Rev. 2.1.0.0057（2020/07/05）

機械学習品質マネジメントガイドライン

第 2 版

（revision 2.1.0）

2021 年 7 月 5 日

国立研究開発法人産業技術総合研究所

デジタルアーキテクチャ研究センター

テクニカルレポート DigiARC-TR-2021-01

サイバーフィジカルセキュリティ研究センター

テクニカルレポート CPSEC-TR-2021001

人工知能研究センター

テクニカルレポート

前書き（Foreword and Disclaimer）

本ガイドラインは、国立研究開発法人新エネルギー・産業技術総合開発機構（NEDO）からの受託事業の一部として、国立研究開発法人産業技術総合研究所（産総研・AIST）と大学共同利用機関法人情報・システム研究機構国立情報学研究所（NII）が企業・大学等の有識者委員とともに構成した「機械学習品質マネジメント検討委員会」においてとりまとめたものである。

本ガイドラインの内容に寄与した委員の意見は技術者としての個人の知見に基づくものであり、各々が所属する会社等の意見を代表するものではない。

本ガイドラインは、機械学習人工知能を利用したシステム・サービスの開発を主導する企業等が、そのビジネス等への影響を踏まえて主体的にその採用の有無を選択し、共同開発者等とともに実践するものであって、法令・公的指針等との関係においては非拘束的なものである。本ガイドライン中で規範的（normative）とされる規定は、あくまで本ガイドラインを任意に採用した場合に限り、規範的な意味を持つものである。

This document is developed under support from the New Energy and Industrial Technology Development Organization (NEDO).

目次

1. ガイドライン全体概要

本章の内容は参考（informative）である。本章に含まれ、本ガイドラインの規範の一部を構成する（normative）内容については、後の章で再掲する。

本概要の全体構成は以下の通りである。第 2 章以降の本編に対応する内容がある場合には、その対応する章節も示す。

- 1.1 節では、本ガイドラインを作成した背景と目的を示す。
- 1.2 節では、本ガイドラインが主に想定する「使われ方」を提示する。（本編 2 章）
- 1.3 節では、背景として掲げた「AI の品質管理が困難である理由」を分析する。
- 1.4 節では、本ガイドラインがベースとする、品質管理のプロセスについての全体的な考え方を述べる。
- 1.5 節では、本ガイドラインが「目標」として設定する 3 つの「外部品質」の観点（実装手段にあまり依存せず、使用を通じてしか評価できない品質観点）を提示する。（本編 3 章）
- 1.6 節では前節を補足して、一般に「AI の品質」として議論される要素のうち、前節で採用しなかった要素について、その理由や本ガイドラインでの考え方を説明する。
- 1.7 節では、本ガイドラインが「管理手段」として設定する 9 つの「内部品質」の観点（実装手段に依存し、計測や作り込みでの管理が可能な品質観点）を提示する。（本編 6 章・7 章）
- 1.8 節では、本ガイドラインを作成するに当たって想定する開発プロセスの全体のイメージを提示する（本編 4 章・5 章）
- 1.9 節では、外部の規範文書などとの関係を明らかにする。（本編 8 章）
- 1.10 節では、本ガイドラインの残りの部分の構成を整理する。

1.1 目的と背景

人工知能（AI）、とりわけ機械学習技術は、製造業、自動運転、ロボット、ヘルスケア、金融、小売などの幅広い応用分野で有効性が確認され、社会実装が本格化する兆しを見せている。その一方、AI を利用した製品・サービスの品質を測定し説明する技術の不足に起因

し、万が一の事故の際に原因が特定できず、また投資に見合う AI システムの優位性を説明できず、結果として、社会的な受容性を得るための制度設計の遅れや、AI 開発ビジネス拡大への大きな障害となっている。

本ガイドラインは、機械学習を利用したシステム、特にその中に含まれる機械学習で実装されたソフトウェアコンポーネント（機械学習要素）の品質に関する基準と達成目標を定めることにより、企業が自ら構築した AI を利用するシステムの品質を測定し向上させ、また AI の誤判断による事故や経済損失などを減少させる一助となる事を目的とする。さらに、このようなアプローチを取ることで、達成した品質の認識を開発のステークホルダ間で共有したり、社会に対し具体的に示し説明することができるようになったりすることで、受発注などの条件の明確化や問題点の特定、高品質による付加価値の提示などが実現し、「良いシステムを作る事業者がより高く評価される」健全なビジネス環境の実現にも寄与することを目的とする。

1.2　本ガイドラインの使われ方

本ガイドラインが想定する一次的な読者は、機械学習を利用して作られる製品やサービスの提供者（ここでは「サービス開発者」とする [1]）と、実際に製品・サービスをソフトウェアとして実装するシステム開発者である。このサービス提供者とシステム開発者は、自社で製品・サービスを開発しエンドユーザに提供する単一の主体（本ガイドラインでは、「自己開発者」と呼ぶ。）である場合もあれば、契約に基づく分担（請負や準委任）の関係により異なる主体である場合も有り得る。さらに、システム開発者の中でも例えばシステム全体の設計開発と機械学習コンポーネント部分の実装などについて、さらに個別の分担関係がある場合もありえる。製品・サービスが利用される状況に応じて必要とされる品質に関して、これらの主体が明確な目標を共有し、システム開発プロセスの全体を通じてその品質を実現するために参照されることが、本ガイドラインが主に想定する利用形態である。

さらに二次的な利用形態として、その製品やサービスの利用者に対し、サービス提供者が本ガイドラインに従って設定し達成した品質を示すことで、安心して利用できる判断根拠を示すことが想定される。その延長として、われわれの社会全体が製品・サービスに一定の

[1] ソフトウェア事業者などが、他者が提供・自己使用するサービスを想定したシステムを予め企画・開発しておき、パッケージとして、あるいはカスタマイズを行って販売するような事業形態については、この企画・開発を行う主体をサービス提供者に準ずるものとして扱う。

品質を求める目的で、規格策定団体や第三者検証機関などが品質を評価し認定する基準を定める際の技術的な出発点となることも将来的には期待される。

このガイドラインは、できるだけ広範な応用に適用できるような汎用的な記述内容になっており、利用者が具体的な応用に即して、記述内容を取捨選択・具体化して用いることを想定している。実際の利用場面においては、各利用者の状況に応じて、応用分野ごとにこのガイドラインの内容を具体化した個別ガイドラインを作ることを想定している。本ガイドライン検討プロジェクトでも、いくつかの事例について、参照事例を作成・公表することを計画している。

本ガイドラインの具体的な利用形態としては、例えば次のようなものが挙げられる。

- AI システム開発に関する体制構築（契約など）の場面
  - 機械学習を利用するシステムまたはその一部となる機械学習の製品要素の開発を依頼する主体（本ガイドラインでは「開発依頼者」と呼ぶ[2]。）が、ガイドラインを開発対象の応用に合わせて具体化し、受託する主体（「開発協力者」とよぶ。）に対して契約要件となる仕様として指定する。
  - 開発協力者となる予定の者が、開発依頼者に対して品質を保証するための工程管理の標準として参照し、工数や受注価格を算出する根拠あるいは参考とする。
- 設計・開発の場面
  - 機械学習利用システムまたは機械学習要素を設計・開発する者（開発協力者または自己開発者）が、その開発の工程設計・システム設計及び品質管理の基準とする。
- 社会的な位置づけ
  - 自己開発者または開発依頼者が、システムを社会に提供・自己業務で活用する際に、社会規範としての品質要求に対する自己適合宣言の拠り所とする。
  - 将来的に機械学習利用システムの品質に関する社会合意としての基準とする。
  - 機械学習利用システムの品質に関する第三者評価制度を設計・運用する際の基準とする。

[2] 経済産業省がまとめた「AI・データの利用に関する契約ガイドライン」は、主に B2B（ビジネス当事者間）の開発契約の視点からまとめられており、本ガイドラインの「開発依頼者」を『ユーザ』と、「開発協力者」を『ベンダ』または『SIer』と整理している。本ガイドラインでは最終的な利用時品質の享受者を B2C（ビジネス～消費者間）の製品・サービス利用者に置く立場から、「ユーザ」の語は「Customer」にあたる製品・サービス利用者を指すものとしてのみ用いる。

なお、本ガイドラインでは、機械学習要素の作り方として主に「教師あり学習」(supervised learning) による実装を想定している。基本的な考え方はその他の実装方法、例えば教師なし学習 (unsupervised learning)、半教師あり学習 (semi-supervised learning)、強化学習 (reinforcement learning) などにも通用するものもあるが、これらの具体的な扱い方については、今後の改訂時に記述を追加する予定である。

1.3 機械学習の品質管理に関する課題

機械学習による人工知能の実装は、単純に言ってしまえばソフトウェアの一種であるといえる。しかし、従来のソフトウェアに対する品質管理の考え方だけでは、機械学習を利用したシステムの品質を向上させ維持していくには技術的に不足する点がいくつかある。本節では、そのような「従来ソフトウェアとの差異」について、いくつかの観点から課題を整理する。

1.3.1 環境分析の重要性

機械学習を利用する製品やサービスは、往々にして人間がその複雑さを把握しきれないような環境条件で用いられることが多い。全ての環境条件の複雑さを人が分析・特定し判断ルールとして逐一プログラムコードを実装しなければならない通常のプログラムと比較して、環境条件の子細を人が分析作業により特定しなくてもデータから自動的に判断ルールを構築できる機械学習技術は、高い複雑さを持つ環境条件で動作するシステムの実装を効率的に行う手段として大いに期待されている。

一方で、理想的な品質管理の観点、特に稀な環境条件への適合性が問われるようなシステムの品質管理の観点からは、システム設計の初期段階でできるだけ環境条件を緻密に分析し、リスクなどを把握しておくことが望ましい。機械学習技術をこのようなシステムに利用すると、従来はプログラムコードの実装時に行われていた環境条件の分析が省かれることになることから、初期段階での環境分析がより高い重要性を持つことになる。

このような実装の効率化と環境適合性の確保の 2 つの特質を両立させる上で、利用状況・環境条件の分析を設計段階でどのような詳細度で行うべきかは、従来のソフトウェア開発との相違点も含めて、システム設計の初期において検討すべき重要な事柄となる。

1.3.2　継続的なリスクアセスメント

一般に実社会で動作し、いわゆるサイバーフィジカルシステム（cyber physical systems, 以下「CPS」という。）の一部を構成するシステムにおいては、通常のソフトウェアシスエムに存在するリスクに加え、外部の環境変化に起因するリスクが常にある。未知の状況をある程度論理的に分析し想定・対策できる可能性のある通常のソフトウェア実装と異なり、実在するデータを元に構築する機械学習利用システムは、（その汎化性能に期待することはあっても）周囲状況の変化に起因するデータ傾向の大きな変化に追従できない可能性がある。

また一般に、機械学習を利用したシステムにおいては、過去のデータに基づく初期の訓練段階だけでなく、運用開始時の実データを用いた追加的学習を経て初めて実用的な品質を達成するような設計がされる場合も想定できる。

このような観点から、機械学習を利用するシステムにおいては、いわゆる「バグ」「初期不良」への対応として必要な修正だけでなく、当初から運用開始後の状況変化への対応を想定し、開発・運用を一体化した継続的ライフサイクルプロセスを導入することが必要となる場合が多い。開発計画の初期段階であらかじめ運用段階を含むライフサイクルを想定し、運用計画や受発注の契約内容などにも運用段階で必要な作業を折り込んでおく必要がある。

ただし、品質管理を継続的に行う場合においても、初期構築された運用開始時点での成果物について一定の品質の担保は必要である。特に、安全性などへの脅威となるリスクを伴うシステムにおいては、初期段階で一定の品質を確保することは不可欠である。本ガイドラインではこのような観点から、特に実装段階から運用を開始するまでに至る段階での品質検査に重点を置いている。さらに、運用の初期と本格運用時でシステムの運用形態（例えば人間による監視・介入の有無などの回避可能性の状況）を変化させることで、リスクの影響を変化させ、必要となる品質レベルを変更することも考えられる。このような場合には、運用形態を変化させる時点と品質管理の目標を変更する時点を同期させる必要がある。

また、運用時点でシステムの実装を更新する場合、その更新が品質を向上させる目的であっても、時によっては品質がかえって劣化する（ソフトウェア工学におけるリグレッション）リスクもある。そのことから、何らかの形での運用時の品質監視・対策も必要になると考えられる。そのような対策は開発・運用の形態によって異なることから、いくつかの運用モデルを 4.3 節に整理した。

