting any material.
12000 [rpm]	The spindle is rotating at 12000 rpm. It is not cutting any material.

The data will be obtained by saving it on a PC connected to the receiver. As mentioned in the previous chapter, the receiver buffers 1000 lines of data and outputs it to the serial console once the buffer is full. The obtained CSI data is represented as complex numbers, resulting in two items per subcarrier. Therefore, one line consists of 60 items, and 1000 lines make up one file.

In this paper, measurements were taken over several days, resulting in 1118 files of data at 0 [rpm] (956 training files, 162 evaluation files), 1164 files of data at 6000 [rpm] (987 training files, 177 evaluation files), and 1127 files of data at 12000 [rpm] (956 training files, 171 evaluation files). These files were divided and used as training and evaluation datasets.

The machine learning model used for evaluation was a GBDT model. We used the Python-based implementation developed in our previous research [11] on Google Colaboratory.

In the previous research [11], experiments were conducted using 500 training files and 50 evaluation files for each rotational speed. These numbers were selected to balance the computational load and ensure robust model training and evaluation. Therefore, the training data was prepared using the following procedure:

1) Combine the file indexes of the training data for each rotational speed.
2) Randomize the combined data.
3) Extract the first 500 files from the randomized data.

The evaluation data was obtained using a similar procedure, resulting in 50 files.

4.2. **Evaluation of data.** First, we present the evaluation results obtained using the same number of training and evaluation files as in the previous research [11], shown in Figure 5. The evaluation results by the GBDT model were able to classify all states. In particular, “0 [rpm]” was classified with 100% accuracy. Next, “12000 [rpm]” was classified with approximately 96%, and “6000 [rpm]” was classified with around 52%. The most common misclassification occurred when “6000 [rpm]” was classified as “12000 [rpm]” with a probability of approximately 25%. The F1 score (macro) for this evaluation was 0.69. In the previous research, “0 [rpm]” was classified with 100%, “6000 [rpm]” with about 90%, and “12000 [rpm]” with about 90%, yielding an F1 score (macro) of 0.93. Overall, it was confirmed that a similar evaluation could be conducted. However, there are differences in the probabilities compared to the previous research. One possible reason for these differences could be the surrounding conditions during the measurements, such as human movement, temperature, and humidity. Especially regarding human movement, as utilized in the study by Tokioka et al. [13], it can influence the obtained data.

Next, the results of the evaluation with the increased number of training data for each rotational speed to 950 files are shown in Figure 6. Having 950 files means that the

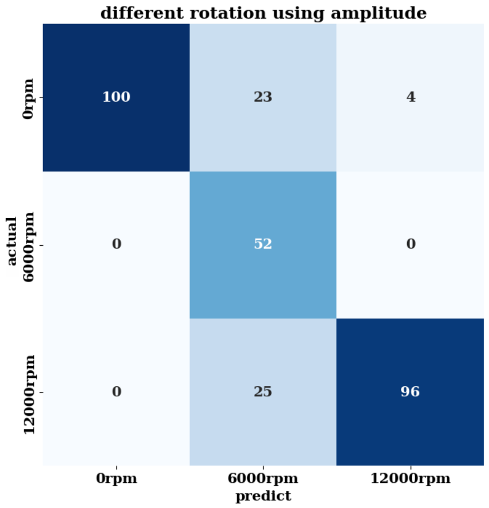

Figure 5. 500 files of training data, 50 files of evaluation data

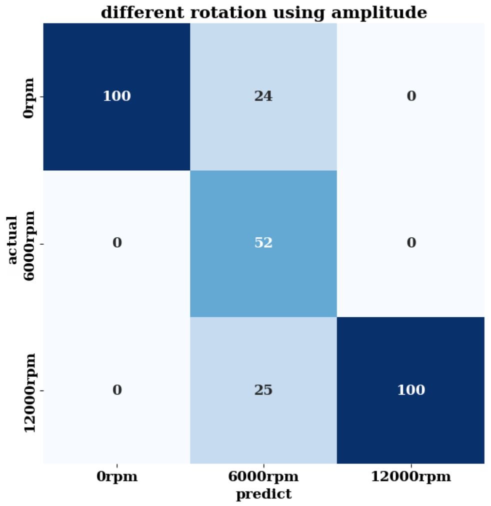

FIGURE 6. 950 files of training data, 50 files of evaluation data

available training data for each rotational speed was standardized to the same number, ensuring efficient utilization of the data. We examined how increasing the amount of training data would affect the evaluation.

