

マシンコントローラに搭載可能な AI 技術の開発 (2)

異常検知コントローラプロトモデル開発における実証例について

阿部 泰明・上山 勇樹・阪谷 信幸・藤井 高史

近年、生産現場では装置や製品品質を常時監視し、それらの異常や不良を予兆の段階から検知・対処する取り組みが進められている。従来は、監視用にセンサを設置し、そこから得られたデータをクラウドなどで解析していたが、コスト面などの理由で導入は限定的だった。そこで、装置制御を管理・実行しているマシンコントローラに AI (Artificial Intelligence) を搭載し、異常監視機能を低コストで生産現場に導入することが提案されている。

しかし AI には様々な機能があり、また同様の機能であっても数々のアルゴリズムが存在しているため、その中から、生産現場にとって受け入れやすく且つマシンコントローラへの搭載に向いている AI を適切に選定するのは難しい。さらに、その AI を装置制御に悪影響を与えないように実装しなくてはならず、AI を搭載したマシンコントローラの実現はハードルが高かった。

そこで、筆者らは装置の異常検知をターゲットとして、生産現場で特に求められる要件を「高速・軽量であること」、「学習データの必要量が少ないこと」、「判定結果に対する納得性が高いこと」と設定することで、異常検知用 AI から候補となる AI を選定し、さらにマシンコントローラのタスク優先度管理及びスケジューリング機能を利用して装置制御と AI が共存可能な「異常検知マシンコントローラプロトモデル」を開発することに成功した。

Development of AI Technology for Machine Automation Controller (2)

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

Yasuaki Abe, Yuhki Ueyama, Nobuyuki Sakatani, Takashi Fujii

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. まえがき

近年、AI (Artificial Intelligence) 技術の発展が著しい。コンピュータの進化に伴い、これまでは扱えなかった多量のデータを処理できるようになり、データに内在する意味を抽出する手法が数多く提案されている。

AI 技術が発展する中、多量のデータを扱える IT システムの必要性が高まっており、その流れの中で、エッジコン

ピューティングというアプローチが構想されている¹⁾。エッジコンピューティングとは、センサなどのデータソースデバイスに近い、システムの下位レイヤ (エッジ) ヘデータ処理機能を持たせ、分散処理させることで、クラウドなど上位レイヤでの処理負荷を軽減させるという考え方である。これは、単に上位レイヤのコンピュータやネットワークにかかる負荷を軽減させるだけでなく、高速な応答性が要求されるようなシステムにも必要な考え方として、自動運転

などのリアルタイムシステムにも適用が進められている。

ファクトリーオートメーション (FA) の分野でも、IT と OT (Operational Technology) の融合が志向される中、エッジコンピューティングの重要性が認識されてきている²⁾。

FA 分野でのエッジコンピューティングのメリットには通信負荷の軽減やセキュリティの向上、高速な応答性などが挙げられる。中でも高速な応答性は、ミリ秒単位で制御されている生産装置に活用するには不可欠な性能である。

エッジコンピューティングと AI を組み合わせて、実際に生産の高度化に活用しようという試みも始まっている。例えば、生産装置に取付けたマイクから取得したデータを、エッジ端末で収集・解析し、AI 技術によって装置の異常を早期に検知する取組みが行われている³⁾。ただし、多量のデータが存在する製造現場ではあるが、AI を十分に活用できた事例はまだ少なく、各社が競って技術開発を行っている状況である。

2. 開発方針

2.1 生産現場での AI 活用 近年、生産現場では、熟練者不足の影響で、これまで熟練者の経験や勘に頼ってきた装置異常や製品不良の予兆を早期に捉える技能が失われ、結果として、装置の故障や不良品の製造による生産性低下が深刻化している。

その中で、装置異常や製品不良を、装置に設置した多数のセンサから収集したデータを、AI 活用により常時監視・解析することで、早期の異常発見・対処を可能にし、更に異常の原因を人の知見により分析することで、ロスの無い開発・設計につなげようという取り組みなどが進められている⁴⁾。

2.2 現状の問題点と本テーマの技術的目的 一般的に、AI の搭載先は、処理能力の高いクラウドなどのサーバであることが多いが、装置異常の検知や制御へのフィードバックを考えた場合、全ての AI 処理をサーバ上で実行するのは、センサ設置費用や通信費などのコスト面や、応答性やセキュリティなどの技術面で限界がある。そこで、このような限界を回避するために、マシンコントローラに AI を搭載することが考えられており、オムロンでも「AI マシンオートメーションコントローラ」を開発している⁵⁾。