1.3.3　データに依存した品質確保

機械学習などのデータ主導による開発において、構築に用いる「データの品質」に関する要求はしばしば言及される。本ガイドラインでは、データの品質そのものは品質管理における最終的な目的ではなく、品質劣化の原因や品質確保の手段であるとの考え方に立つ。もちろん、実際に機械学習の品質を、特に本ガイドラインのようなライフサイクルプロセス管理を通じて確保するためには、データに一定の品質を確保することがほぼ必須の手段となる。また、機械学習利用システムを分業で開発する際には、学習用のデータそのものが売買や開発役割分担の対象となることがあり、このような場合には「データの品質」を単体の性質として議論する余地がある。更には、最近では学会などにおいて、不正データの意図的混入による学習結果の汚染のセキュリティリスクも指摘されている。このような不正データの影響は単純な数値評価では検出できず、データの出所についての何らかの広い視点からの担保が必要である。

本ガイドラインはできるだけ品質を数値評価で担保することを是としつつも、必要な場面ではデータに対する定性的な品質管理を通じて品質を実現することが必要となると考える。

1.4　品質管理の基本的な考え方

参考) 本ガイドラインでは、システム全体で最終的な利用者に提供すべき品質として「利用時品質」を捉える。他方、実際にシステムが提供する品質としては、Systems Engineering の考え方から、システムの構成要素を階層的に整理し、その構成要素毎に「外部品質」「内部品質」を考える [3]。

各構成要素の「外部品質」は、その構成要素がシステムの部品として要求される、客観的な視点の品質のことを言い、例えば、セキュリティ、信頼性、一貫性などが挙げられる。一般論としてこの外部品質は、定性的や定量的なものを含んでいて、必ずしも、計測可能な単一の指標によって示せるものではない。

[3] 本ガイドラインで用いる「外部品質」「内部品質」の語は、ISO/IEC 9126 およびその後継である ISO/IEC 25000 シリーズにおいて用いられる External/Internal Quality の語とは、厳密には一致しない。特に外部品質に関して本ガイドラインは、要求される品質の「レベル」を設定するが、品質指標として必ずしも直接的に測定できるものではないと考える点に、注意が必要である。

一方で、各要素の「内部品質」は、構成要素を作成する際に具体的に測定したり、設計などの開発行為を通じて評価したりする、その要素が固有で持つ特性としての品質のことを言う。

このような整理をもとに、各要素の外部品質はその要素の内部品質の向上を通じて実現され、1 つ外側の要素の内部品質を達成するために要求されるという、図 1 のような階層的な品質モデルを想定する。システム全体の「外部品質」は、最終的な製品・サービス利用者から見た「利用時品質」のために、製品・サービスの提供者が実現し提供するものとして考えられる。

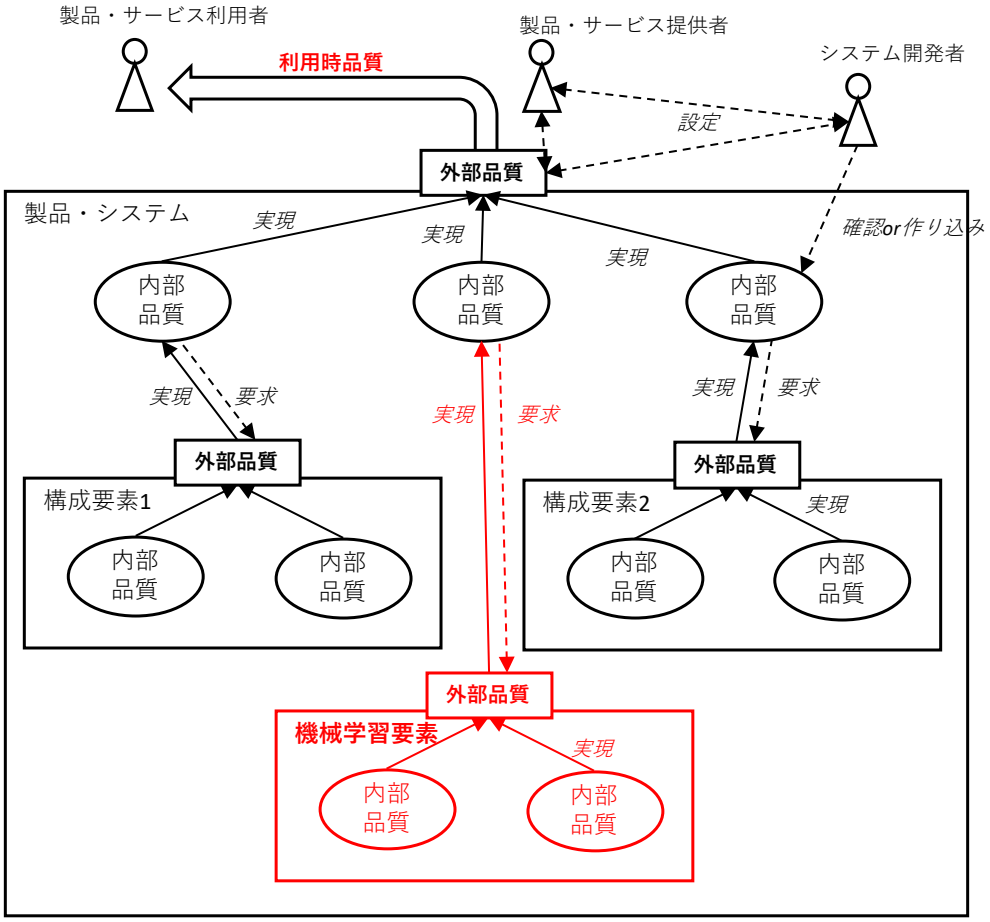

図 1　階層的な品質モデル

本ガイドラインでは、機械学習システムにおける「品質」そのものを、

- システムがその全体として利用時に満たすことが期待される「利用時品質」

- システムのうち機械学習で構築された構成要素が満たすことが期待される「外部品質」
- 機械学習による構成要素が固有に持つ「内部品質」

の 3 つに分けて理解し、機械学習要素の「内部品質」の向上を通じてその「外部品質」を必要となるレベルで達成し、最終的なシステムの「利用時品質」を実現するものと整理する（図 2）。

品質管理の目標とする機械学習要素の外部品質としては 1.5 節に掲げる 3 つの観点を設定した。そして、その達成手段としての内部品質としては機械学習要素特有の観点に注視し、現時点では 1.7 節に掲げる 9 つの観点をそれぞれ抽出した。外部品質の 3 つの観点に対して品質目標のレベル付けを行い、そのレベルに応じて 9 つの内部品質観点それぞれに達成目標を設定し、様々な技術や開発プロセス管理を通じてその目標を達成するという流れが、本ガイドラインにおける品質管理の基本的な考え方となる。

なお、最終的に実現するシステム全体の利用時品質については、その応用毎に着目すべき内容が異なることから、本ガイドラインでは具体的に規定しない（下記補足も参照）。

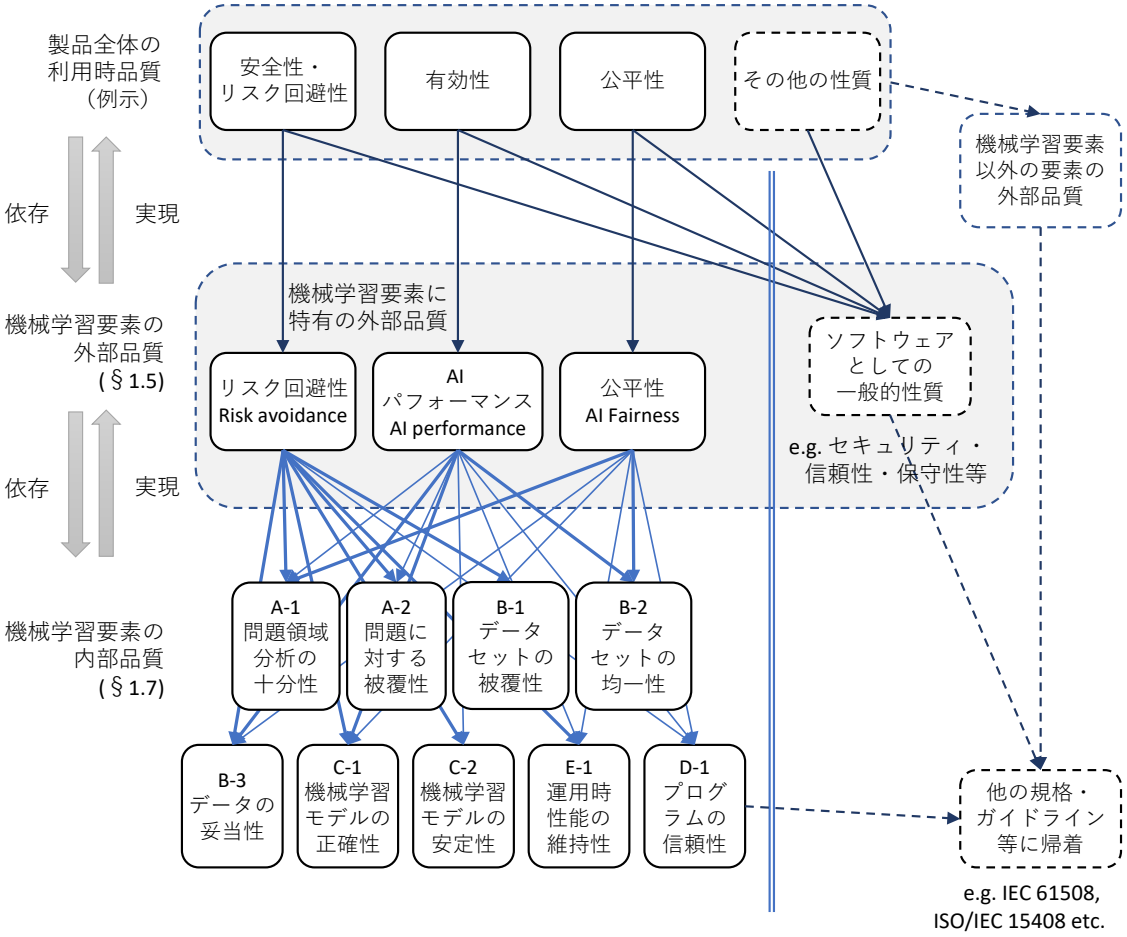

図 2: 製品品質実現の全体構造

例 1）

自動運転を行う自動車に搭載される前方映像からの物体認識モジュールを想定する。自動車全体の利用時品質の 1 つとしては「運転可能な環境条件下で障害物に衝突しない安全性」が考えられ、それを実現する為に物体認識モジュールには、「想定される天候や時間帯などの全てにおいて、物体を正しく認識する」性質が外部品質として求められる。それを実現する内部品質として、例えば学習状況の網羅性などがあり、これを機械学習の訓練用データの構成プロセスなどにより実現する。

例 2）

株の自動取引を行うサービスに内包される、株価予想 AI を想定する。サービス全体の利用時品質の 1 つとしては「利益の最大化」が考えられ、株価予想を行うモジュールには、「株価予測の誤差の最小化や、想定される取引結果の総和を最大化する」

などの性質が外部品質として求められる。それを実現する内部品質としては、例えば機械学習の推論精度などがあり、これを機械学習の訓練パラメータの最適化などを通じて実現する。

（補足）IEC 61508 [12] や IEC 62278 [15] などの、産業システム全体の安全性・信頼性の評価を行う厳格なプロセスを用いて開発する場面においては、従来からのシステム全体の設計プロセスにおいて、その一部の構成要素としての機械学習要素に対する外部品質要求は当然に特定される。

一方で、特に機械学習技術の需要が大きい IT 系サービスなどの応用においては、産業システムほど想定されるリスクが大きくない場合が多く、厳格なリスク管理プロセスをシステム開発全体にわたって適用することが必要でないこともあるだろう。

また、いわゆる人工知能技術全般の使い方として、「賢い判断」を機械学習要素に多少なりとも期待している場合、機械学習要素の外部品質のうち、特に本節で掲げた 3 つの観点は、システム全体の外部品質や利用時品質に、ある程度直接的な対応が考えられるような場合が多いことも想定される。上に掲げた 2 つの例は、そのような事例の具体的な例示になっている。

そのような観察から、図 2 ではシステム全体の利用時品質と機械学習要素の外部品質に対応があるような表記とし、また利用時品質として「安全性・リスク回避性」「有効性」「公平性」の 3 つを例示した。

1.5　実現目標とする外部品質特性

本ガイドラインでは、以下の 3 つの異なる性質（リスク回避性、AI パフォーマンス、公平性）を機械学習要素の外部品質特性の軸として抽出し、それぞれに品質レベルを設定した。

1.5.1　リスク回避性

『リスク回避性』（危害・危険回避性／安全性）は、機械学習要素が望ましくない判断動作を行うことを抑制し、システムを用いたサービス提供者・システムにより提供されるサー