In this evaluation, all states were classified successfully. When compared to the measurements in Figure 5, the classification accuracy for "12000 [rpm]" increased to 100%, but no significant overall change was observed. The F1 score (macro) for this evaluation is 0.69.

Finally, the results of the evaluation with the training data for each rotational speed reduced to 330 files are shown in Figure 7. This evaluation aimed to verify whether similar classification accuracy could be achieved with fewer training data compared to the previous research.

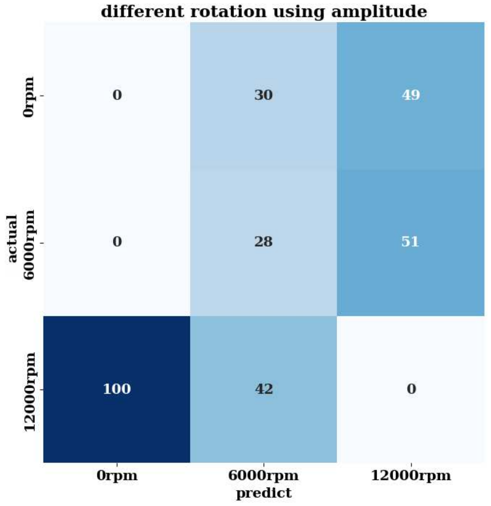

FIGURE 7. 330 files of training data, 50 files of evaluation data

In these conditions, accurate classification could not be achieved. Specifically, for "0 [rpm]", the classification was 100% incorrect. The F1 score (macro) for this evaluation was 0.11.

From these results, it has been confirmed that when a sufficient number of training data files are prepared, the classification accuracy is comparable to that of the previous research. It was also confirmed that with the model used in this research, classification accuracy significantly deteriorates when the number of training data files is reduced to 330. Therefore, it has been verified that measurements using the ESP32 module can be utilized as a device in the new measurement environment for future research.

5. **Conclusion.** In this paper, we investigated the applicability of ESP32 modules as an alternative measurement environment for detecting the operational state of NC machine tools using Channel State Information (CSI).

First, we confirmed that the ESP-CSI framework running on the Atom S3 device was able to acquire CSI data with characteristics similar to those obtained using the Linux 802.11n CSI Tool, which is no longer supported. Based on the acquired data, we classified the NC machine tool's rotational states (0 [rpm], 6000 [rpm], and 12000 [rpm]) with a Gradient Boosted Decision Tree (GBDT) model. When using 500 training samples per class, the model achieved high accuracy for 0 [rpm] and 12000 [rpm], but the classification accuracy for 6000 [rpm] was relatively low, resulting in a macro F1 score of 0.69, compared to 0.93 in the previous research [11]. We also confirmed that increasing the training data to 950 samples per class did not significantly improve the overall score, while reducing the training data to 330 samples led to a considerable drop in classification performance (F1 score of 0.11).

These results indicate that the ESP32-based environment is a promising alternative to the legacy Linux 802.11n CSI Tool for CSI data acquisition and basic classification tasks. Although the classification performance for 6000 [rpm] was lower than expected, the system reproduced similar behavior for other states and achieved stable CSI acquisition with minimal data loss.

The current classification results should be interpreted as reference values to validate the potential of the new measurement system, rather than final performance indicators. We recognize that external noise – such as ambient wireless signals or human movement – can affect classification accuracy in general, regardless of the operational state.

