Technical Note

Predictive Maintenance Platform with Sound Stream Analysis in Edges

Yoji Yamato[1,a]   Yoshifumi Fukumoto[1]   Hiroki Kumazaki[1]

Abstract: Recently, progress has been made in IoT technologies and applications in the maintenance area are expected. However, IoT maintenance applications are not widespread in Japan yet because of the one-off solution of sensing and analyzing for each case, the high cost collecting sensing data and insufficient maintenance automation. This paper proposes a maintenance platform which analyzes sound data in edges, analyzes only anomaly data in cloud and orders maintenance automatically.

Keywords: Jubatus, predictive maintenance, IoT, sound stream, cloud computing, Industrie 4.0,

1.   Introduction

Recently, progress has been made in IoT (Internet of Things) technologies and cloud technologies such as Refs. [1], [2], [3]. IoT application areas are wide but in particular manufacturing and maintenance are regarded as the most likely application areas which Industrie 4.0 also targets. We can visualize the statuses of factories, facilities and products by collecting and analyzing sensor data on a cloud, and we can also monitor production effectiveness, reflect a production plan, control logistics, change defective products to optimize the supply chain and to accelerate manufacturing and maintenance businesses.

IoT platforms also appeared to develop and operate IoT applications effectively such as Ref. [4]. However, existing IoT platforms mainly target to visualize things statuses of a fixed period by batch processing of collected sensor data and those are insufficient to accelerate maintenance businesses. Specifically, current problems are finding a one-off solution for sensing and analyzing each case, high cost of collecting sensor data and insufficient maintenance automation. Those increase in deployment/operation costs, users in Japan have hesitated so far to adopt IoT technologies.

This paper targets maintenance of business machines in factories. To resolve existing problems, we propose a maintenance platform which analyzes sound data collected by microphones attached to business machines in edges, analyzes only anomaly data and orders maintenance automatically using service coordination technologies such as Refs. [5], [6], [7] in a cloud. We also implement a sample application. The work of Ref. [8] is our previous work and this paper improves on it.

2.   Problems of Existing Technologies

AWS IoT [9] is a platform to integrate Amazon Web Services for IoT applications. For example, users can deliver IoT data to Amazon cloud via MQTT (MQ Telemetry Transport) protocol by Amazon Kinesis and users can analyze IoT big data using various methods such as regression or classification by Amazon Machine Learning.

NTT DOCOMO and GE release an IoT solution which provides GE's industrial wireless router Orbit with NTT DOCOMO's communication module in 2015 [4]. Users can collect operation statuses of facilities by setting Orbit. Moreover, users can develop IoT applications on Toami which is an IoT cloud platform provided by NTT DOCOMO and enables visualization of collected data easily.

In Japan, VEC (Virtual Engineering Community) [10] is an organization to promote manufacturing solutions and VEC discusses the application of IoT technologies to factories actively. VEC aims to detect failures of facilities automatically from sensing data based on predefined rules and thresholds. VEC expects IoT monitoring to improve restoration support and working effectiveness in factories.

Moreover, many IoT platforms have appeared such as Google Brillo [11], Microsoft Azure IoT services [12] and Sakura IoT platform [13].

But, to adopt these platforms for maintenance, there are three problems.

The first is that the sensing/analyzing method is one-off for each case. In other words, users need different sensor, configuration and analyzing methods for each machine. In VEC community, defining rules and thresholds to judge failures is difficult at present. Moreover, to adapt each situation is also unresolved because appropriate thresholds are different in each environment, season or age.

The second is that the cost of collecting sensor data is high. In

---
[1]   NTT Software Innovation Center, NTT Corporation, Musashino, Tokyo 180–8585, Japan
[a]   yamato.yoji@lab.ntt.co.jp

AWS IoT, to analyze IoT data, users need to collect all data to a cloud and need a network for many machines in multiple regions. For example, when a mobile or satellite communication is used to collect a business vehicle's sensing data, network cost is huge.

The third is that maintenance automation is insufficient. Reference [4]'s IoT applications developed on Toami are mainly visualize applications of collected data by batch processing. Therefore, applications which take real-time actions such as repair parts orders based on analyzed data are not considered.

