Development of AI Technology for Machine Automation Controller (2)

The Insight Gained Through Implementation of Anomaly Detection AIs to the Machine Controller

ABE Yasuaki, UEYAMA Yuhki, SAKATANI Nobuyuki and FUJII Takashi

Recently, activities of productivity improvement have been attempted at production sites with predictive approaches. Cloud-based machine monitoring systems were once introduced but were not prevailing due to cost and others. As a countermeasure, it has been proposed to install an anomaly monitoring AI in the machine controller so as to introduce the feature to the production site at low cost. However, it is difficult to select an AI suitable for controller in implementation from various ones. Furthermore, it is difficult to implement the AI so as not to affect the machine control.

Therefore, the authors achieved to develop "Anomaly Detection AI equipped Machine Controller Prototype". This controller makes it possible to coexist with machine control process and AI process without interfering each other. In order to achieve that, they selected candidate AI programs from the anomaly detection AI programs and utilized task priority management and task scheduling functions of the machine controller.

1. Introduction

In recent years, AI (Artificial Intelligence) technology has been significantly developed. With the evolution of computers, it became possible to process large amounts of data which could not be handled thus far, and numerous methods for extracting the meanings inherent in data are proposed.

Along with the evolution of AI technology, the necessity of IT systems which can handle large amounts of data has been increasing. Against this background, an approach called edge computing is being designed[1]. Edge computing is based on the concept of reducing the processing load in the upper layer such as the cloud by equipping the lower layer of a system (edge) which is closed to a data source device such as a sensor with a data processing function to perform decentralized processing. The application of this approach to real-time systems such as automatic operation has been promoted; this approach does not simply reduce the load applied to computers and networks in the upper layer but also is accepted as a concept which is necessary for systems requiring high-speed responsiveness.

The importance of edge computing has also been recognized in the factory automation (FA) field, while the fusion of IT with OT (Operational Technology) is aspired to[2].

Contact : *ABE Yasuaki* yasuaki.abe@omron.com

The advantages of edge computing in the FA field include the reduction of communication load, the improvement of security, high-speed responsiveness, etc. In particular, high-speed responsiveness is a performance which is essential to apply to production devices controlled on a millisecond time scale.

Attempts to combine edge computing with AI to actually utilize these technologies for the sophistication of production have been launched. For example, efforts are being made to collect and analyze the data obtained from a microphone mounted on a production device with an edge terminal for the early detection of device anomalies using AI technology[3]. However, although large amounts of data exist at production sites, there are only a few cases where AI is fully utilized. Therefore, various companies are competing to develop such technology.

2. Development policy

2.1 Utilization of AI at production sites

In recent years, at production sites, the skills for the early detection of signs of device anomalies and product failures, which have relied on the experiences and hunches of experts so far, are being lost owing to a lack of experts. As a result, a reduction in productivity owing to device failure and the manufacture of defective products is becoming a serious problem.

Under such circumstances, efforts are being made to enable the early detection of and response to device anomalies and defective products by constantly monitoring and analyzing the

data collected from numerous sensors mounted on a device by using AI, as well as to lead to lossless development and design by analyzing the causes of anomalies with human knowledge[4].

2.2 Present issues and technological purpose of this subject

AI is generally found on servers with high processing capacity such as the cloud. However, considering the detection of device anomalies and feedback to control, there are limits to executing all AI processing on the server in terms of costs, such as sensor installation cost and communication cost, as well as in terms of technology, such as responsiveness and security. Therefore, to avoid such limits, adding AI to machine controllers is under consideration, and OMRON is also developing an "AI machine automation controller.[5]"

As one of the advantages of adding AI to a machine controller, it is sometimes possible to exhaustively obtain the latest data on device control because data communication is performed constantly with numerous sensors and actuators used for device control. Incorporating AI processing into this process realizes a control design which grasps the latest device condition accurately to instantaneously give feedback to the device depending on its condition.

However, AI has various applications, and there are a great number of algorithms for similar uses. Therefore, it is difficult to select AI which is easy to be adopted at production sites, as well as appropriate to be added to a machine controller from among them. In addition, it is necessary to implement the AI so as to avoid any negative influence on device control. For these reasons, it was very difficult to realize a machine controller with AI.