Future research will aim to improve classification robustness by incorporating CSI phase information in addition to amplitude, using multiple receiver devices to suppress environmental noise, and exploring time-series-aware algorithms that can better capture temporal dynamics. Optimizing data collection strategies with consideration for information freshness and redundancy, as discussed in recent IoT research [26], may offer potential improvements in the efficiency of time-series data acquisition and processing in our system.

We also plan to address more practical and realistic scenarios, such as classifying the state of the tool while it is actively cutting material, and detecting the condition of the tool – whether it is new or worn. These directions are important for applying the system to real-world production environments.

Acknowledgment. This research was supported by JSPS KAKENHI Grant Number 19KK0257.

REFERENCES

[1] S. M. Hernandez and E. Bulut, WiFi sensing on the edge: Signal processing techniques and challenges for real-world systems, *IEEE Communications Surveys and Tutorials*, vol.25, no.1, pp.46-76, 2022.

[2] M. S. Mahdavinejad, M. Rezvan, M. Barekatain, P. Adibi, P. Barnaghi and A. P. Sheth, Machine learning for Internet of Things data analysis: A survey, *Digital Communications and Networks*, vol.4, no.3, pp.161-175, 2018.

[3] M. Mohammadi, A. Al-Fuqaha, S. Sorour and M. Guizani, Deep learning for IoT big data and streaming analytics: A survey, *IEEE Communications Surveys and Tutorials*, vol.20, no.4, pp.2923-2960, 2018.

[4] F. Zantalis, G. Koulouras, S. Karabetsos and D. Kandris, A review of machine learning and IoT in smart transportation, *Future Internet*, vol.11, no.4, 94, 2019.

[5] R. H. Hariri, E. M. Fredericks and K. M. Bowers, Uncertainty in big data analytics: Survey, opportunities, and challenges, *Journal of Big Data*, vol.6, Article no.44, 2019.

[6] J. B. Jane and E. N. Ganesh, Big data and Internet of Things for smart data analytics using machine learning techniques, *Proc. of the International Conference on Computer Networks, Big Data and IoT*, pp.213-223, 2020.

[7] S. F. Chevtchenko, E. Da Silva Rocha, M. C. M. Dos Santos, R. L. Mota, D. M. Vieira, E. C. De Andrade and D. R. B. De Araujo, Anomaly detection in industrial machinery using IoT devices and machine learning: A systematic mapping, *IEEE Access*, vol.11, pp.128288-128305, 2023.

[8] S. Jian, S. Ishida and Y. Arakawa, Initial attempt on Wi-Fi CSI based vibration sensing for factory equipment fault detection, *Adjunct Proceedings of the 2021 International Conference on Distributed Computing and Networking*, pp.163-168, 2021.

[9] T. Matsuzaki, S. Ueyama, H. Matsuoka, K. Horiuchi and H. Shiratsuchi, Preliminary evaluation for application of machine vibration detection using Wi-Fi sensing to machine failure prediction system, *ICIC Express Letters, Part B: Applications*, vol.14, no.8, pp.813-819, 2023.

[10] T. Matsuzaki, A. Satoh, R. Fukumoto, K. Kubo, T. Kamiuezono, K. Horiuchi and H. Shiratsuchi, Evaluation of vibration measurement method for NC machine tools using Wi-Fi sensing, *ICIC Express Letters, Part B: Applications*, vol.15, no.9, pp.913-920, 2024.

[11] T. Matsuzaki, S. Kato, Y. Asano, Y. Sako, K. Horiuchi and H. Shiratsuchi, Evaluation of a method for estimating the operating state of NC machine tools using CSI sensing, *Proc. of the 36th Annual Conference of Biomedical Fuzzy Systems Association (BMFSA2023)*, Tokyo, Japan, p.83, 2023.

[12] T. Matsuzaki, S. Kato, Y. Asano, Y. Sako, K. Horiuchi and H. Shiratsuchi, Evaluation of machine tool operating state estimation using channel state information, *Proc. of the 22nd IEEE/ACIS International Conference on Software Engineering, Management and Applications (SERA 2024)*, Honolulu, HI, USA, 2024.