3. Proposal of Maintenance Platform with Sound Stream Data Analysis in Edges

3.1 Ideas to Solve Existing Problems

For a one-off solution for sensing and analyzing, we propose a method of analyzing sound stream data by Jubatus [14] to enhance generality. It is difficult to attach various sensors to existing machines in factories because they may need modifications. However, sound data can be collected from outside machines, as it is easier to attach microphones near machines. Unlike batch processing of Hadoop, Jubatus is a machine learning framework suitable for the sequential processing of stream data. Because the sound frequency spectrum of each business machine in usual operation is fixed, anomaly of unusual state can be detected by analyzing spectrum data. And because Jubatus can use anomaly detection or a classification algorithm which detects differences from usual values, we can deploy Jubatus to each site without detailed rules or threshold settings like VEC.

To deal with the high cost of data collection, we propose a method of running Jubatus in edges to filter the usual sensing data. By sending only suspicious data including raw data which Jubatus judge results, we can reduce the network cost. Because Jubatus only runs simple anomaly detection or classification in edges, we can use cheap micro computers such as Raspberry Pi [15] for edge nodes.

For maintenance automation, we propose a method of distributing data via in-memory DB such as VoltDB [16]. In-memory DB can process faster than disk DB and reflect the analysis of results to a backend system such as ERP (Enterprise Resource Planning) to order repair parts quickly. In general, there are already system coordination technologies (e.g., Refs. [17], [18], [19]) but DB of information system and backend system are different and data transfer between both DBs is done by batch processing because of disk performance bottleneck. We can adopt in-memory DB at reasonable cost because we greatly reduce inserted data by using Jubatus.

3.2 Proposed Platform Architecture

Figure 1 shows the proposed platform architecture based on the above ideas. Figure 1 architecture has the following functions.

In an edge site such as a factory, sensors and network gateways are deployed to send sensing data. Raspberry Pi with Jubatus installed is one of example of a gateway. Sensing data is sent via a closed VPN network by MQTT protocol.

In a cloud data center, cloud maintenance applications are deployed and controls collect data or maintenance.
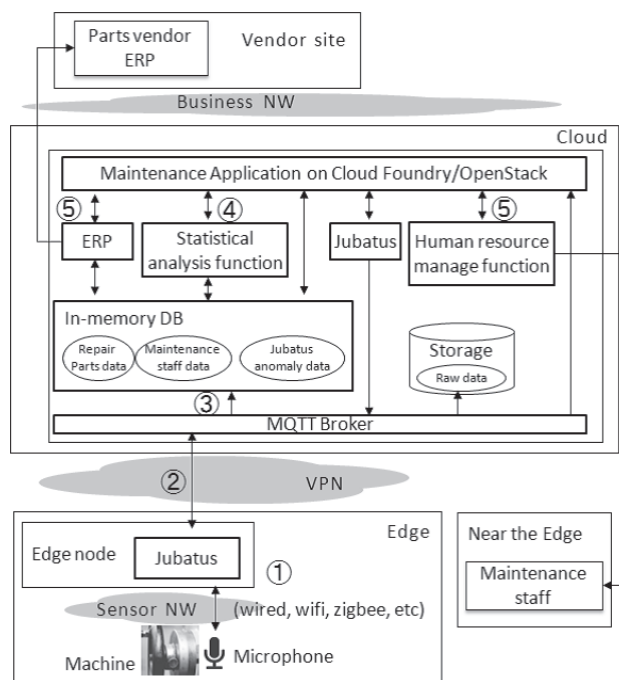

Fig. 1   Proposed maintenance platform architecture.

Cloud maintenance applications can be deployed on Cloud Foundry/OpenStack. Here, OpenStack is open source IaaS software and Cloud Foundry is open source PaaS software, and we can use them for cloud application development and operation. We previously contributed to the development of OpenStack and many cloud providers such as NTT Communications adopt OpenStack and Cloud Foundry for their cloud services [20], [21], [22].

In a cloud data center, an in-memory DB is also deployed on a baremetal server for backend system coordination. In-memory DB is uniformly referred both from an information system and a backend system. The statistical analysis function refers in-memory DB and evaluates the predicted failure impact. ERP also refers in-memory DB and orders parts for vendors. The human resource management function also refers in-memory DB and assigns maintenance staffs.

In this architecture, our original function is only a maintenance application on cloud. Its main tasks are to evaluate failure impacts based on Jubatus anomaly data using statistical analysis software and manage parts and staff if maintenance is needed. Except for hardware (Baremetal servers, Gateway), other functions can be implemented using Open Source Software. E.g., ADempiere [23] for ERP, PSPP [24] for statistical analysis, Jubatus, Moodle for human resource management, VoltDB for in memory DB, Mosquitto for MQTT Broker. To deploy these functions, we can use existing cloud deployment and verification technologies such as Refs. [25], [26], [27].