マシンコントローラに AI を搭載することのメリットとして、装置制御に使用する多くのセンサやアクチュエータと常時データ通信をしているため、装置制御に関する最新のデータを網羅的に取得できることがある。そこに AI 処理を組み込むことで、装置の最新の状態を的確に把握し、装置の状態に応じて即時に装置へフィードバックを行うような制御設計が可能になる。

しかし、AI には様々な用途があり、同様の用途であっても数々のアルゴリズムが存在しているため、それらの中から、生産現場で受け入れやすく且つマシンコントローラ

への搭載に適した AI を選定することは難しい。しかも、その AI を装置制御に悪影響を与えないように実装しなくてはならない。そのため、AI を搭載したマシンコントローラの実現はハードルが高かった。

そこで、筆者らは装置の異常検知をターゲットアプリケーションと設定し、「異常検知マシンコントローラプロトモデル」の開発を行い、その中で上記の課題を解決していくことを試みた。

2.3 開発方針 本論文では、装置の異常を検知するなどの用途を想定した「異常検知 AI」を対象に、コントローラ搭載に適したものを選定した際の実現内容を紹介する。

異常検知マシンコントローラプロトモデル開発において、代表的な異常検知 AI の中から、まず生産現場に導入する際に満たすべき要件により絞り込み、更にそれらをマシンコントローラ上に実装し実行性能を検証することで、マシンコントローラに搭載するのに最適な異常検知 AI を選定する。

また、制御処理と AI 処理をマシンコントローラ上で共存させるために、制御処理が確実に優先されるようにし、制御処理が AI 処理に阻害されないようにする方針とする。

3. 異常検知マシンコントローラプロトモデル開発

3.1 開発概要 本開発では、生産中に発生する装置の異常を瞬時に捉え、故障や不良品の発生を未然に防ぐような用途を想定し、オムロンのマシンコントローラに、異常検知 AI を搭載した異常検知マシンコントローラプロトモデルを開発した。

オムロンのマシンコントローラは、制御プログラムを制御周期と呼ぶ一定の時間間隔で実行する。制御周期は、最短 125 μ s が実現されており、センサ・アクチュエータとの主なデータ通信も、この周期に同期して実行される。このプロトモデルでは、このような制御データを監視し、異常検知結果を制御プログラムにフィードバックする仕組みをオムロンのマシンコントローラ上で実現した。

本開発では、主に異常検知 AI の選定と、マシンコントローラ上で制御と AI が共存するための設計・開発を行った。

異常検知 AI の選定においては、まず異常検知 AI の代表的なものを挙げ、それらを生産現場に導入する際に満たすべき要件により絞り込み、外れ値検知型と呼ばれる 2 種類の異常検知 AI をマシンコントローラ搭載の候補に挙げた。更にそれらを実際にマシンコントローラに実装し、処理時間などの実行性能を測定することで、ISF (Isolation Forest) というアルゴリズムを選定した。

制御と AI の共存設計に関しては本論文では詳細については触れないが、オムロンのマシンコントローラが備える厳格なタスク優先度管理機能とタスクスケジューリング機能を利用し、AI 処理実行時でも制御周期を確実に守るようにし、常に制御処理が優先されるようにすることで、制御と AI が共存できるようにした。

3.2 異常検知AI 異常検知機能とは、期待される正常な挙動とは異なる挙動を、正常な挙動から判別する機能を指す。このような異常検知機能を実現するAI（異常検知AI）を分類し、代表的なアルゴリズムを表1に示す。

表1 異常検知AIの分類

分類	ターゲットとなる異常	代表的アルゴリズム
外れ値検知型	分布の大勢から外れた値	OneClassSVM, k-NN, LOF, k-means, ISF
変化検知型	振る舞い・状態の変化	統計的検定 (t検定など), 隠れマルコフ
予測モデル型	学習したモデルによる予測値からの誤差が大きな値	教師あり学習の主な手法 (線形回帰モデル, ナイーブベイズ, SVM, ランダムフォレストなど)
系の異常検知型	系の構造・相互依存関係の崩れ	相関係数の差の検定