ビスの利用者または第三者などに人的被害や経済損失・機会損失などの悪影響を及ぼすリスクを低減する品質特性である。昨今、一層重要視されている「セーフティ＆セキュリティ」や「安全・安心」に深く関係する。

具体的な事例として、

- 自動運転のための物体認識における、物体の見落としやその種別の誤認識
- 食品工場ラインの混入物検知における異物の見落とし
- 有価証券の自動取引における、不正な入力（見せ玉など）による許容範囲を超えた発注

などが挙げられる。

リスク回避性については、多数の人命や企業体の存続に影響するレベルから、軽微な利益機会の逸失に留まるレベルまで、7 レベルを設定した（3.1 節）。リスク回避性の特性は、従来のシステムの安全性規格などで扱われてきた性質と密接に関係することから、品質レベルの設定に当たって、これらの規格との併用や親和性を考慮し、主に強い要求については、既存規格（IEC 61508-1 [12] の SIL など）と対応する 4 レベル（レベル 4〜1）を設定した。一方で、IT サービスやスマートデバイスなどの機械学習の応用では、従来の工業製品でリスクとして捕捉しないような軽微な損害しか想定しないような需要も多く、これらの品質要求は既存安全性規格では「対象外」の単一レベルに対応してしまうことから、実用性の観点からさらに 3 レベルを追加することとした。

一般に機械学習システムにおいて、「どんなときにでも必ず安全な動作をする」といったような厳密な性質の保証は本質的に馴染まず、機械学習要素単体だけではシステム全体に必要の利用時品質を達成できないことも考えられる。実際のシステム構築においては、その実現を機械学習要素に依存せずに、周辺のソフトウェア実装による「安全確保弁」などに依ることも多いと想定される。本ガイドラインでは、従来の安全性の規格同様、システム全体の利用時品質の要求と機械学習要素の外部品質の要求を別個として認識し、システム全体のリスクアセスメントと、システム構成から機械学習要素の外部品質要求レベルを設定する（5.1.1.5 節）こととする。

1.5.2　AI パフォーマンス（有用性）

2 つ目の特性軸として、機械学習提供機能の有用性が重視される分野への応用に着目し、「**AI パフォーマンス**」の特性軸として整理する。リスク回避より AI パフォーマンスが重要視される典型的な応用としては、個別の誤判断による悪影響よりも平均的な成績の高さが

要求されるシナリオがあり、一例として小売店の仕入れにおける需要予測や、投資判断の予測などが挙げられる。

もちろん応用によっては、「AI パフォーマンス」と「リスク回避性」の双方が要求されることもありえる。例えば需要予測において、極端な過剰仕入れなどは運用環境の想定を超えた間接的な悪影響をもたらし、利潤の平均値の大小だけでは評価できない大きな経済損失を引き起こす場合なども考えられる。このような場合は、「AI パフォーマンス」と「リスク回避性」二つの特性軸について、最適な妥協（バランス）ポイントを、要件としても明確にすることが必要となる。

AI パフォーマンスについては、達成すべき具体的な目標値そのものはいわゆる Key Performance Indicator（KPI）として応用毎に異なることから、その目標達成がどの程度強く求められるかという観点から、3 つのレベルを設定した。

1.5.3　公平性

AI にはしばしば、ある種の社会規範性や倫理性が要求されることがある。人文科学的な立場ではなく工学的な観点からは、これらの社会規範性の多くは既存の様々な利用時品質特性と、その特性の設定過程に帰着されることが多い。そのような中で、特に近年の統計的な技術による AI に特有の、従来あまり考慮されていない品質観点として**「公平性」**を抽出することができる。

もちろん従来のソフトウェアでも「公平な判断」をしなければならない場面は多いが、その実現は主に設計段階での検討で完了しており、実装段階ではその設計通りに正しく動作する「信頼性」などの別の品質特性に帰着されていた。しかし機械学習においては、実装過程や学習結果そのものに確率・統計的な振る舞いが含まれるため、事前の検討だけでは公平性の実現を担保しきれず、ソフトウェア要素としての機械学習要素が、直接「公平性」を品質として実現する必要が生じる。

このような視点から、本ガイドラインは 3 つ目の外部品質特性として「公平性」を掲げる。本ガイドラインにおいて「公平性」とは、システムあるいは機械学習要素のレベルで規定された機能要件を基準に、機械学習要素が判断材料にすべき状況以外の要因により入力に異なる取扱いをしないこと、と整理する。公平性の事例としては、例えば健康保険の見積もりや人事採用のスコアリングにおける人種差や性別の取扱いなどに偏りが生じていないことを明示的・暗示的に要求されるケースが想定される。

より具体的には、

- 「公平性」（Fairness）は、システムあるいはコンポーネントが行う判断・分類等に要求される機能として定義された、判断材料にすべき状況に関する要件と比較して、複数の有り得る入力または入力を生み出す人などが、定義された要件以外の状況の差異に起因して、定義された何らかの基準で異なる取扱いをされないこと、と定義する。
  - なお、「定義された要件以外の状況」については、システムの機能定義によっては逆に「公平に扱うべき特定の状況の差異」を特定する場合も有り得る。また、「状況」には機械学習要素への入力の値に明示的に含まれる属性も、含まれないような隠れた特徴もある。
  - 「異なる取扱い」の基準については、システムに要求される公平性の要求の強さにより、例えば「意図的に異なる取扱いをしない」「統計的に差異がない」「アルゴリズム的に差がないことが保証される」など、様々な具体化が有り得る。
  - この「公平性」の定義を個別の事例に応じて詳細化する方法については、本文書の 8.5 節で更に具体的に述べる。
- これと対比して用いられる用語としての「バイアス」（Bias）は、実装された機械学習要素の出力が、入力のもつ様々な特徴に対して統計的に相関を持つことをいう。この定義では出力が定数関数または乱数的な関数でない限り何らかのバイアスを持つことになるが、意図しないバイアスの中には公平性を毀損するものが含まれる。

一般に機械学習利用システムに要求される社会的要求として、「FAST（fairness, accountability, sustainability, transparency）」の 4 要素が指摘されることがあるが、本ガイドラインではそのうちでも特に統計的な性質として直接的に分析可能な公平性にまず着目する。

公平性については、他の 2 つの性質に共通する分析に加えて、8 章で背景を含めた分析を行っている。

なお、どのような属性に対しても一定のリスク回避性や AI パフォーマンスが求められることについては、ここでは公平性とみなさず、リスク回避性や AI パフォーマンスとして整理する。例えば、安全性のための防護装置において性別や体格の違いの影響に関する品質の要求は、「どのような性別・体格であっても安全であるべきとするリスク回避性の要求」と整理し、「性別で安全性の差が出ない公平性の要求」とは整理しない。これは、公平性の技術的実現方法が他の性能指標とのトレードオフになることが十分考えられ、「一方の安全性を下げて公平性を向上する」といった解決が望ましくないためである。

また、本外部品質特性では、「どのような公平性が計測可能で品質管理の対象にできるか」と、「それらの公平性を、機械学習要素にどう実現するか」の 2 点を本ガイドラインの検討対象とし、「特定のシステムにおいて、どのような公平性が社会的正義を実現するか」はガイドラインのスコープ外であらかじめ検討すべき事柄とする。これらの社会規範性や倫理性については、1.6.3 節でさらに議論する。

1.6　その他の「AI 品質」の観点についての取扱い

本ガイドラインでは、上記の通り「リスク回避性」「AI パフォーマンス」「公平性」の 3 つの観点に着目して品質管理を行うこととしたが、一般に「AI の品質」としては他にも様々な観点が議論されている。本節では、これまで整理した 3 つ以外のいくつかの観点について、本ガイドラインとの関係の考え方を整理する。

1.6.1　AI の説明性

人工知能利用システムの「説明性」(explainability) は、信用性 (trustworthiness) の構成要素の 1 つとして重視されることがある。実際に説明性を品質と捉えることには、いくつかの側面があると考えられる。まず、「安心して使用できることの説明性」、つまり品質の説明性という観点では、本ガイドラインの立場ではこれを品質そのものと言うより、品質マネジメントに求められる性質として考えている。本ガイドライン全体を通じて品質マネジメントは、品質管理の活動を納得性を持って後から説明し、納得することを可能とするような一連の活動と捉えており、この観点での説明性は、まさにこのガイドラインの目的である。

次に、動作の内容が説明可能という性質と捉える立場からは、この意味での説明性は本ガイドラインが目的とする品質の説明性を達成のための 1 つの手段と考えられる。確かに機械学習システムの動作が完全に説明されれば、その動作を通常のプログラムと同じように論理的に分析し、その品質を説明できる可能性が高い。また、たとえ完璧に動作を説明できなくても、構造の単純な機械学習要素は、動作の内容の説明が容易であると同時に、予想と異なる動作を引き起こす品質上の懸念がより低くなることが期待される場合もあるだろう。しかし一方で、1.3.1 節で述べたように環境が複雑な場合、論理的に完璧に説明できるであろう通常のソフトウェアプログラムであっても、それが信頼性に繋がるとは限らないことも多く、動作の説明性と品質の説明性は、常に一致するものではないと考えられる。

1.6.1.1 説明できる人工知能（explainable AI）

動作の説明性の達成形態の 1 つとして、「説明できる AI」explainable AI の考え方が言及されることがよくある。論理的に AI の動作を再記述でき、かつそれが十分に理解可能なレベルに整理されるような「説明可能 AI」の技術が確立すれば、品質マネジメントの観点からも諸品質の達成を論理的に説明できる可能性が期待されている。しかし、少なくとも現時点においては、このような技術は研究途上であり、品質達成のための必須手段として安定してかつ汎用に採用できる段階には至っていないと考える。もちろん技術適用が可能である場合には、本ガイドラインとの関係においても有力な品質の実現手段となり得る。特に、公平性（8 章）に特有の内部品質や、機械学習モデルの安定性（6.7 節、内部品質 C-2）の説明を得るための手段としては、強いものになる可能性がある。また、上に記したとおり、アルゴリズムとしての機械学習要素の計算処理がたとえ完璧に説明できても、実社会での品質を常に説明できるとは限らないことは留意しておく必要がある。

1.6.1.2 透明性

同じ理由で、透明性 transparency も品質マネジメントの立場からは達成手段の 1 つであると考えられる。少なくとも機械学習の分野では、動作の透明性は動作の説明性とかなり類似する性質であり、公平性や安定性などの分析で有効になる可能性がある。また、機械学習の結果の機械学習モデルをそのまま用いず、得られたモデル内部のパラメータを元に論理設計を行いルールベースや通常のプログラムの形態で記述するような応用（本ガイドラインでは機械学習モデルとは考えない）においては、通常のプログラムとしての透明性・説明性は検討の対象になり得ると考えられる。

1.6.2 セキュリティ・プライバシー

機械学習利用システムのセキュリティには、「従来型の情報処理システムで対応されてきた脅威と脆弱性に対するセキュリティ」と「機械学習特有の脅威と脆弱性に対するセキュリティ」の 2 つの観点がある。

前者については、情報セキュリティの 3 要素として列挙される「機密性（confidentiality）」「完全性（integrity）」「可用性（availability）」を確保する必要がある。これについては、必要に応じて従来規格などにより、例えば ISO/IEC 15408 [3] の評価保証レベル（EAL）を

設定するなどして管理することとする。また、データの機密性に関しては、個人情報保護法の匿名加工データの取扱いなど、従来型の情報処理システムで対応されてきたプライバシー保護の対応を要する場合がある。

一方、「機械学習特有の脅威に対するセキュリティ」については、「機密性」と「完全性」に関する攻撃の低減策や防止策を検討する必要がある。

- 訓練済み機械学習モデルあるいはその学習に用いる訓練用データに意図的な改変を加えることにより、運用時の機械学習利用システムの完全性を低下させる攻撃として、「ポイズニング攻撃（poisoning attack）」が知られている。
- 運用時に敵対的データ（adversarial example）を入力することにより、機械学習利用システムの完全性を低下させる攻撃として、「回避攻撃（evasion attack）」が知られている。
- 機密性については、一定の条件下で訓練済み機械学習モデルの利用を通じて、訓練用データについての情報を窃取できる攻撃が指摘されており、個人情報・プライバシーの保護の観点から課題が生じる可能性がある。

なお、セキュリティ攻撃の分析と対策においては、セキュリティリスクアセスメントから対策に至る施策全体を定期的に実施する必要がある。その際、アセスメントの結果から優先度の高いものから実施するなど、実施規模を適切に調整する。