3.3 Processing Steps of Proposed Platform

Using Fig. 1, we explain the processing steps.

1. A microphone is attached to a business machine and sound data is continuously collected in an edge such as a factory. Jubatus in an edge node analyzes spectrum data of sound by anomaly detection or classification algorithm. When sound data is differ-

ent from the usual operation, Jubatus detects an anomaly and then only the anomaly data with raw data which Jubatus judge is sent to a cloud.

2. When data is sent to a cloud from an edge, MQTT which is a light weight publish-subscribe message queue protocol for sensor is used over secure VPN.

3. Suspicious anomaly data is inserted to in-memory DB.

4. A maintenance application predicts a failure such as which part has failed, and how much failure probability is within a certain period to analyze anomaly data. To predict failures, a maintenance application uses a statistical analysis function such as PSPP. And if we use deep learning, we may use GPU power [28].

5. Based on predicted failures, ordering data of maintenance is inserted to in-memory DB and orders are requested to ERP for repair parts arrangement and human resource management functions for maintenance staff assignments.

In parallel with these normal maintenance operations, a data scientist may analyze anomaly raw data stored in storage, update a machine learning model using Jubatus in a cloud and distribute the model to edges periodically. This can improve stream analysis in edges.

4. Implemented Example Application

Based on the proposed platform, we implemented an example application. This application shows a simple use case of maintenance which detects fan failures by analyzing the sound data of microphones attached to factory machines.

Figure 2 (a) right shows an edge in a factory. The edge node is Raspberry Pi and Jubatus 0.8.1 is installed on this. The sound of a machine fan is collected by a microphone, converted to 100 dimensional spectrum data by Fast Fourier Transformation calculated on Raspberry Pi. Jubatus analyzes stream frequency data sequentially.

When, we insert a foreign object to a fan in Fig. 2 (a), Jubatus detects the anomaly of a strange sound. Raspberry Pi sends anomaly data including judge results to a cloud. A cloud maintenance application analyzes the anomaly data in detail, predicts a failure and shows an alert with a dark grey mark in Fig. 2 (a) left. When a factory operator clicks the alert mark, failure parts and causes are shown and maintenance orders of repair parts and maintenance staffs are proposed. If the operator approves it, orders are requested to ERP and the human resource management function (Fig. 2 (b)). These steps are processed in real time (within few seconds) via in-memory DB.

To confirm the Jubatus effect to reduce the network cost, Fig. 2 (c) shows the network bandwidth usage. Figure 2 (c) left graph shows unfiltered data with light grey and anomaly data detected by Jubatus with dark grey. Figure 2 (c) right image shows the insertion of a foreign object in a fan. When a foreign object is inserted in a fan, Jubatus detects an anomaly and sends the data to a cloud. Through Fig. 2 (c) left graph, we can see Jubatus only sends the anomaly data with dark grey and Jubatus enables the network bandwidth usage to be reduced.