3.3 生産現場への導入要件 生産現場にAIを導入するということは、そのAIは生産現場で実際に使用する人にとって、使いやすいものである必要がある。

筆者らは、生産現場で使用しやすいAIとは、主に下記の3つの条件を満たすものであると考える。

(1) 高速・軽量であること

生産を安定して行うには装置制御を確実に実行する必要がある。その上で装置制御とAI処理とを協調させるには、制御と共存しても十分高速に実行できる程度の高速性が必要になる。制御プログラムを圧迫するようなメモリ使用量のアルゴリズムは適していない。

(2) 学習データの必要量が少ないこと

生産現場では立ち上げやメンテナンスにかけられる時間が限られている。その中で発生頻度の低い異常時のデータを多量に収集することは困難であると想定される。そのため、異常データが少ない場合でも使用可能なアルゴリズムが適している。

(3) 判定結果に対する納得性が高いこと

生産現場は、品質保証に対する責任を負っているため、製品不良やそれにつながる装置異常が発生した場合、その原因を説明する必要がある。そのため、異常検知機能を導入するにあたって、「なぜそれを異常と判断したのか」という根拠が理解しやすいものが好まれる。

上記の観点を踏まえ、比較的高速なアルゴリズムである外れ値検知型で、教師なし学習で動作しカーネル関数など説明性の低いアルゴリズムを含まない点で、LOF (Local Outlier Factor) と、より高速・軽量に特化しておりコントローラ搭載の適性が高いと見込まれる ISF を選定した。

3.4 LOF・ISF 次に、これらのアルゴリズムについて説明する。

・LOF

LOF は、あらかじめ取得された学習データ点群と監視

対象データ点の乖離度を示す指標の一つである。LOF が大きいほど監視対象の異常度合が高く、値が1に近いほど異常度合が低いことを意味する。このLOF に対して閾値を設定することにより、正常・異常の判別が可能となる。

任意次元の空間における特徴点 u のLOFは次式で定義される。

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

ここで、 $N_k(u)$ は u の k 近傍である。また、 $dist_k(u)$ は、 u から u' への近傍有効距離 $l_k(u \rightarrow u')$ を N_k について平均をとったものであり、次のように定義される。

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

$$l_k(u \rightarrow u') \equiv \begin{cases} \epsilon_k(u') & u' \in N_k(u) \wedge u \in N_k(u') \\ d(u, u') & (otherwise) \end{cases}$$

なお、 $\epsilon_k(u)$ は $N_k(u)$ をすべて含む u を中心とした最小の球の半径、 $d(u, u')$ はユークリッド距離などの距離関数である。

例として、 $k=1$ とした場合のLOFについて考えてみる。学習データの集合 Q が与えられた場合、ある監視対象点 p のLOFは次の手順で算出される。算出の概念図を図1に示す。

1. 監視対象点 p の最近傍点 $q \in Q$ を探索する (図1 (a))
2. $l_k(p \rightarrow q)$ を求め、 $dist_k(p)$ を算出する (図1 (b))
3. q の最近傍点 $r \in Q \cup \{p\}$ を探索する (図1 (c))
4. $l_k(q \rightarrow r)$ を求め $dist_k(q)$ を算出する (図1 (d))
5. $dist_k(p)$ および $dist_k(q)$ から $lof_k(p)$ を算出する

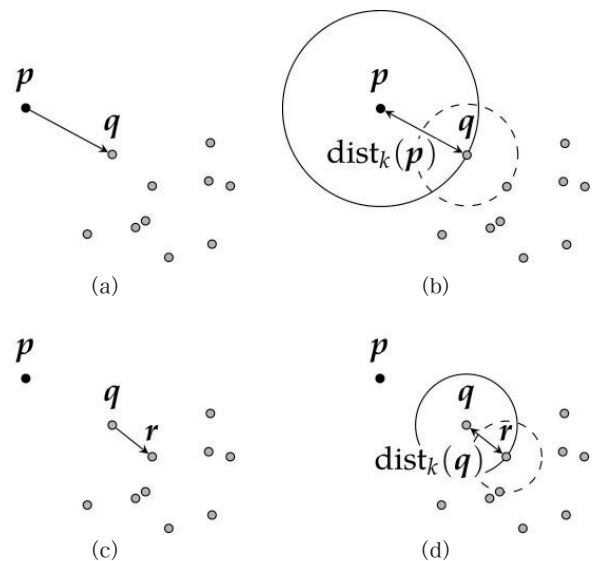


図1 訓練データの一部

以上からわかるように、LOF の算出においては監視対象点の近傍に加えて、比較対象となる点の近傍も考慮される。そのため、監視対象点の近傍のみを用いる k 近傍法が苦手とするデータ分布に疎密があるような場合でも、LOF