Therefore, the authors set the detection of device anomalies as a target application and developed an "anomaly detection machine controller prototype model" to attempt to solve the aforementioned issues.

2.3 Development policy

In this paper, we introduce the details on the development efforts where "anomaly detection AI," the assumed applications of which include the detection of device anomalies and which is appropriate to be added to a controller, was selected.

In the development of the anomaly detection machine controller prototype model, we select the anomaly detection AI which is most appropriate to be added to a machine controller by narrowing it down from among typical anomaly detection AIs based on the requirements which should be satisfied when introducing it to a production site, and then implement it on a machine controller to verify its execution performance.

In addition, we pursue a policy of ensuring that control processing takes priority and control processing is not inhibited by AI processing, so that control processing and AI processing can coexist on a machine controller.

3. Development of the anomaly detection machine controller prototype model

3.1 Development outline

In this development effort, we developed an anomaly detection machine controller prototype model by adding anomaly detection AI to OMRON's machine controller, assuming an application where a device anomaly occurred during production is detected instantaneously to prevent failures and defective products from being generated.

OMRON's machine controllers execute the control program at a fixed time interval called the control period. The shortest control period realized is 125 μs, and major data communications with the sensors and actuators are executed in synchronization with this period. This prototype model realized the mechanism of monitoring such control data and providing feedback on the anomaly detection results to the control program on OMRON's machine controller.

Furthermore, we mainly selected the anomaly detection AI, as well as designed and developed the coexistence of control and AI on the machine controller.

In the selection of the anomaly detection AI, we first listed typical anomaly detection AIs and then narrowed them down based on the requirements which should be satisfied when introducing the AI to a production site. We listed two types of anomaly detection AIs known as "outlier detection types" as candidates for the anomaly detection AI to be added to the machine controller. After that, we implemented such AIs on an actual machine controller and measured their execution performance such as processing time to select an algorithm called ISF (Isolation Forest).

Although the design of the coexistence of control and AI is not described in detail in this paper, we enabled control and AI to coexist by utilizing the strict task priority management function and task scheduling function equipped on OMRON's machine controller, so that the control period is strictly observed even during AI processing and the control processing always takes priority.

3.2 Anomaly detection AI

The anomaly detection function refers to a function to detect a behavior which is different from a normal expected behavior, based on normal behaviors. Table 1 shows the categories of AIs (anomaly detection AI) which realize such anomaly detection function and their typical algorithms.

Table 1 The categories of anomaly detection AIs

Category	Targeted anomaly	Typical algorithm
Outlier detection type	A value which deviates from major distribution trend	OneClassSVM, k-NN, LOF, k-means, ISF
Change detection type	A change in behavior and condition	Statistical test (such as t-test), hidden Markov
Prediction model type	A value which significantly deviates from a predicted value based on the learned model	Major methods of supervised learning (such as linear regression model, naive Bayes, SVM and random forest)
System anomaly detection type	Breakup of system structure and mutual dependence relation	Test of the difference in correlation coefficient

3.3 Requirements for introduction to production sites

When introducing AI to a production site, the AI needs to be friendly to those who actually use it at the production site.
The authors consider that the AIs which are easy to use at production sites satisfy the three major conditions shown below:

(1) High in terms of speed and light in terms of weight
For stable production, device control needs to be executed without fail. In addition, to make device control work with AI processing, the high-speed performance needs to be high enough to execute processing at high speeds even when coexisting with control. Algorithms for which the memory usage strains the control program are not suitable.

(2) Only a small amount of learning data is required
At production sites, the amount of time available for start-up and maintenance is limited. Under such circumstances, it is assumed to be difficult to collect large amounts of anomaly data, because anomalies occur with low frequency. Therefore, algorithms which can be used even if only a small amount of anomaly data are available are suitable.

(3) High interpretability of determination results
Since production site personnel are responsible for quality assurance, they need to explain the cause if a product failure or a device anomaly leading to a failure occurs. Therefore, in introducing the anomaly detection function, such function that enables the easy understanding of the grounds as to "why it was determined as an anomaly" is preferred.