1.6.2.1　耐攻撃性

現在の機械学習では、入力データの意図的改変に対して訓練済み機械学習モデルが脆弱であることが、さまざまな研究において明らかにされている。さらに、画像カメラなどへの入力となるシステム外部の状況そのものの改変を通じて、そのような脆弱性への攻撃を行いうる可能性も指摘されている。一方で、その実運用上のリスクなどについてはまだ評価が定まっていない。また、そのような攻撃に対して対策を講じることが、通常運用の機械学習要素の性能に対しては負の影響を与える可能性も指摘されている。

本ガイドラインでは現時点においては、単独の品質要件として耐攻撃性を評価するよりは、前 1.5 節に掲げた品質特性軸のそれぞれについて、動作時の周辺環境に攻撃者が意図的な改変を加える可能性やその影響度合いをきちんと評価することが現実的であると考え、その上で、実運用の意図的改変をどのようにリスク評価・低減するかに応じて、品質マネジメントプロセスの中で意図的改変への対策の必要性を検討することとする。対策が必要な場合には、基本的には 1.7.7 節に掲げる「機械学習モデルの安定性」を通じてその効果を評

価することにした。その上で、防御策の検討などに参考になる攻撃の分析や防御手法などについて 9 章で検討する。

1.6.3 倫理性などの社会的側面

機械学習利用システムに求められる品質として、機械学習要素が行う判断処理結果の倫理性や社会的正当性が含まれることがある。例えば、個人情報に強く関連する分野（人事採用の事前スコアリング、犯罪予測など）においては、性別や人種などについて、法律的・社会的にその差異が判断に影響しないことを保証することが求められることがある。また、人的・経済的被害を与えるリスクが高い分野では、複数のリスクを伴う判断の選択肢の正当性が問われる場面（何らかの人的被害は不可避な状況下における自動運転システムの判断など）が有り得る。

本ガイドラインでは、このような場面で機械学習要素が「どのような判断を行えば」社会的正当性を持つかについては、システムの開発の最初に要求定義の一部として事前に人間によって整理されるべきものとして、その直接の検討の対象としない。その上で、機械学習利用システムが、人間が整理した「正しい」出力に向かって可能な限り高い確からしさで処理結果として導出することを、利用時品質の要求と捉える。

そのような中で、特に従来のソフトウェア工学では直接的な外部品質としてあまり扱われてこなかった「公平性」について、外部品質軸の 1 つとして 1.5.3 節および 8 章に掲げた。なお、公平性・透明性などの倫理的側面については参考として、国際的な取り組みを 10.2 節にまとめた。

1.6.4 外部環境の複雑性への対応限界

人工知能の一般的な問題として、対応できる環境の複雑性の限界に関する議論が着目されることがある。機械学習利用システムの利用形態には、路上や公共空間などの開放環境でいわゆるサイバーフィジカルシステムの一部として組み込まれる形態も多くあり、全ての環境条件において期待通りの判断を行う事は、極限的な状況も含めると不可能である。この問題は本質的には、機械学習に依らず、開放環境で動作する装置やソフトウェア全般に共通する問題であり、従来の信頼性工学関係の規格においても、例えば IEC 62998 [16] などがこの問題に多かれ少なかれ対応しようとしていると考えられる。

本ガイドラインでは、全体的な考え方は従前の規格などを踏襲しつつ、複雑な外部環境で

の品質の実現のために必要なリスク分析やシステム設計の妥当性などについて、従来のソフトウェアとの特性の差異や、それに起因する固有の分析・設計の留意点などに限って、品質管理手法の検討対象とする。

1.7　品質管理の対象とする内部品質特性

本ガイドラインでは、前記の 3 つの外部品質の達成に対応して、機械学習要素の内部品質の管理する具体的な特性として、以下の 5 分野 9 特性を品質管理の特性軸として抽出した[4]。この抽出に至る分析については、11.1 節にその概要を述べる。

- A: 品質構造・データセットの設計
  - A-1: 問題領域分析の十分性
  - A-2: データ設計の十分性
- B: データセットの品質
  - B-1: データセットの被覆性
  - B-2: データセットの均一性
  - B-3: データの妥当性
- C: 機械学習モデルの品質
  - C-1: 機械学習モデルの正確性
  - C-2: 機械学習モデルの安定性
- D: ソフトウェア実装の品質
  - D-1: プログラムの信頼性
- E: 運用時の品質
  - E-1: 運用時品質の維持性

本節では、主に「リスク回避性」「AI パフォーマンス」の 2 つの外部品質について、特性の考え方を説明する。「公平性」については、本節（および 6 章）に加えて、公平性に特有の検討を要する面もあるため、8 章において考え方を改めて整理する。

[4] 第 1 版から 1 特性を分割し、分野の構造を追加した。以前の内部品質特性 1〜8 は、A-1, A-2, B-1, B-2, (B-3 と C-1 に分割), C-2, D-1, E-1 に対応する。

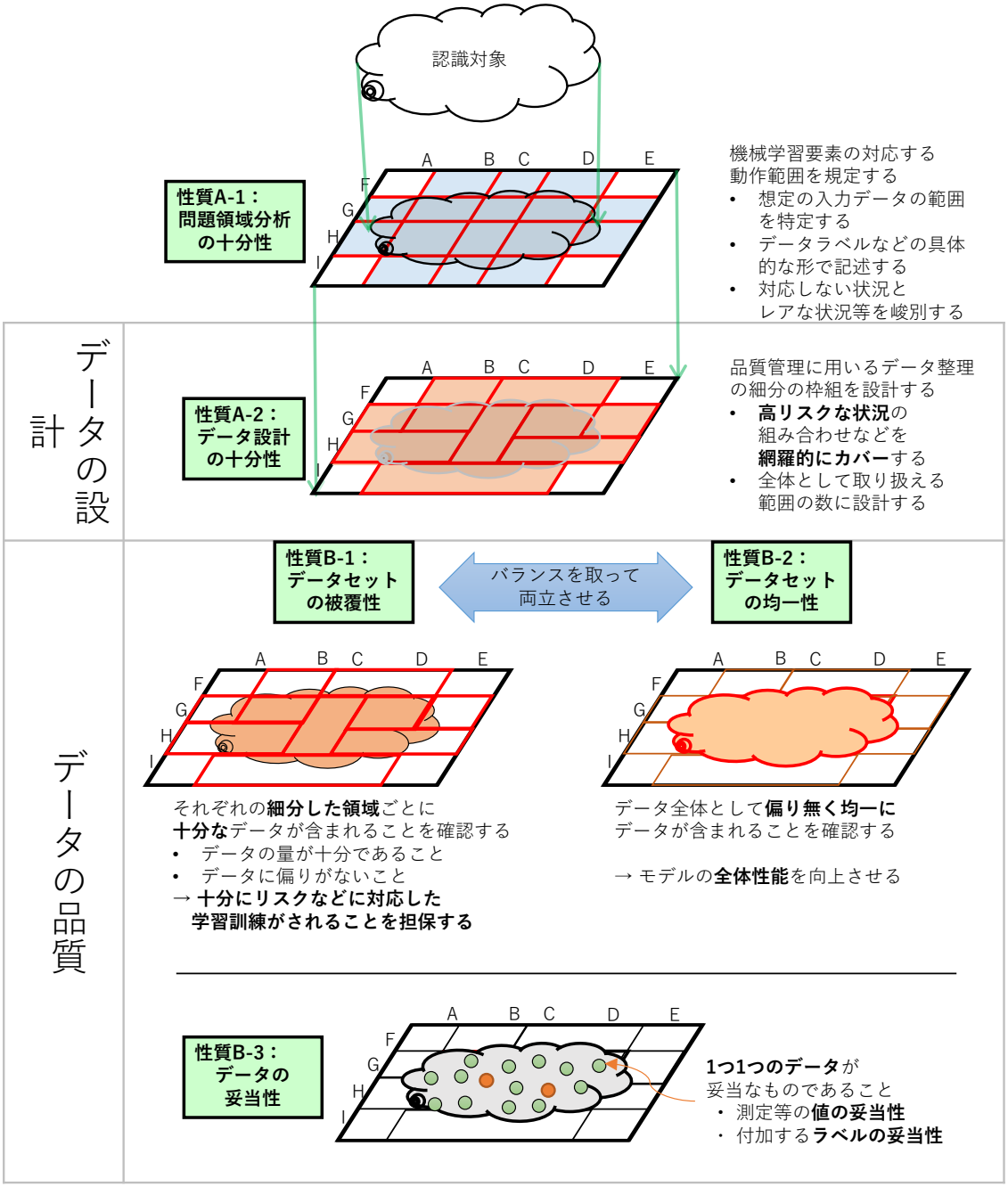

図 3: 着目する内部品質特性（1）

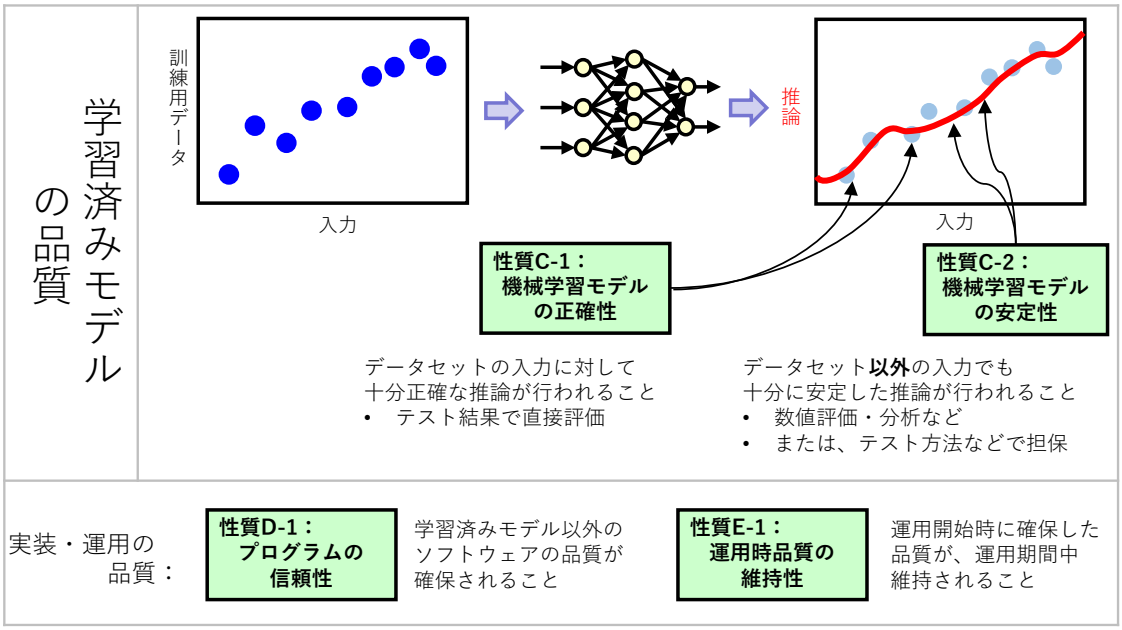

図 4: 着目する内部品質特性（2）

1.7.1 A-1: 問題領域分析の十分性

まず初めに、機械学習利用システムの実世界での利用状況に対応して機械学習要素に入力されると想定される運用時の実データの性質について分析が行われ、その分析結果が想定される全ての利用状況を被覆していることを、「問題領域分析の十分性」（sufficiency of problem domain analysis）とする。

通常の開発プロセスにおいて、基本的な要求分析は上流設計の一部として完了していると考えられる。この段階においては、事前の要求分析段階で特定された対象システムの動作領域に対して、後の段階で訓練用データの整理や必要な検査用データの有無の確認などに用いられるよう、具体的な個々のデータと紐付けられるレベルまで、その分析した要件を具体的な言葉で書き下し、データを元に品質を分析する「軸」を定めることを目標とする。

このような具体的なデータ分析の「軸」の定め方にはいくつかの方法が有り得るが、本ガイドラインの基本的な考え方としては、従来からあるソフトウェアプロダクトライン工学における feature tree の考え方や、それを簡易化した、下の例に挙げるようないくつかの独立した条件の箇条書きとして分類整理し、特定の利用状況をそれらの組み合わせとして把握する手法を想定する。また、この段階でリスク分析・故障モード分析などの手法による要求側からのトップダウン分析と、PoC（Proof of Concept）段階での予備的なデータ分析によるボトムアップな分析を両方行い、外部品質が変化する（特に劣化する）可能性のある状

況などを十分に整理する。

（例 1）
屋外で走行する自動運転車両において、交通信号機の現示を画像から認識する機械学習要素においては、遭遇する状況として例えば以下のような書き出しができる。
（この例は実応用上十分に網羅的ではない）