を用いることにより自然な外れ値検知が期待できる。

・ ISF

ISFは、ランダムに決定した座標軸に垂直な超平面によって学習データを再帰的に分割することで2分木を生成し、その2分木のノードの深さ情報をもとに異常度合を算出する手法である。2次元の場合のデータ分割の例を図2に示す。同図に示されるように、疎な領域に属する点は比較的少ない分割によって分離できるが、密な領域に属する点を分離するにはより多くの分割が必要となる。つまり、2分木の深いノードには学習データによく出現するデータが、浅いノードには学習データではまれなデータが含まれている可能性が高い。

ISFによる外れ値検知の手順について詳述する。まず、 n 点からなる学習データから ψ 点のデータを N_{tree} 回サブサンプリングする。次に、サブサンプリングした各データについて2分木を生成する。2分木は、ランダムに選んだ軸の最大値と最小値をそれぞれ上限と下限とするランダムな値によってデータを分割することで生成する、データの分割は図3に示すように、ノードに含まれるデータが1点以下となるか木の高さが $\log_2(\psi)$ となるまで再帰的に行う。あるデータ点 x についての t 個の2分木における木の深さの期待値を $E[h(x)]$ とすると、サンプリング数 ψ における x の異常度合 $s(x, \psi)$ は次式で定義される：

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

ここで、

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

$$H(i) = \log(i) + \gamma$$

であり、 γ はオイラー定数 (≈ 0.57721) である。異常度合 $s(x, \psi)$ は $(0, 1]$ の範囲をとり、この異常度合に対して閾値を設定することによって、学習データを基準とした外れ値の判別が可能となる。

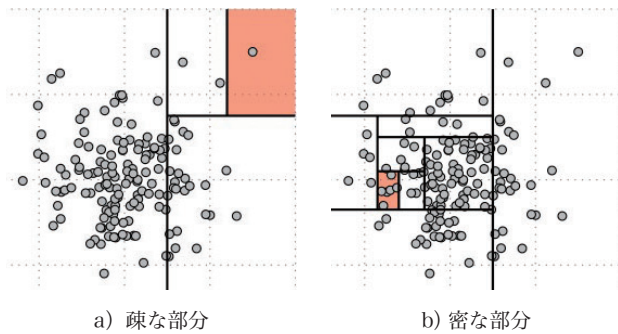


図2 2分木によるデータの分割

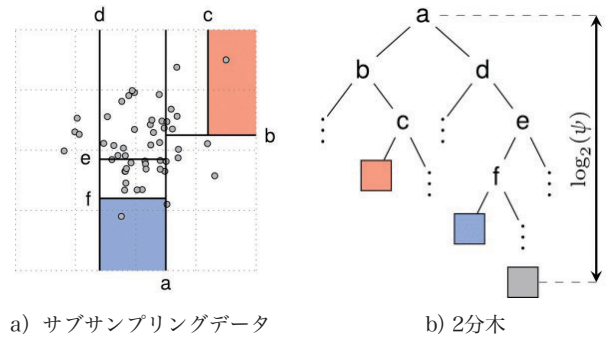


図3 2分木の生成

3.5 性能検証内容 マシンコントローラへの搭載可否の評価において、異常検知処理の処理時間とメモリ使用量に関する性能検証が必要である。本論文では異常検知処理時間の性能検証について述べる。

制御処理と異常検知処理が1つのコントローラに共存するため、異常検知処理に割り当て可能な時間は制限を受ける。ユーザごとにこれらの条件は異なるため、一意に基準を設けることはできないが、様々な用途に対応できる点で、より高速であることが望ましく、本開発では制御処理下での性能が数ミリ秒程度であることを目安とした。

オムロンのコントローラでは図4に示すようなタスクスケジューリングモデルを取っているため、任意の制御タスク実行時間下での異常検知処理時間を測定できれば、制御処理内容に依らず、異常検知性能を推定することができる。