Based on the above perspective, from among relatively fast outlier detection type algorithms, we chose LOF (Local Outlier Factor), because it operates in unsupervised learning mode and does not contain algorithms with low precision of explanation such as Kernel functions, as well as ISF, which is specialized in terms of higher speed and lighter weight and is expected to have high applicability for being added to a controller.

3.4 LOF and ISF

Next, we would like to explain these algorithms.

· LOF

LOF is one of the indicators of the degree of divergence between the learning data point group obtained in advance and the data point to be monitored. The larger the LOF, the higher the degree of anomaly of the subject of monitoring, and the closer to 1 the value, the lower the degree of anomaly. Setting a threshold value to the LOF enables the determination of normal and abnormal conditions.

The LOF of the feature point u in the space of an arbitrary dimension is defined by the following equation:

$$lof_k(u) = \frac{1}{k} \sum_{u' \in N_k(u)} \frac{dist_k(u)}{dist_k(u')} \quad (1)$$

Where, $N_k(u)$ is k-nearest neighbor of u. Furthermore, $dist_k(u)$ is the value obtained by taking the average of the neighborhood effective range from u to u' "$l_k(u–u')$" for N_k, which is defined as follows:

$$dist_k = \frac{1}{k} \sum_{u' \in N_k(u)} l_k(u \to u') \quad (2)$$

$$l_k(u \to u') \equiv \begin{cases} \varepsilon_k(u') \\ d(u,u) \end{cases} \quad (3)$$

$$u' \in N_k(u) \Lambda u \in N_k(u') \quad (4)$$
$$(otherwise)$$

$\varepsilon_k(u)$ is the diameter of the smallest sphere centered at u including $N_k(u)$ entirely, and $d(u, u')$ is a distance function of Euclidean distance, etc.

For example, we would like to consider the LOF when k is 1. If learning data set Q is given, the LOF of a certain subject point of monitoring p is calculated according to the following procedures. Fig. 1 shows the conceptual diagram of the calculation.

1. Explore the nearest neighbor point $q \in Q$ of the subject point of monitoring p [Fig. 1 (a)]
2. Evaluate $l_k(p \to q)$ to calculate $dist_k(p)$ [Fig. 1 (b)]
3. Explore the nearest neighbor point $r \in Q \cup \{p\}$ of q [Fig. 1 (c)]
4. Evaluate $l_k(q \to r)$ to calculate $dist_k(q)$ [Fig. 1 (d)]
5. Calculate $lof_k(p)$ from $dist_k(p)$ and $dist_k(q)$

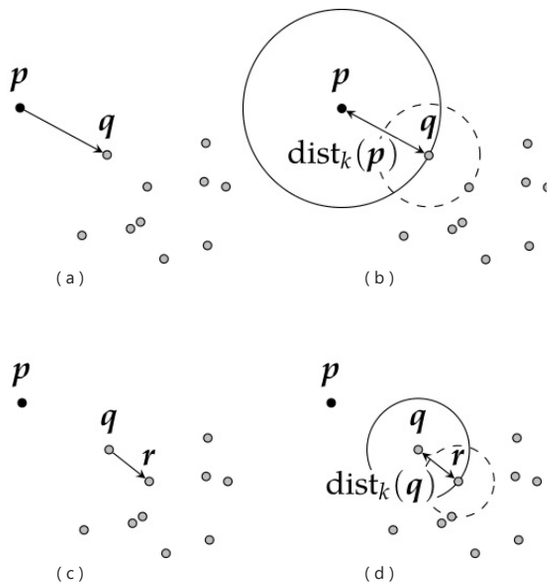

Fig. 1 Procedure of outlier detection

As we can see from the above, the neighborhood of the subject point of monitoring and the neighborhood of the subject point of comparison are considered in the calculation of LOF. Therefore, even in the case of varied data distribution density at which the k-nearest neighbor algorithm using only the neighborhood of the subject point of monitoring is not good, using LOF is expected to enable natural outlier detection.