表示の意味： 緑色、黄色、赤色
時間帯： 昼間、夜間
天候条件： 晴れ、曇り、小雨、大雨、雪、霧 など
その他

このように分析すると、例えば「**晴れ**の**夜間**に、**赤色**の信号が点灯している」状況が、1 つの「状況の組み合わせ」となる。

（例 2）
小売店の販売傾向の予測を行う機械学習要素においては、その利用状況について例えば以下のような書き出しができる。（この例も、網羅的ではない）

曜日： 月曜日・週中日・金曜日・土曜日・休日
天候条件： 晴れ、曇り、小雨、大雨、雪
時間帯： 朝、午前、昼、午後、夕、夜、深夜
季節： 春、夏、秋、冬
近隣のイベント： あり、なし

この例で、例えば「**冬**の**晴天**下の**休日**の**朝**に近隣イベントが**ある**」が 1 つの組み合わせとなる。

この要求分析の十分性は、従来型のソフトウェアにおけるリスク要因の分析や、ブラックボックステストを行う際にそのリスク要因をきちんと検査対象にするためのテスト要件の分析に対応し、品質を把握し確認する単位を規定する重要な特性となる。一方で、機械学習の実装においては、ある程度以上に些細な特徴は機械学習の訓練段階での学習処理に委ね、システムの実装者が個別の条件判定を細かく指示しないことが最大のメリットでもあり、また場合によっては人による実装よりも高い性能を期待することがありえる。このような観点から、「どの程度の詳細さで要件分析を行うか」の設定は、品質を考慮した機械学習要素の実装戦略を考える上で極めて重要なポイントとなる。

また、このような分析を行うと、実環境で実際には遭遇しない・極めて稀で動作を保証することが現実的でない（対応を要求されない）状況（例えば、夏期の雪）や、逆に収集する訓練用データに出現しないが対応しなければならない稀な状況（例えば、東京地方での春の雪）などが識別できることがある。具体的に、このような 2 つの状況を峻別することは、特にリスク回避性を要求されるシステムにおいては極めて重要で、システム全体の安全性・堅牢性の設計と直結する。実環境から収集したデータだけでは発見が困難な対応すべき稀な状況（レアケース）の状況を特定し、同時にまた対象システムの設計を進める過程で、実際には発生し得ない状況を定義し、後の段階での検討対象から排除することも、この「要求分析の十分性」を考える段階での重要な作業となる。

具体的な設定の考え方については、6.1 節でさらに詳しく述べる。

1.7.2　A-2: 問題に対する被覆性

問題領域分析の十分性を前提として、システムが対応すべき様々な状況に対して十分な訓練用データやテスト用データを収集し整理するためのデータ設計の十分な検討を、「問題に対する被覆性」（coverage of distinguished problem cases）として要求する。

極めて単純なシステムでは、前項の問題領域分析において特定された「状況の組み合わせ」の全てに対して、各々対応するデータが訓練用データセットやテスト用データセットなどに含まれていることをいえばよい。しかし、システムの想定する利用状況が複雑な場合には、取りうる組みあわせの数が膨大になるため、全ての組み合わせについてデータセットで網羅することは現実的でない（例えば、上記の例 2 の簡易な事例だけでも、組み合わせの総数は 700 になる。実際の事例では、1 万のオーダになることも想定される）。その場合には、複数の状況を包括する粗い粒度レベルで広く網羅性を確認しつつ、危害や性能低下の可能性が高い細かな状況の組み合わせにも十分に対応し漏れがないことが必要になる。ソフトウェア工学の分野ではテスト設計などに「網羅性基準」といった考え方があり、応用毎に適切な手段を選択することで、現実的かつ実用上十分な網羅性を達成することを目標とする。

1.7.3　B-1: データセットの被覆性

前項で設計した「対応すべき状況の組み合わせ」の各々に対して、状況の抜け漏れがなく、十分な量のデータが与えられていることを、「データセットの被覆性」（coverage of dataset）とする。品質管理の観点からは主にテスト用・バリデーション用データセットについて着目

するが、品質を実際に達成するためには訓練用データセットにおいても重要な性質である。

通常のソフトウェア開発においては、ソフトウェアの動作が依存する全ての実世界の特徴の子細については、要求分析から実装までの少なくともいずれかの段階で把握され、最終的にプログラム内の条件分岐や計算式などとして反映されることになるが、機械学習要素の構築においては、ある程度より子細な状況は、訓練用データセットの「特徴量」や「正解ラベル」などとして明示的に把握されず、訓練用データセット内に暗黙的に含まれるのみであり、機械学習の訓練段階を通じて最終的な動作に反映されることになる。要求分析やデータ設計において特定された状況やケースについて、データの不足による学習不足や、偏ったデータによる特定の状況への学習漏れが起こらないことを保証するのが、本特性軸を設定する目的である。

（例 1）
前記の交通信号機の画像認識のケースでは、各都道府県の設置形態や信号機の塗色、設置高や距離、前後の道路の形態などが偏り無くデータとして含まれ、例えば特定の市の狭い範囲のデータのみで訓練していないことがこの特性に相当する。

（例 2）
街中の小動物から「猫」を認識する画像認識 AI を構築する際に、猫の品種や体長などを個別の特徴として認識しないこととした際に、その製品が利用が想定される環境において十分に多様な「猫」や「犬など他の小動物」の画像がデータとして用意されること、例えば三毛猫など特定の品種だけが学習対象になっていないことが、この特性に当たる。

1.7.4　B-2: データセットの均一性

上記の「被覆性」と対となる概念として、想定する入力データ集合全体に対する「データセットの均一性」（uniformity of datasets）がある。データセット内の各状況や各ケースが、入力されるデータ全体におけるそれらの発生頻度に応じて抽出されているとき、均一であるとする（図 5）。この均一性と、先述の被覆性の間のバランスが評価の対象となる。機械学習の技術は一般には、入力環境に対して均一に抽出したサンプルを訓練用データセットとして用いれば予測精度は高くなるとされる。しかし、実際の応用やそこで求められる品質特性によっては、前項の状況に対する被覆性が重要視される場合もある。被覆性と本項の全

体に対する均一性のどちらを優先するか、またどのように両立させるかは考慮する必要がある。

一般論として、リスク回避性が強く求められるケースでは、正しい判断で回避すべき高リスクな状況に対して十分な訓練用データがあることが求められるであろうが、特にそのような状況が稀に発生する場合、その稀なケースに対する十分なデータ量を確保しつつ、他の全ての状況について均一性を保ちながら訓練しようとすると、必要なデータ量が膨大になる可能性がある。このような場合、特に稀で高リスクな状況を重点的に訓練することは十分に考えられる。

一方で、全体的な性能（AI パフォーマンス）が求められる場合には、稀なケースを実際の発現確率以上に重点的に訓練することで、他のケースにおける推論結果の確からしさが劣化し、全体としての平均性能を悪化させる可能性もある。このような場合には、前節の状況ごとの被覆性基準は適切でないといえる。

また、公平性が強く求められる場合には、「どのような公平性」が求められているかに依存して、データを選別あるいは追加するといった人為的な処理を行ってケース間で人工的に同等な学習をさせるべきか、抽出された訓練用データの分布に即して無作為に学習をさせるべきかが変わってくる可能性がある。

このように、前 1.7.3 節と本節の 2 つの観点は、両立することも相反することもあり、妥当なレベルで両立させるための適切な訓練用データの調整が求められる可能性がある。また、訓練の段階と品質検査の段階でも、求められる性質が違うことも有り得る。

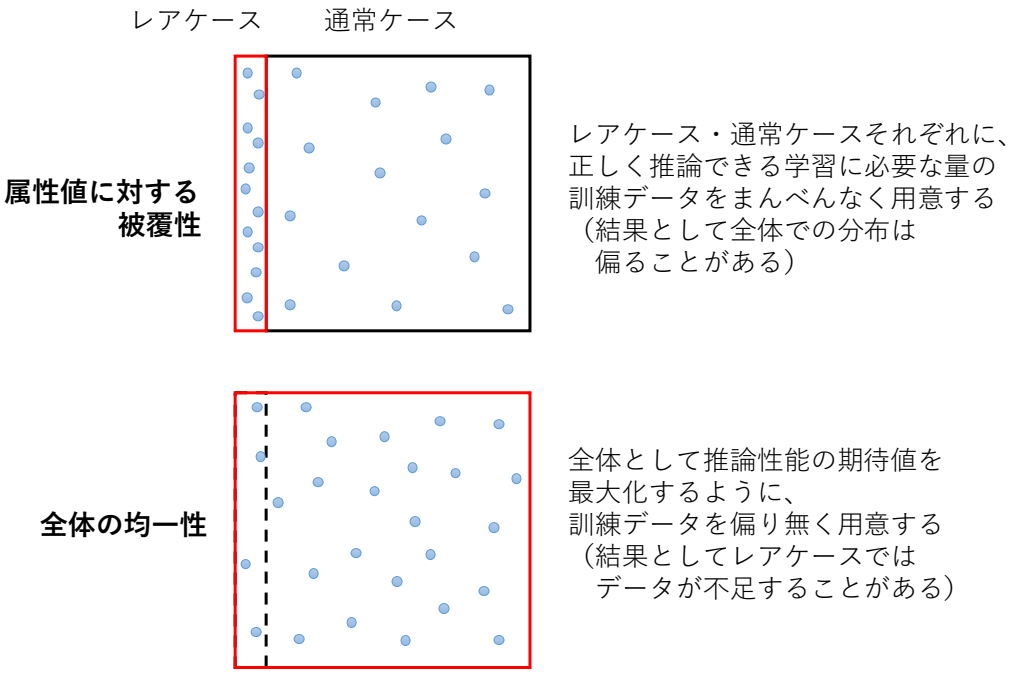

図 5: 被覆性と均一性の関係

（例 1）
例えば自動運転車において積雪が信号画像認識の性能に影響があり得る場合、運用地域として設定した東京で年に 1〜2 日程度と想定される積雪の状況であっても、誤認識の影響を押さえるために必要な、学習または検証データを用意しなければならない。このような理由で、雪の画像の含まれる割合を実際の雪天の発生確率より多くする場合が有り得る。
この場合、「データセットの被覆性」を「データセットの均一性」より優先する必要が考えられる。

（例 2）
一方で、同じ東京でも小売店の売り上げ予測においては、このような年 1〜2 日程度の雪の事例を強く訓練することで、他の天候における予測性能を悪化させ、平均の利潤を最大化できない可能性もある。この場合、「データセットの均一性」のほうを重視する可能性が考えられる。もちろん、実際にどのようなデータを訓練用データセットとして用意するかは、最終的に双方の内部品質特性のバランスを考えながら実装工程で検討することになる。

1.7.5　B-3: データの妥当性

B-1・B-2 のデータセットの分布に関する性質とは対照的に、データセット中のデータ 1 つ 1 つが訓練の目的に照らして妥当であることを、「データの妥当性」(Adequacy of data)とする。 妥当性には、値に誤りが無いことだけでなく、訓練に使われるべきでないデータが（たとえ値そのものが正確であっても）含まれないこと（一貫性)、データに不適切な改変などがされていないこと（信憑性)、データが十分適切に新しいものであること（最新性）などを含む。また、教師あり機械学習の文脈では、訓練対象としての測定値等（機械学習要素を関数と見たときの入力側にあたる値）の妥当性である「データ選択妥当性」と、訓練用に付加された正解値（出力側にあたる値）の妥当性である「ラベリングの適切性」の 2 つの観点が含まれる。

データの妥当性の品質は、基本的には機械学習要素を実装する際の要件定義に依存して判断される。部分的には汎用のデータとしての妥当性として判断できるもの（明らかに妥当でないデータの除去や、データセット全体としての信憑性や追跡可能性）などもあるが、一

般的には要件定義が更新されたときには品質の再判断を要する。1 つのシステムの開発プロセスにおいても、PoC 段階での試行や本番での手戻りなどにより要件定義やラベリングのポリシーなどが更新された際には、その新しい要件・ポリシーに対応したデータの再整理が必要となるため、開発工程においては工数や手順を十分に検討しておく必要がある。

本ガイドラインでは、できるだけデータの検査などにより妥当性を評価することを目指すものであるが、信憑性や追跡可能性のようにデータそのものからは確認が特に難しいものや、ラベリングポリシーの統一などプロセス管理で達成すべきものもある。

1.7.6 C-1: 機械学習モデルの正確性

B-3 までで一定の品質を担保されたデータセット（訓練用データセット、テスト用データセット、バリデーション用データセットからなる）に含まれる具体的な入力データに対して、機械学習要素が期待通りの反応を示すことを、「機械学習モデルの正確性」（correctness of the trained model）とする。