[13] W. Tokioka, H. Yanagimoto and K. Hashimotom, Indoor position estimation with CSI, *The 37th Annual Conference of the Japanese Society for Artificial Intelligence*, 2M6GS1005, 2023.

[14] E. Akay, E. Sengul and E. Ayanoglu, Performance analysis of beamforming for MIMO OFDM with BICM, *IEEE International Conference on Communications*, vol.1, 2005.

[15] D. Halperin, W. Hu, A. Sheth and D. Wetherall, Tool release: Gathering 802.11n traces with channel state information, *ACM SIGCOMM Computer Communication Review*, vol.41, no.1, p.53, 2011.

[16] D. Halperin, W. Hu, A. Sheth and D. Wetherall, *Linux 802.11n CSI Tool*, https://dhalperi.github.io/linux-80211n-csitool/, Accessed on March 29, 2022.

[17] Intel Corporation, *Customer Support Options for Discontinued Intel Wireless Products*, https://www.intel.com/content/www/us/en/support/articles/000006507/wireless.html, Accessed on March 29, 2022.

[18] Y. Ohashi, S. Saeki, T. Sasaki and M. Kawahara, Indoor human location estimation using Wi-Fi CSI of multiple devices, *Proc. of the 85th National Convention of IPSJ*, 6ZA-02, 2023.

[19] Raspberry Pi Ltd., *Raspberry Pi*, https://www.raspberrypi.com/, Accessed on March 21, 2024.

[20] Secure Mobile Networking Lab, *Nexmon*, https://github.com/seemoo-lab/nexmon, Accessed on March 15, 2024.

[21] Espressif Systems, *ESP32 Wi-Fi and Bluetooth SoC*, https://www.espressif.com/en/products/socs/esp32, Accessed on August 14, 2023.

[22] S. M. Hernandez and E. Bulut, Lightweight and standalone IoT based WiFi sensing for active repositioning and mobility, *2020 IEEE 21st International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, Cork, Ireland, pp.277-286, 2020.

[23] S. M. Hernandez and MoWiNG Lab, *ESP32 CSI Toolkit*, https://stevenmhernandez.github.io/ESP32-CSI-Tool/, Accessed on March 29, 2023.

[24] Espressif Systems, *ESP-CSI*, https://github.com/espressif/esp-csi, Accessed on May 24, 2024.

[25] Espressif Systems, *ESP-IDF (Espressif IoT Development Framework)*, https://github.com/espressif/esp-idf, Accessed on April 4, 2024.

[26] W. Hao, H. Ye and F. Huang, Data collection algorithm for Internet of Things based on age of information and sample extrusion awareness, *International Journal of Innovative Computing, Information and Control*, vol.18, no.2, pp.497-509, 2022.

Author Biography

Kozo Horiuchi received his Master of Engineering degree from Kindai University, Japan, in 2005. He completed the coursework for the doctoral program at the same university in 2008, without obtaining a degree. In 2011, he co-founded Gururi Co., Ltd., Japan, and has served as its director and CTO. Since 2014, he has been a part-time lecturer at Kindai University, Japan. His main research interests are computer systems and embedded systems. He is a member of IEICE.

Hiroshi Shiratsuchi graduated with a Doctor of Information Engineering from Kyushu Institute of Technology, Japan, in 1999. He has been a research assistant at the University of the Ryukyu, Japan since 1999, a lecturer at Kindai University, Japan since 2005, an associate professor in 2009, and a professor from 2024 to the present. His main research interests are information and communication and intelligent informatics. He is a member of IEICE.

Takanori Matsuzaki received his Doctor of Engineering from Kyushu University, Japan, in 2006. He was a research assistant at Kyushu University, Japan, in 2003, a lecturer at Kindai University, Japan, in 2006, an associate professor in 2015, and a professor from 2024 to the present. His main research interests are computer architecture and embedded systems. He is a member of IEICE, IPSJ and BMFSA.