· ISF

ISF is a method of creating a binary tree by recursively partitioning the learning data based on a hyperplane perpendicular to a coordinate axis which was determined randomly and then calculating the degree of anomaly based on the information of the binary tree node depth. Fig. 2 shows an example of data partitioning in the case of two dimensions. As this figure shows, the points which belong to a sparse region can be separated by a relatively small number of partitions, but separating the points which belong to a dense region requires a large number of partitions. In other words, it is highly possible that the data which appear frequently in the learning data are contained in the deep node of the binary tree and the data which appear rarely in the learning data are contained in the shallow node.

We would like to describe the detailed outlier detection procedures based on ISF. First, sub-sample the data of ψ point(s) from the learning data composed of n point(s) N_{tree} times. Next, create a binary tree for the respective sub-sampled data. The binary tree is created by partitioning the data using a random value for which the upper and lower limits are the maximum and minimum values of the randomly-selected axis. As Fig. 3 shows, partition the data recursively until the number of data contained in the node becomes 1 or less or the tree height becomes $log_2(\psi)$. On the assumption that the expected value of tree depth in t binary tree(s) for a certain data point x is $E[h(x)]$, the degree of anomaly $s(x,\psi)$ of x in the number of sampling times ψ can be defined by the following equation:

$$s(x,\psi) = 2^{-\frac{E[h(x)]}{c(\psi)}} \tag{5}$$

Where,

$$c(\psi) = H(\psi - 1) - \frac{2(\psi - 1)}{\psi} \tag{6}$$

$$H(i) = \log(i) + \gamma \tag{7}$$

γ is a Euler's constant ($\approx$ 0.57721). The degree of anomaly $s(x,\psi)$ ranges from 0 to 1, and setting a threshold to the degree of anomaly enables the determination of the outlier based on the learning data.

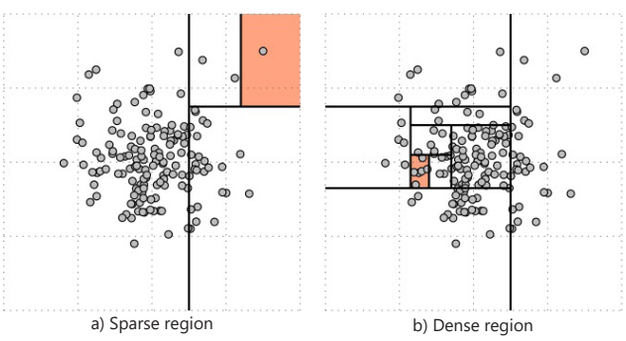

Fig. 2 Partitioning of data using a binary tree

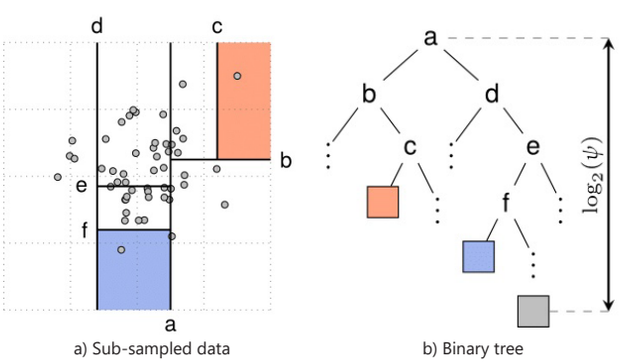

Fig. 3 Creation of a binary tree

3.5 Details of performance verification

To evaluate the advisability of adding to a machine controller, a performance verification concerning the anomaly detection processing time and memory usage is required. In this paper, we describe the performance verification of anomaly detection processing time.

Since control processing and anomaly detection processing coexist in one controller, the amount of time which can be allotted for anomaly detection processing is limited. Since these conditions vary depending on the user, it is too difficult to set uniform standards. However, it is preferred that processing can be performed at higher speeds to make it possible to respond to various applications. Therefore, in this development effort, we set the performance under control processing of several milliseconds as a rough indication.

Since OMRON's controllers are based on the task scheduling model as Fig. 4 shows, it is possible to estimate the anomaly detection performance independent of the contents of control processing if the anomaly detection processing time under arbitrary control task execution time can be measured.