通常、学習に用いられる訓練用データセットは、運用時に環境から与えられる入力データのすべてを捉えきれず偏りを含むこともある。そのため、機械学習モデルが訓練用データに対してのみ性能が高く、それ以外のデータに対しては性能が低いという場合（過学習）がありえる。学習データセットのみを用いた評価では、環境からの入力データ一般に対して機械学習要素が意図した動作を行うかどうかを確認できるわけではない。そこで、本項で掲げる正確性と、次項において取り上げるモデルの安定性はその両立をはかることが重要となる。

1.7.7 C-2: 機械学習モデルの安定性

学習データセットに含まれない入力データに対して、機械学習要素が期待する反応を示すことを、「機械学習モデルの安定性」（stability of the training model）と称する。低い汎化能力や敵対的データによる予測不可能な振る舞いを排除することにより、機械学習要素の振る舞いの予測可能性を高める。

1.7.8 D-1: プログラムの信頼性

機械学習の訓練段階に用いる訓練用プログラムや、実行時に使われる予測・推論プログラムが、与えられたデータや訓練済み機械学習モデルなどに対してソフトウェアプログラム

として正しく動作することを、「プログラムの信頼性」（reliability of underlying software system）とする。アルゴリズムとしての正しさの他、メモリリソース制約や時間制約の充足、ソフトウェアセキュリティなど一般的なソフトウェアとしての品質要求がここに包含される。

1.7.9 E-1: 運用時品質の維持性

運用開始時点で充足されていた内部品質が、運用期間中を通じて維持されることを「運用時品質の維持性」（maintainability of quality during operation）と称する。AI システム外部の環境変化に十分に追従できること、AI への入力に関わるシステム自体の状態変化に十分に追従できることと、その追従のための訓練済み機械学習モデルなどの変更が品質の不用意な劣化を引き起こさないことの 3 点を含む。

具体的な実現方法は運用の形態、特に追加学習・再訓練の行われ方に強く依存する。これについては、6.9.1 節で述べる。

1.8 開発プロセスについての考え方

1.8.1 反復訓練による開発と品質管理ライフサイクルの関係

機械学習を利用したシステムにおいては、いわゆる従来の V 字型プロセスの開発のみでなく、Proof of Concept（PoC）と呼ばれる予備的実験段階の導入やアジャイル開発など、より柔軟で適応的な開発プロセスの適用が広く行われている。また、運用段階で得られる実データを元にして、運用中により高い品質の実現や環境変化への追従を行う継続的開発も、機械学習では効果的と考えられる。

このような背景から本ガイドラインでは、実際にソフトウェアを構築する上流〜下流工程だけではなく、システムの仕様策定の段階から、実際の運用段階までを含む一貫した工程を、開発・運用一体型のシステムライフサイクルプロセスとして位置づける。特に、システム設計前の要件分析段階における品質目標の把握と、運用中の動作監視や追加データの取得・追加学習などの工程を、システム開発の重要な一工程として位置づけ、品質管理の取り組み対象の一部として扱う。

その上で、ライフサイクル全体の構成については、品質検査（テスト）結果により進捗を

管理するアジャイル型（反復型）の開発段階と、工程管理に注力するトップダウンの V 字開発の全体工程を複合した、ハイブリッドなプロセスとして全体をモデル化する（図 6）。

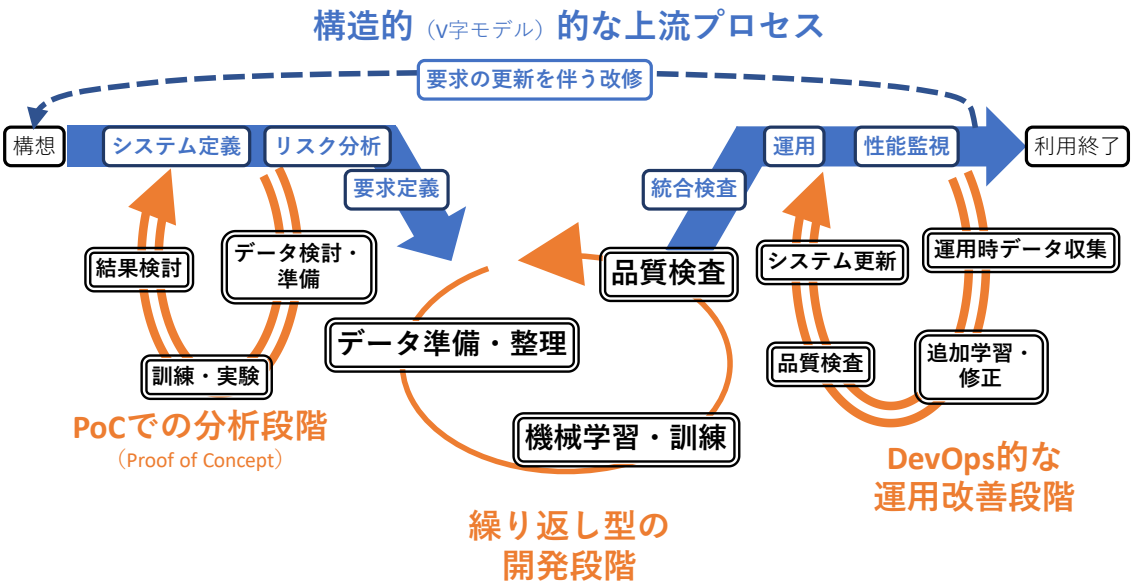

図 6: 混合型機械学習ライフサイクルプロセスの概念図

具体的には、システム全体の品質の必要要件の分析作業については、整理としてはトップダウン型プロセスに類似した流れ式の工程として捉え、実際のシステム開発やデータ整理などに入る前に、独立した工程としてきっちりと分析を行う事を想定する。この段階での手戻りや反復作業については、最終的な結果として開発工程途中から改めてやり直したのとの同等の一貫性を確保することを前提とする。また、運用時のデータ追加取得やモニタリングなどの工程についても、DevOps の考え方に基づき、繰り返し型の品質管理プロセスの一部として位置づけ、必要な場合には RAMS 規格などでのトップダウン型の運用フェーズの考え方とも対応付けできるようにする。

さらに、AI 開発でしばしば行われるいわゆる PoC 開発の段階については、品質管理上の重要な段階として、要求定義に至るまでの事前分析段階のプロセスの一部として整理する。これは、PoC で得られた知見やデータを活用する際に、改めて品質に関する要件を洗い出して再整理することにして、本格的な開発段階における品質マネジメントと、PoC 段階での自由な探索的開発を両立させる考え方である。もちろん、実際の開発工程においては、PoC 段階から本格的な開発段階での品質マネジメントの考え方を部分的に導入することで、再整理の手間を減らすようなやりかたも考えてよい。

なお、本節で掲げたライフサイクルプロセスはあくまで「モデル」であり、各開発者が実

際に行う開発プロセスをここで 1 つに集約するものではない。このモデルは、各々の事情に合わせて工夫される開発プロセスの個々の工程を、本ガイドラインと「対応づけて」理解し、本ガイドラインの各章で書かれる品質管理技術などが、各々の実践する開発工程においてどの段階に対応するかを探す手がかりとなることを意図している。

（参考）従来、AI 開発プロセスは、非ウォーターフォール型の反復型プロセスとして整理されることが多い。反復訓練を行ううちにおいては、アルゴリズム実装のコード変更以外に、収集データの追加、選別の変更、パラメータ調整などさまざまな作業を行い、時にはその変更内容をもとに実装方針や仕様を修正し、新しい仕様に従って訓練を行い、精度を向上させることで目標達成を目指すことも広く行われる。一方で品質を要求されるプロダクトにおいて、現実には、開発プロセスモデルとしては循環型を採用していても、運用前の合否判定や、受発注の検収作業などの形で「品質の関門」を設けることが多い。

図 7 に示すモデルは、このような非ウォーターフォール型の循環型プロセスとして理解される AI 開発プロセスをもとに、本ガイドラインの基礎となる図 6 の混合型のプロセスとの対応関係を例示するものである。

具体的には、実装方針（実装仕様と呼ぶことも有り得る）が固まるまでの複数回の開発試行を、品質を検討する PoC 段階の複数回の循環と対応付ける。その上で、最終的な仕様に基づいて訓練・品質検査を行いリリースに至る、運用段階の直前の 1 回ないし数回の試行を、本ガイドラインの混合プロセスにおける「本開発段階」と位置づける。

時には、本開発段階として開発を行った結果、さらに仕様の修正が必要となることも十分有り得る。ウォーターフォール型のプロセスとして捉えれば、実装段階から要求分析段階への手戻りに相当するが、反復的なプロセスとして捉える場合には、「結果的にまだ PoC 段階であった」というだけであり、深刻な手戻りと捉える必要は無い。PoC 開発段階は実装の先読み（Implementation Forecast）の要素があり、この段階で得られた知見は暗黙に本開発段階に反映されるため、ウォーターフォール型として捉えた場合であっても、PoC 開発段階での実装の労力が無駄になることはない。

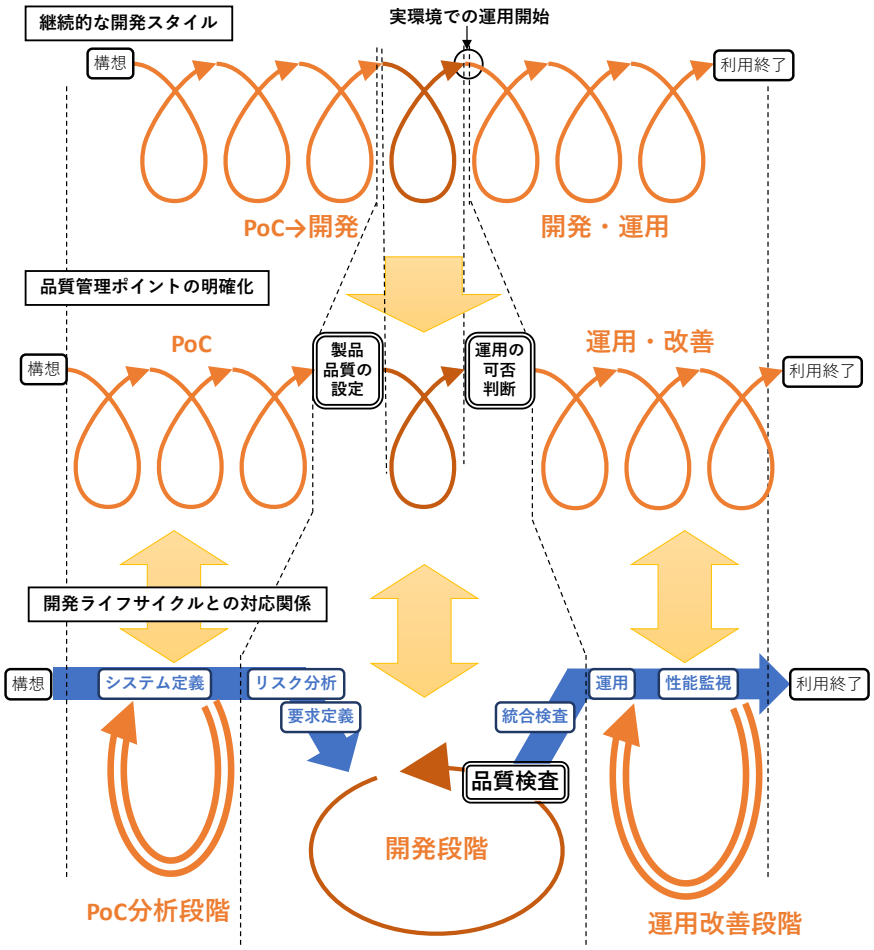

図 7: 循環型のプロセスモデリングと本ガイドラインのモデルの対応（例示）

（参考 2）応用事例として、運用状況の変化や多段の「関門」を有する運用の考え方を 4.1.1 節（52 ページ）に掲載している。

1.8.2 分業による開発と開発プロセスとの関係

実際の AI 開発においては、サービス提供者が自ら企画・開発・運用を一手に行う場合もあれば、訓練段階のみを外注する、システム設計から訓練モデルの構築・システムへの組み込みまでを委託するなど、様々な形で受発注関係などでの業務分担が考えられる。本ガイドラインで定義する品質管理プロセスのうえでは、製品企画を行う開発依頼者を中心とし開発者などを全て含んだ一体の品質管理活動の中で、プロセスの個々の段階の管理作業を作業者間の合意の元に分担する形として整理する。結果として、応用システムの目的に基づく利用時品質や外部品質の設定については、その分担の形態により、開発依頼者側の担当にな