5. Conclusion

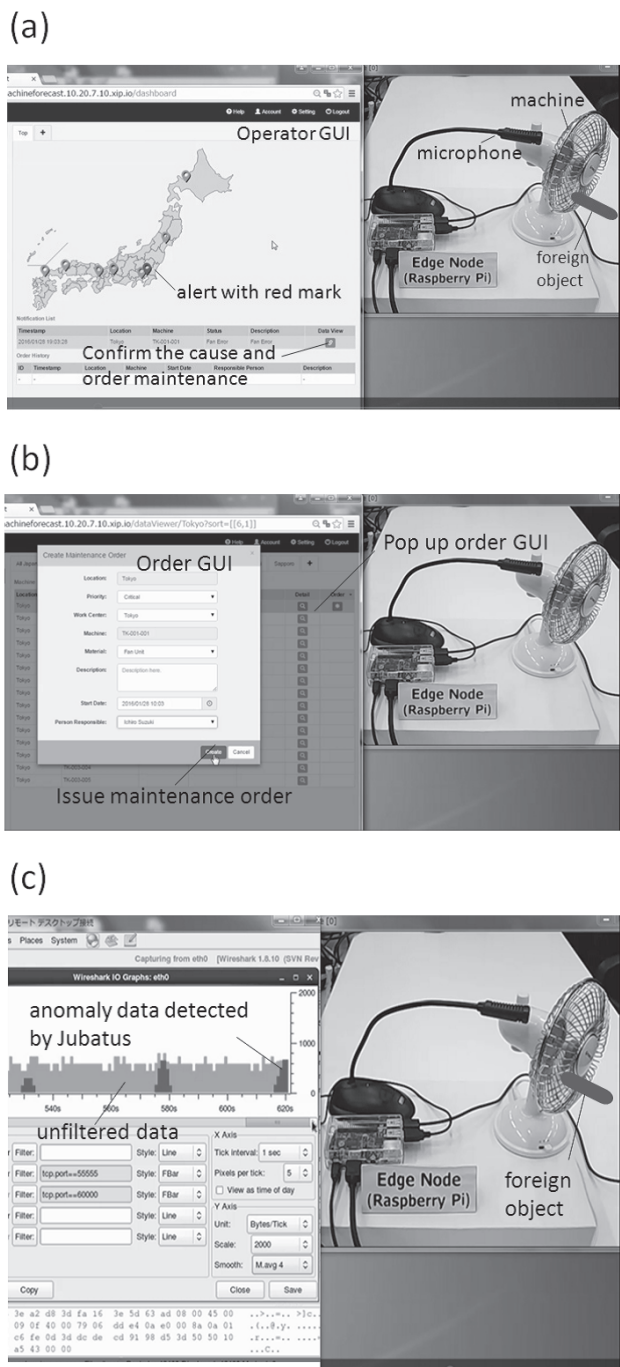This paper proposed a platform to accelerate the automatic

Fig. 2   Example application. (a) failure detection and shows an alert. (b) maintenance order to ERP. (c) comparing network bandwidth usage of unfiltered data and anomaly data detected by Jubatus.

maintenance of business machines and implemented a sample application. In our platform, Jubatus analyzes sound stream data in edges, only anomaly data are sent to a cloud, a cloud application predicts failures and orders maintenances such as repair parts preparations automatically via in-memory DB. In the future, we will propose our platform with the sample application to actual factories, and verify and improve the analysis precision ratio for actual business machines.

References

[1] Yamato, Y., Nishizawa, Y., Muroi, M. and Tanaka, K.: Development of Resource Management Server for Production IaaS Services Based on OpenStack, *Journal of Information Processing*, Vol.23, No.1, pp.58–66 (2015).

[2] Yamato, Y., Nishizawa, Y., Nagao, S. and Sato, K.: Fast and Reliable Restoration Method of Virtual Resources on OpenStack, *IEEE Trans. Cloud Computing*, DOI: 10.1109/TCC.2015.2481392, 12 pages (Sep. 2015).

[3] Yamato, Y.: Cloud Storage Application Area of HDD-SSD Hybrid Storage, Distributed Storage and HDD Storage, *IEEJ Trans. Electrical and Electronic Engineering*, Vol.11, No.5, pp.674–675 (2016).

[4] NTT Docomo press release website, available from ⟨https://www.nttdocomo.co.jp/english/info/media_center/pr/2015/0708_00.html⟩.

[5] Sunaga, H., Yamato, Y., Ohnishi, H., Kaneko, M., Iio, M. and Hirano, M.: Service Delivery Platform Architecture for the Next-Generation Network, *ICIN2008*, Session 9-A, Bordeaux (Oct. 2008).

[6] Yokohata, Y., Yamato, Y., Takemoto, M. Tanaka, E. and Nishiki, K.: Context-Aware Content-Provision Service for Shopping Malls Based on Ubiquitous Service-Oriented Network Framework and Authentication and Access Control Agent Framework, *IEEE Consumer Communications and Networking Conference* (*CCNC 2006*), pp.1330–1331, Las Vegas (Jan. 2006).

[7] Yamato, Y., Fukumoto, Y. and Kumazaki, H.: Proposal of Shoplifting Prevention Service Using Image Analysis and ERP Check, *IEEJ Trans. Electrical and Electronic Engineering*, Vol.12 (2017).

[8] Yamato, Y., Fukumoto, Y. and Kumazaki, H.: Analyzing Machine Noise for Real Time Maintenance, *2016 International Workshop on Signal Acquisition and Processing* (*ICSAP 2016*), Tokyo (Oct. 2016).

[9] AWS IoT Platform website, available from ⟨https://aws.amazon.com/iot/how-it-works/?nc1=h_ls⟩.

[10] Virtual Engineering Community website, available from ⟨https://www.vec-community.com/en/⟩.

[11] Google Project Brillo website, available from ⟨https://developers.google.com/brillo/⟩.