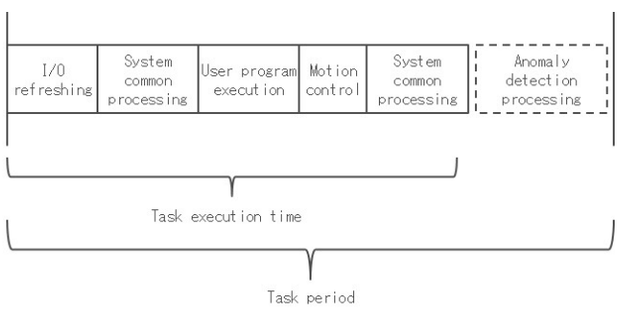

Fig. 4 task scheduling model

The anomaly detection processing time is considered to be mainly dependent on input/output to algorithms such as the number of learning data points, the number of learning data input/output dimensions and internal parameters, owing to the properties of the respective algorithms. Therefore, we conducted an experiment on the anomaly detection controller prototype model aimed at clarifying the relationship between the input/output conditions of anomaly detection algorithms and the processing time of the anomaly detection AI.

3.6 Performance verification environment

Table 2 shows the verification environment. The prototype model was developed based on an existing machine controller.

Table 2 Verification environment

Element	Targeted anomaly		Detail
Processing environment	Control period		1 ms
	Task execution time(The task execution time in one control period)		Approximately 100 μs
Anomaly detection algorithm	Method name		- LOF- ISF
	Learning data	Number of dimensions	6 / 8 / 10 / 12
		Number of points	100 / 1000/ 10000
	Parameter (LOF)	K	15
	Parameter (ISF)	Ntree	100 (Recommended value)
		Ψ	256 (Recommended value)

3.7 Performance verification results

Table 3 shows the maximum values of anomaly detection processing time. Although the processing time of ISF is at most 1 ms, it was found that the processing time of LOF increased significantly with an increase in the number of learning data points.

Table 3 Verification results (Maximum execution time)

Maximum execution time [ms]		Number of dimensions			
Algorithm name	Number of data points	6	8	10	12
ISF	100	0.270	0.229	0.254	0.241
	1000	0.391	0.360	0.376	0.373
	10000	0.322	0.341	0.331	0.360
LOF	100	4.00	4.56	5.47	5.19
	1000	36.5	41.8	44.8	56.1
	10000	336	441	509	644

Fig. 5 shows the results when the number of learning data points is 100. The reason why the processing time of ISF is so much shorter than that of LOF is considered to be because of the characteristics of their algorithms. The main cause is that the absolute number of processes is large, because LOF calculates the correlation between learning data and monitoring data each time monitoring data are input without creating a model for anomaly determination in advance, while ISF only requires the processing of tracing the tree structure when monitoring data are input because a model for anomaly determination is constructed in advance in tree structure at the time of learning. In addition, since ISF defines the upper tree depth limit as $log_2\psi$, the processing time will not increase even if the amount of learning data increases beyond a certain level.

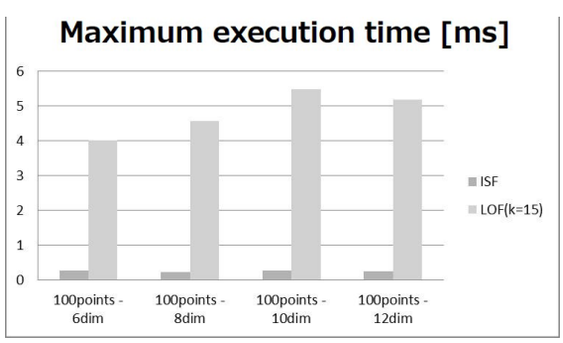

Fig. 5 Maximum execution time (Learning data: 100 points)

Based on the above results, ISF was found to be superior in anomaly detection algorithm execution performance. Since this verification was conducted under a condition with a relatively smaller load on control processing such that the proportion of task execution time was approximately 10%, the maximum execution time was at most approximately 0.4 ms. This means

that the processing is completed within 3.1 ms based on the task scheduling model shown in Fig. 4, even under high load, such as a task execution time of 900 μs. In addition, it is also possible to set more than one anomaly detection subject to perform high-speed anomaly detection. For example, on a production device in which many workpieces are input continuously, it is possible to conduct anomaly monitoring related to the production of the respective workpieces in a parallel manner.