ったり、開発協力者側の担当になったりすることが有り得る[5]。いずれの場合も、最終的な利用者にとって十分な品質を共同で確保することと、そのためのコスト負担を明確に契約などに反映し、運用時まで続く品質管理プロセスの活動が破綻しないように合意しておくことが重要と考えられる。

委託開発や差分開発など、複数の主体が開発に関与する場合の分担モデルの在り方や、その際の留意点については 5.2 節および 5.3 節で述べる。

1.9 他の文書・規範類との関係について

1.9.1 「人間中心の AI 社会原則」

日本においては、内閣府が「人間中心の AI 社会原則」[20] として、主に機械学習利用システムの運用者や開発者と、社会及び一般市民との間のあるべき関係についてまとめている。同原則との関係では、本ガイドラインは一義的には、このような原則を運用者や開発者が十分に理解した上で、実際に機械学習利用システムやその内部の機械学習要素を開発する際に、その品質を作り込み、開発者の意図しない形で「安全・安心」などを損なわないようにするために、開発者が自ら参照して実践する技術的な事項を整理する、非拘束的なガイダンス類として位置づけられる（図 8）。また、本ガイドラインは、他のガイドライン類や将来の国際規格類などと合わせて、同原則中で言及されている「人間中心の AI 開発原則」を構成する要素とも位置づけられる。

[5] 特に、AI 開発にしばしば見受けられる発注要件の形態として、「開発依頼者より与えられたデータ上で一定の性能指標（Accuracy など）を達成すること」を検収要件とした場合には、本ガイドラインの 1.7.1 から 1.7.4 に掲げた「そのデータが製品の目的に見合う学習に十分適する」こと、いわゆる「データ品質」の担保は、開発依頼者が予め完了させておき責任を持つことになることに留意する必要がある。開発依頼者側でデータ品質の検証・担保ができない場合には、その作業の一部を「設計支援作業」などとして明示的に委託することが必要と考えられるほか、実際に訓練を行った際にデータ品質の不足によりシステム構築が成功しない（開発依頼者側まで手戻りが生じる）などの可能性も考慮しておく必要がある。

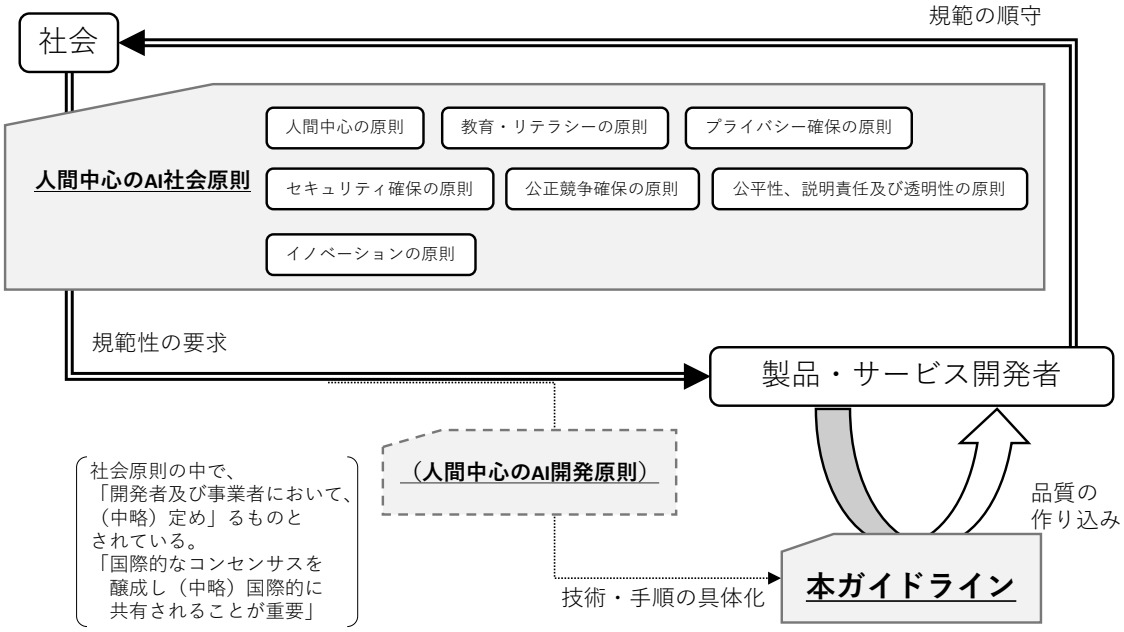

図 8: 「人間中心の AI 社会原則」との関係

1.9.2 人工知能技術に関する海外・国際機関の規範・ガイドライン類

人工知能技術の開発や利用については、上記の原則の他にも近年、その社会性や安全・倫理などに関する規範が、様々な形で文書化されている（OECD [21] や EU [26] など)。本ガイドラインは、これらの文書類との関係においては、前節の「人間中心の AI 社会原則」と同様、これらの社会規範を言語化した文書類の下位に位置づけ、これらの規範の一部をシステム開発において実現するための具体的方法論の 1 つを提示するものとして整理する。

1.10 本ガイドラインの構成

本ガイドラインの残りの部分は、以下の構成となっている。

- 2 章では、本ガイドラインのスコープや既存規格類との関係について改めて整理する。
- 3 章では、1.5 節で述べた利用時品質について、レベル分けの決定を含む詳細について整理する。

- 4 章では、1.8 節で概要を説明した開発プロセスについて、その参照モデルを示す。
- 5 章では、本ガイドラインの具体的な運用・適用方法について述べる。
- 6 章では、1.7 節で述べた内部品質特性をより詳細に整理する。
- 7 章では、6 章の内部品質特性毎に、その留意点や具体的な実現方法の可能性を整理する。
- 8 章では、公平性の外部品質特性のマネジメントに特有の事柄について整理する。
- 9 章では、機械学習品質マネジメントにおけるセキュリティの考え方について整理する。
- 10 章では、他のガイドラインなどとの関係性などの参考情報を示す。
- 11 章では、本ガイドラインの検討委員会が 1.7 節（および 6 章）で掲げた内部品質を洗い出すまでの分析過程および考え方を、参考情報として示す。

（参考）

本ガイドラインが想定する（先頭から順番に読む他の）読み方は、以下のようなものである。

- 5 章でプロセスの手順の概略を理解する。
- 3 章を読み、開発プロセスで参照する品質レベルを決定する。
- 6 章の各節（と 12.1 節の表）を参照し、各レベルで必要なチェック項目を洗い出す。
- 必要に応じて 7 章の各節を参照し、上記のチェック項目の具体的な考え方や適用可能な技術の参考とする。
- 各節で参照されている開発段階などの定義は 4 章にまとめられている。

2. 基本的事項

2.1 ガイドラインのスコープ

2.1.1 対象とする製品・システム

本ガイドラインが品質マネジメントの対象とする製品・サービスは、産業製品・消費者機器・情報サービスなど情報処理を行うシステム全般のうちで、内包される一部のソフトウェア要素の構築に機械学習技術を用いたもの（以下、本ガイドラインにおいて「機械学習利用システム」という。）とする。

なお、本ガイドラインでは、機械学習要素の作り方として主に「教師あり学習」(supervised learning)による実装を想定している。その他の実装方法、例えば教師なし学習(unsupervised learning)、半教師あり学習 (semi-supervised learning)、強化学習 (reinforcement learning) などにより構築されるシステムのうちで、テスト用データセットとして正解ラベル付きのデータを部分的には用意できる応用（例えば、強化学習を用いた機械操作の最適化のうち、行ってはいけない操作は明確に考えられるもの）については、このガイドラインの考え方を応用することができるが、これらの具体的な扱い方については、今後の改訂時に関連する内容を追加する。

一方で、強化学習などの応用のうち、構築者が正解を提示できないような種類の問題（例えば、構築者より強いゲームの人工知能）については、機能要件の考え方を含めて異なるアプローチが必要と考えられる。このような応用についての品質マネジメントについては、現版のガイドラインの対象外であり、今後事例の分析を踏まえて検討する。

2.1.2 品質マネジメントの対象

本ガイドラインが目標を設定し管理・保証しようとする品質特性は、機械学習利用システムを構成する内部要素である、機械学習技術を用いて構築されたソフトウェア要素（以下、「機械学習要素」という。）が、これを用いるシステムに対して及ぼす影響に対応した外部品質とする。

一方で、本ガイドラインにおける品質マネジメントの直接の対象は、機械学習要素がもつ

特性としての内部品質とする。

システム全体の利用時品質は、そのシステムを構成する要素部品の外部品質が総合的に実現するものとして、また各要素の外部品質はその内部の部分要素の品質と、その要素自身の内部的に持つ品質である内部品質特性に依存するものと考える。機械学習要素に内部品質特性として求められる要件を整理し、その要件の達成にいたる工程を管理することで品質管理を行う（図 9)。ただし、システムの構成要素のうち、機械学習要素以外のソフトウェア・ハードウェアなどについては、最低限本ガイドラインの目的に必要な範囲で関係性などを述べるに留める。

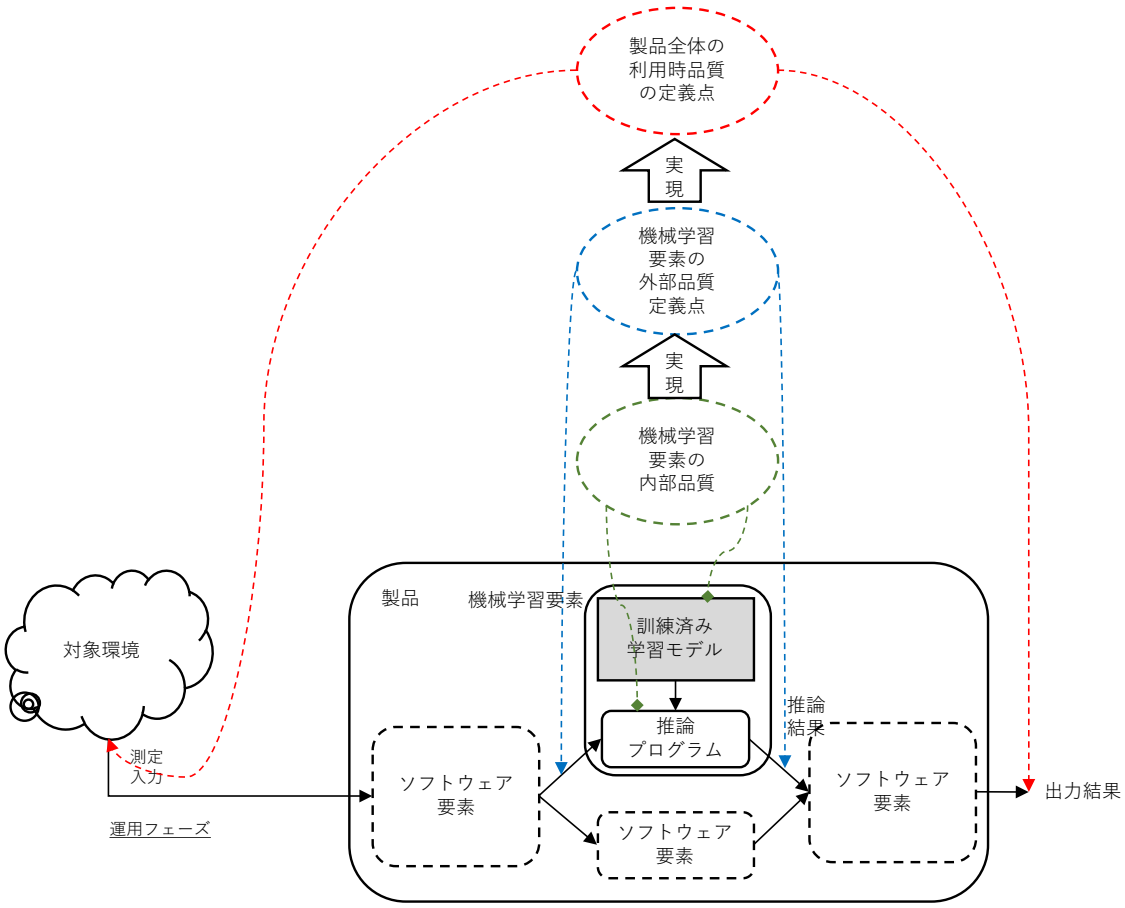

図 9: AI 利用システム全体の品質要求と達成手段の概念

2.1.3 品質マネジメントの範囲

本ガイドラインは「品質マネジメント」を、機械学習利用システムの企画段階から運用までのライフサイクル全体に渡る品質活動プロセスを指す用語として、品質目標の設定、計画、確認、品質保証、管理を包含する意味で用いる。これは、一般のソフトウェアの品質保証や管理よりは広範であるが、例えば ISO 9001 における品質マネジメントに含まれる組織計画や責任・資源などの管理などは含まない。