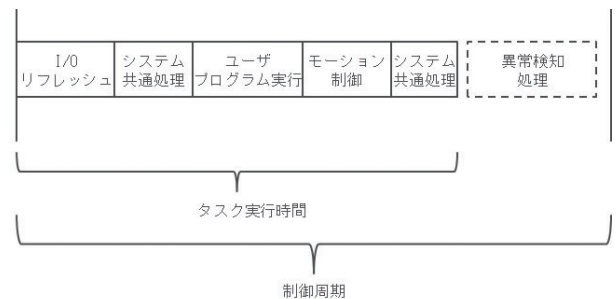


図4 タスクスケジューリングモデル

異常検知処理時間は、各アルゴリズムの性質から、主に学習データ点数・学習データ入出力次元数・内部パラメタといったアルゴリズムへの入出力に依存していると考えられる。そこで、異常検知アルゴリズムの入出力条件と異常検知AIの処理時間との関係を明らかにすることを目指し、異常検知コントローラプロトモデル上で実験を行った。

3.6 性能検証環境 検証環境を表2に示す。プロトモデルは、既存のマシンコントローラをベースに開発した。

表2 検証環境

要素	ターゲットとなる異常	詳細
処理環境	制御周期	1 ms
	タスク実行時間 (1制御周期に占めるタスク実行時間)	約100 μ s

異常検知 アルゴリズム	手法名		-LOF -ISF
	学習 データ	次元数	6 / 8 / 10 / 12
		点数	100 / 1000 / 10000
	パラメタ (LOF)	k	15
	パラメタ (ISF)	Ntree	100 (推奨値)
ψ		256 (推奨値)	

3.7 性能検証結果 異常検知処理時間の最大値を表3に示す。ISFの処理時間は最大でも1 msに満たないが、LOFは学習データ点数が増加すると、極端に大きな結果となることが分かった。

表3 検証結果(最大実行時間)

最大実行時間 [ms]		次元数			
アルゴ名	データ点数	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

学習データ点数が100点の時の結果を、図5に示す。ISFの処理時間がLOFよりも極端に短いのが、これは、それぞれのアルゴリズムの特性に依るものと考えられる。ISFが学習時に予め異常判定用モデルを木構造で構築するため、監視データ入力時には木構造をたどる処理で済むのに対し、LOFは事前に異常判定用モデルを作ることはせず、監視データの入力時に都度学習データと監視データとの相対関係を計算するため、絶対的な処理数が多いことが主な原因として考えられる。また、ISFは木の深さの上限を $\log_2\psi$ と定義しているため、学習データが一定以上増えても処理時間が増えなくなっている。

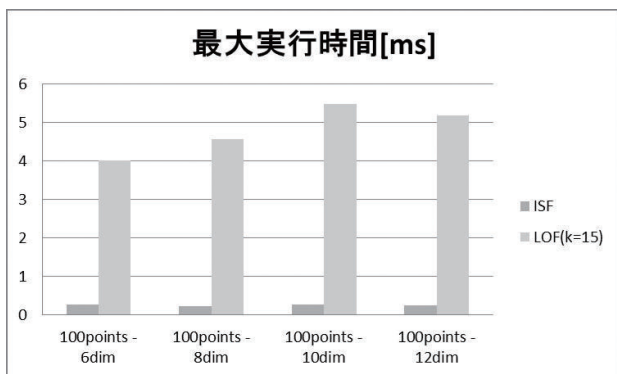


図5 最大実行時間(学習データ:100点)

以上の結果より、異常検知アルゴリズムの実行性能において、ISFが優れていることが分かった。本検証ではタスク実行時間割合が10%程度と制御処理の負荷が比較的小

さい条件であり、最大0.4 ms程度であった。これは、仮にタスク実行時間が900 μ sの高負荷下でも、図4で示したタスクスケジューリングモデルに基づく3.1 ms以内で処理が完了することになる。また、異常検知対象を1つだけでなく複数設定し、高速な異常検知を行うことも可能である。例えば、多数のワークが連続して投入されるような生産装置において、各ワークのそれぞれの生産に関する異常監視を並行して実施することも可能である。