3.8 Conclusion

In this chapter, we introduced the selection of the anomaly detection AI and the design which enabled the coexistence of control and AI on a machine controller, which we conducted in the development of an anomaly detection machine controller prototype model, and also described the process of selecting an anomaly detection AI in detail.

As requirements for introducing AI to production fields, we listed “High in terms of speed and light in terms of weight,” “Only a small amount of learning data is required” and “High level of interpretability,” and selected LOF and ISF as algorithms which met such requirements from among anomaly detection AIs. Furthermore, we also verified their execution performance on a machine controller and determined that ISF was an algorithm which was more suitable for being added to a machine controller than LOF.

In addition, we also introduced our efforts to prevent AI from influencing control by making sure to observe the control period even while AI processing is being executed and giving constant priority to control processing so that control and AI can coexist on a machine controller.

It can be said that the requirements for introduction to production sites we described here are matters which should be evaluated in common when considering adding not only anomaly detection, but also AI to a machine controller.

Furthermore, the execution performance verification method was targeted at only some algorithms, namely LOF and ISF. However, the same evaluation method can be utilized for algorithms focused on the distance between learning data and monitoring subject data like LOF, as well as those with a binary tree structure like ISF.

4. Summary

In this paper, we introduced an actual example of the development of an anomaly detection machine controller prototype model targeting the detection of device anomalies, and also described issues that need to be considered when adding AI to a machine controller, as well as the process of considering the advisability of adding AI to a machine controller.

In the future, since the utilization of not only anomaly detection, but also AI at production sites is expected to be promoted, we will proceed with a study on adding AIs which were not studied this time, based on the knowledge gained in this development effort.

In addition, we are planning to introduce an AI-added machine controller to an actual production site on a trial basis to improve the requirements for introducing AI to production sites.

Finally, we would like to express our sincere gratitude to those who were involved in the technological and production development of the AI machine automation controller for their extensive cooperation in this development effort.

References

1) Nippon Telegraph and Telephone Corporation. “Announcing the “Edge computing” concept and the “Edge accelerated Web platform” prototype to improve response time of cloud applications”. http://www.ntt.co.jp/news2014/1401/140123a.html, (accessed 2018-03-05).

2) Ministry of Economy, Trade and Industry. “Information Economy Subcommittee, Industrial Structure Council, WG on Distribution Strategy (1st)”. http://www.meti.go.jp/committee/sankoushin/shojo/johokeizai/bunsan_senryaku_wg/pdf/001_03_00.pdf, (accessed 2018-3-29)

3) NTT DATA Corporation. “Activity of edge computing in AI & IoT era”. http://www.nttdata.com/jp/ja/insights/blog/20170316.html, (accessed 2018-03-29)

4) Ministry of Economy, Trade and Industry. "The White Paper on Manufacturing Industries 2017, Part 1 Current State of Manufacturing Infrastructure Technology and Related Issues". http://www.meti.go.jp/report/whitepaper/mono/2017/honbun_pdf/pdf/honbun01_01_02.pdf, (accessed 2018-4-10)

5) OMRON Corporation. “OMRON Develops AI-Equipped Machine Automation Controller –Real-time integration of machine control and AI– ”. http://www.omron.co.jp/press/2017/04/c0425.html, (accessed 2018-03-05).

6) Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. “Lof: identifyingdensity-based local outliers”. In ACM sigmodrecord, ACM. 2000, Vol. 29, p. 93–104.

7) Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. “Isolation-based anomaly detection”. ACM Transactions on Knowledge Discovery from Data (TKDD), 2012, Vol. 6, No. 1, p. 3.

About the Authors

ABE Yasuaki
Embedded System Research Center
Technology And Intellectual Property H.Q.
Specialty: Computer Science

UEYAMA Yuhki
Embedded System Research Center
Technology And Intellectual Property H.Q.
Specialty: Computer Science

SAKATANI Nobuyuki
Embedded System Research Center
Technology And Intellectual Property H.Q.
Specialty: Computer Science

FUJII Takashi
Technology Department 1
Technology Development Division H.Q.
Industrial Automation Company
Specialty: Control Engineering
Affiliated Academic Society: IEEJ, SICE