[12] Microsoft Azure IoT service document, available from ⟨http://download.microsoft.com/download/F/B/0/FB093814-BECE-4B54-A898-F7B66CCAF8AF/Get_started_with_the_Internet_of_Things.pdf⟩.

[13] Sakura IoT Platfomr website, available from ⟨https://iot.sakura.ad.jp/⟩.

[14] Hido, S., Tokui, S. and Oda, S.: Jubatus: An Open Source Platform for Distributed Online Machine Learning, *NIPS 2013 Workshop on Big Learning* (2013).

[15] Raspberry Pi website, available from ⟨https://www.raspberrypi.org/⟩.

[16] Stonebraker, M. and Weisberg, A.: The VoltDB main memory DBMS, *IEEE Data Engineering Bulletin*, Vol.36, No.2, pp.21–27 (2013).

[17] Yokohata, Y., Yamato, Y., Takemoto, M. and Sunaga, H.: Service Composition Architecture for Programmability and Flexibility in Ubiquitous Communication Networks, *IEEE International Symposium on Applications and the Internet Workshops* (*SAINTW'06*), pp.145–148, Phoenix (Jan. 2006).

[18] Yamato, Y., Nakano, Y. and Sunaga, H.: Study and Evaluation of Context-Aware Service Composition and Change-Over Using BPEL Engine and Semantic Web Techniques, *IEEE CCNC 2008*, pp.863–867 (Jan. 2008).

[19] Yamato, Y., Ohnishi, H. and Sunaga, H.: Development of Service Control Server for Web-Telecom Coordination Service, *IEEE International Conference on Web Services* (*ICWS 2008*), pp.600–607, Beijing (Sep. 2008).

[20] Yamato, Y., Shigematsu, N. and Miura, N.: Evaluation of Agile Software Development Method for Carrier Cloud Service Platform Development, *IEICE Trans. Inf. Syst.*, Vol.E97-D, No.11, pp.2959–2962 (2014).

[21] Yamato, Y., Katsuragi, S., Nagao, S. and Miura, N.: Software Maintenance Evaluation of Agile Software Development Method Based on OpenStack, *IEICE Trans. Inf. Syst.*, Vol.E98-D, No.7, pp.1377–1380 (2015).

[22] Yamato, Y.: OpenStack Hypervisor, Container and Baremetal Servers Performance Comparison, *IEICE Communication Express*, Vol.4, No.7, pp.228–232 (2015).

[23] ADempiere website, available from ⟨https://sourceforge.net/projects/adempiere/⟩.

[24] PSPP website, available from ⟨https://www.gnu.org/software/pspp/⟩.

[25] Yamato, Y., Muroi, M. Tanaka, K. and Uchimura, M.: Development of Template Management Technology for Easy Deployment of Virtual Resources on OpenStack, *Journal of Cloud Computing*, Springer, 2014, 3:7, DOI: 10.1186/s13677-014-0007-3, 12 pages (June 2014).

[26] Yamato, Y.: Automatic verification technology of software patches for user virtual environments on IaaS cloud, *Journal of Cloud Computing*, Springer, 2015, 4:4, DOI: 10.1186/s13677-015-0028-6, 14 pages (Feb. 2015).

[27] Yamato, Y.: Automatic system test technology of user virtual machine software patch on IaaS cloud, *IEEJ Trans. Electrical and Electronic Engineering*, Vol.10, pp.165–167 (2015).

[28] Yamato, Y.: Optimum Application Deployment Technology for Heterogeneous IaaS Cloud, *Journal of Information Processing*, Vol.25, No.1, pp.56–58 (2017).

Yoji Yamato received his B.S., M.S. degrees in physics and Ph.D. degree in general systems studies from University of Tokyo, Japan in 2000, 2002 and 2009, respectively. He joined NTT Corporation, Japan in 2002. There, he has been engaged in developmental research of Cloud computing platform, Peer-to-Peer computing and Service Delivery Platform. Currently he is a senior research engineer of NTT Software Innovation Center. Dr. Yamato is a senior member of IEEE and IEICE.

Yoshifumi Fukumto joined NTT Corporation, Japan in 2009. There, he has been engaged in developmental research of storage and IoT. Currently he is an engineer of NTT West Corporation, Japan.

Hiroki Kumazaki joined NTT Corporation, Japan in 2012. There, he has been engaged in developmental research of distributed system and machine learning. Currently he is an engineer of NTT Software Innovation Center, Japan.