3.8 結論 本章では、異常検知マシンコントローラプロトモデル開発において実施した、異常検知AIの選定と、マシンコントローラ上で制御とAIが共存するための設計について紹介し、異常検知AIの選定プロセスについて詳述した。

AIを生産現場に導入する際の要件として、「高速・軽量であること」「学習データの必要数が少ないこと」「説明性が高いこと」を挙げ、異常検知AIの中からその要件を満たすアルゴリズムとして、LOFとISFを選定した。更にそれらについてコントローラ上での実行性能を検証することで、LOFよりもISFの方がマシンコントローラへの搭載に適したアルゴリズムであることを導いた。

また、マシンコントローラ上で制御とAIを共存させるため、AI処理実行時でも制御周期を確実に守るようにし、常に制御処理が優先されるようにすることで、AIが制御に影響を与えないようにした点についても紹介した。

ここで挙げた生産現場への導入要件は、異常検知に限らずAIをマシンコントローラに搭載することを検討する上で、共通して評価すべき項目であると言える。

また、実行性能検証方法についても、LOFとISFという一部のアルゴリズムを対象にした検証ではあるが、LOFのように学習データと監視対象データとの距離に着目したアルゴリズムや、ISFのように2分木構造を有するアルゴリズムに対しては同様の評価方法が活用できる。

4. まとめ

本論文では、装置の異常検知をターゲットとした、異常検知マシンコントローラプロトモデルの開発の実例を紹介し、AIをマシンコントローラに搭載する際の課題と、搭載の適否を検討するプロセスについて述べた。

今後は、異常検知に限らず生産現場でのAIの活用が進んでいくと想定されるため、本開発で得た知見を基に、今回対象としなかったAIについても搭載の検討を進めていく。

また、生産現場へのAIの導入要件についても、実際の生産現場にAIを搭載したマシンコントローラを試験導入し、その取り組みを通じてブラッシュアップしていく予定である。

最後に、今回の開発にあたり、多大なご協力をいただいたAI搭載マシンオートメーションコントローラの技術開発及び商品開発に携わった方々に深く感謝申し上げる。

参考文献

- 1) 日本電信電話株式会社. “高レスポンスやビックデータ処理が要求される新たなアプリケーションの開拓を推進する「エッジコンピューティング構想」を策定”. <http://www.ntt.co.jp/news2014/1401/140123a.html>, (参照 2018-03-05)
- 2) 経済産業省.” 産業構造審議会情報経済小委員会 分散戦略WG (第 1 回) ”. http://www.meti.go.jp/committee/sankoushin/shojo/johokeizai/bunsan_senryaku_wg/pdf/001_03_00.pdf, (参照 2018-3-29)
- 3) 株式会社NTTデータ.” IoT & AI時代におけるエッジコンピューティングへの取り組み”. <http://www.nttdata.com/jp/ja/insights/blog/20170316.html>, (参照2018-03-29)
- 4) 経済産業省.” 2017年版ものづくり白書—第一部 ものづくり基盤技術の現状と課題”. http://www.meti.go.jp/report/whitepaper/mono/2017/honbun_pdf/pdf/honbun01_01_02.pdf, (参照 2018-4-10)
- 5) オムロン株式会社. “マシン制御とAIをリアルタイムに融合「AI搭載マシンオートメーションコントローラ」を開発”. <http://www.omron.co.jp/press/2017/04/c0425.html>, (参 照 2018-03-05) .
- 6) Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifyingdensity-based local outliers. In ACM sigmodrecord, Vol. 29, pp. 93–104. ACM,2000.
- 7) Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation-based anomaly detection. ACM Transactions on Knowledge Discovery from Data (TKDD) , Vol. 6, No. 1, p. 3, 2012.

執筆者紹介



阿部 泰明 Yasuaki Abe
技術・知財本部 組込システム研究開発センタ
専門：情報工学
所属学会：なし



上山 勇樹 Yuhki Ueyama
技術・知財本部 組込システム研究開発センタ
専門：情報工学
所属学会：なし



阪谷 信幸 Nobuyuki Sakatani
技術・知財本部 組込システム研究開発センタ
専門：情報工学
所属学会：なし



藤井 高史 Takashi Fujii
インダストリアルオートメーションビジネス
カンパニー 技術開発本部 第1技術部
専門：制御工学
所属学会：電気学会、計測自動制御学会