2.2 システムの品質に関する他の規格などとの関係

本節では、主に情報システムの品質に関する既存の国際規格との関係を整理する。社会性などの上位概念に相当する指針などとの関係については 1.9 節も参考にされたい。

本ガイドラインは、機械学習の様々な応用に対してその品質管理の方法を提示するものであるが、特に安全性を要求されるシステムについては、従来の機能安全性規格の一部（IEC 61508 における第 3・第 4 分冊［13][14］など）を補完・拡張する位置づけとする。

- 機能安全が強く要求されるシステムについては、機能安全規格 IEC 61508-1 [12] または相当する規格を優先して適用する。
- その上で、そのシステム中の機械学習要素について、要求される安全性レベルを担保するために、機械学習固有の安全性確保の課題や手法について整理し、従来のソフトウェアと比較して IEC 61508-3 などの手法を補完・部分的に置換する方法論を提案する。

2.2.1 セキュリティ規格 ISO/IEC 15408

ISO/IEC 15408 [3] などの情報システムのセキュリティ規格と本ガイドラインは独立した関係にあり、必要な場合は同時に適用されるべきである。本ガイドラインが対象とするリスク回避性の実現の為に一貫性や可用性が求められるシステムでは情報セキュリティは必須要件であるが、基本的な対策は機械学習利用システムであっても同規格で対応できると考えられる。

2.2.2　ソフトウェア品質モデル ISO/IEC 25000 シリーズ

ソフトウェアの品質モデルを定義する ISO/IEC 25000 シリーズ、特に ISO/IEC 25010 [7] は部分的に機械学習利用システムに適応しうる。

ただし、特にソフトウェア要素に関する製品品質については、従来ソフトウェアの観点から分析されている点もあり、同規格が整理した品質特性は機械学習要素でも達成されていると考えられるが、本ガイドラインで着目する分析・要素分解とは異なる部分も存在するため、明確な対応関係は無いと考えられる。現在国際的に同規格の機械学習・人工知能向け拡張の提案なども存在するが、今後の標準化なども踏まえて検討を進める。

2.3　用語の定義

本節では本ガイドラインで使用する用語の定義を行う。なお、人工知能及び機械学習に関する用語については、ISO/IEC 22989 [4]（2021 年 6 月時点で DIS 投票中）で議論が行われている。今後、本ガイドラインでの定義の整合を図る予定である。

2.3.1　機械学習システムの構成に関する用語

1. 機械学習利用システム
   machine learning based systems / systems using machine learning technology
   機械学習技術を応用して実装されたソフトウェアコンポーネント（機械学習要素、2.3.1.2）を、コンポーネントとして内包するシステム。
   （参考）一般的な「機械学習システム」(machine learning system）は、その文脈により機械学習利用システムか、機械学習要素（2.3.1.2）に対応すると考えられる。

2. 機械学習要素
   machine learning component
   機械学習技術を応用して実装されたソフトウェアコンポーネント。訓練済み機械学習モデル（2.3.1.5）の機能をソフトウェアとして実現する。通常、ソフトウェア実装としての予測・推論プログラム（2.3.1.6）と、固定的入力として組み込まれる訓練済み機械学習モデル（2.3.1.5）から構成される。

3. 機械学習アルゴリズム

machine learning algorithms

機械学習の予測・推論に用いる計算方法と、その計算方法を訓練により獲得するための計算方法を与えるアルゴリズム。それぞれの計算方法は、予測・推論プログラム（2.3.1.6）と訓練用プログラム（2.3.1.7）に対応する。

ニューラルネットワーク（neural networks）、サポートベクターマシン（support vector machines）、決定木（decision trees）など様々なものがあり、機械学習要素に獲得させる知識の種類や目的により適切なものを選択する。

4. ハイパーパラメータ

hyperparameter

機械学習の訓練を実行するうえで、設定として訓練用プログラム（2.3.1.7）に入力する設定値。訓練の進捗に応じて、調整が必要な可能性がある。ハイパーパラメータが無い場合や、あえてハイパーパラメータ調整を省く場合、またハイパーパラメータの調整を自動で行う技法を用いる場合もある。

（参考）機械学習の対象となりうる数学的構造の多様性について、どの範囲を「機械学習アルゴリズム」として訓練の開始前に固定し、どの範囲を「ハイパーパラメータ」として訓練中に設定・調整するかは、ソフトウェアの実装方法や開発者の訓練方針の選択による任意性がある。場合によっては、計算式としての構造そのものもデータとして扱い、ハイパーパラメータの一部として自動的な最適化調整の対象とすることもある。

5. 訓練済み機械学習モデル

trained model / trained machine learning model / knowledge base / trained parameter

訓練の出力として求められる、機械学習の機能的動作を規定する情報。

（参考 1）本ガイドラインにおいて「機械学習モデル」は、訓練済み機械学習モデルか、それを得るための途中段階のデータを指す。

（参考 2）機械学習技術の文脈で単に「パラメータ」と呼ばれるものは、この訓練済み機械学習モデルを、またはその中に含まれる数値データに着目して指すことが多い。

（参考 3）ニューラルネットワークやベイジアンネットワークなどの数学的構造を「グラフモデル（graphical model）」「統計的モデル（statistical model）」と称することがあり、それに対応して「機械学習モデルの選択」「モデル設計」「未訓練のモデル」などの

用語を、機械学習アルゴリズムの選択及びそのパラメータ設定を指して用いることがある。

（参考 4）trained model の語は、「モデル訓練の出力」として、あるいは具体的なソフトウェアとして実装された訓練済み機械学習モデルとして trained parameter と対比して用いられることがある。本ガイドラインでは（訓練自体の出力と、事前実装された予測・推論プログラムを明確に区別する観点から）前者の意味として用い、後者は「機械学習要素」として整理する。

（参考 5）機械学習あるいは人工知能に関する文脈が明らかな場合には、単に trained model と称しても混乱が無いものと考えられる。

6. 予測・推論プログラム

prediction/inference software component

運用中に用いられる機械学習要素のうち、機械学習モデルのソフトウェア実装として固定されている部分。訓練済み機械学習モデル (2.3.1.5) を静的 (準静的) な入力とし、さらに実環境などから取得されたデータを動的な入力として受け取る。

7. 訓練用プログラム

training software component

訓練用データセットを用いて、訓練済み機械学習モデル (2.3.1.5) を生成するためのプログラム。予測・推論プログラム (2.3.1.6) とは同じ機械学習アルゴリズムの異なる側面からの実装として暗黙に対応していて、通常は組として用いられる。

運用中に用いられる機械学習要素には含まれないことが多いが、運用中にオンライン学習を行うような系・強化学習などでは、一部として含まれることもある。

（参考）本節で整理する各用語の相互関係を次図に示す。

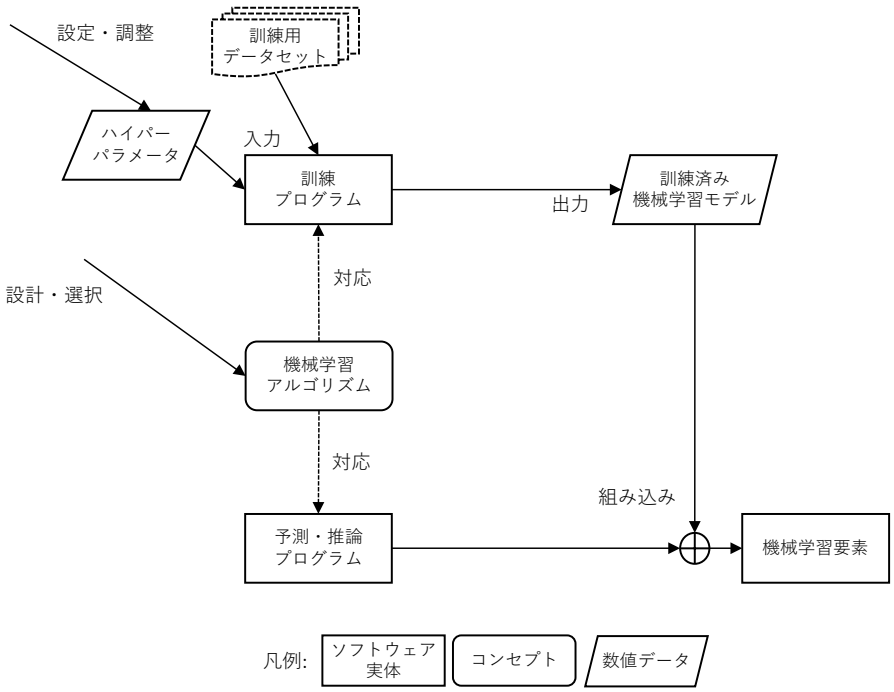

図 10: 機械学習システムの構成に関する用語の相互関係

2.3.2 開発の当事者・ロールに関する用語

1. サービス提供者

   service provider

   機械学習利用システムを自らの目的で、または顧客向けの販売・サービス提供に用いる主体。

   また、ソフトウェア事業者などが、他者が提供・自己使用するサービスを想定したシステムを予め企画・開発しておき、パッケージとして、あるいはカスタマイズを行って販売するような事業形態については、この企画・開発を行う主体をサービス提供者に準じて扱う。

2. 自己開発者

   self-development entity

   機械学習要素の設計・実装を自ら行う製品開発主体。

3. 開発依頼者

development entruster
機械学習要素の実装を他者に依頼する製品開発主体。契約の形態（役務・委託など）を問わない。

4. 開発協力者
development entrustee
機械学習要素の実装を開発依頼者から依頼されて行う者。

5. 開発当事者・ステークホルダー
(development) stakeholders
機械学習利用システムの開発・運用に関わる全ての当事者。製品運用主体・自己開発者・開発依頼者・開発協力者を含む。

2.3.3 品質に関する用語

1. リスク回避性（危害回避性・安全性）
harm avoidance (risk avoidance / safety)
機械学習要素の望ましくない判断動作によって、その製品の運用者・利用者または第三者などに人的被害や経済損失・機会損失などの悪影響を及ぼすことを回避する品質特性である。「リスク回避性」を高めることが、安全分野における「リスク低減」の概念と対応する。
[本ガイドライン 1.5.1 および 3.1 で定義]

2. AI パフォーマンス
AI performance
機械学習要素が、機械学習利用システム及びその利用者が期待する出力を、長期的に平均してより高い精度・確率で出力するという性質。個々の出力の是非よりも、総合的な性能を元に評価する。
[本ガイドライン 1.5.2 および AI パフォーマンス **3.2 で定義]**

3. 公平性

fairness
機械学習要素の出力またはその分布が、入力の元となる人などのもつ属性のうちのいくつかについて、その差異に影響されない（影響が十分低く抑えられる）こと。
（本ガイドライン 1.5.3 および 3.3 を参照）

4. 耐攻撃性
attack resistance
機械学習要素（または機械学習利用システム）が、「攻撃者」によって意図的に構築された入力データまたは外部環境に対して、運用者の期待に反する（「攻撃者」の意図に沿う）反応を示さないことが期待できる性質。

5. 倫理性
ethicalness
機械学習利用システムの動作が、人間を中心とする社会のなかで適切なものであること。

6. 堅牢性
robustness
システムが性能のレベルをどのような状況下でも維持できる性質。

2.3.4 開発プロセスに関する用語

1. システムライフサイクルプロセス
system lifecycle process
システムの企画立案から運用終了廃棄までの一連の流れを俯瞰した工程管理モデル。
[ISO/IEC/IEEE 15288] [2]

2. アジャイル型開発プロセス
agile development process
価値駆動型の俊敏性のある開発アプローチの総称。
（参考）2001 年のアジャイルマニフェストに端を発したもので、スクラムや XP など

様々な手法が活用可能。

3. PoC
   proof of concept
   製品としての実現ではなく、実現したいアイデアや想定する課題解決方法についての実現可能性の検証を目的として行う、予備的な開発活動。

4. システムズエンジニアリング
   systems engineering
   多様な利害関係者のニーズに対応するバランスの取れたシステムソリューションを展開するための複数の分野にまたがるアプローチ。マネジメントプロセスと技術的プロセスの両者を適用し、これらのバランスをとってプロジェクトの成功に影響するリスクを低減する。

5. RAMS
   Specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS)
   本ガイドラインでは主に、鉄道分野の信頼性管理プロセスについて定めた IEC 62278 [15]（EN 50126）規格に定められた総合的なシステムライフサイクルプロセスの概念を指すものとして用いる。
   （参考）「RAMS 規格」としては、一般に IEC 62278 自体のことを指すこともある。

2.3.5 利用環境に関する用語

1. 環境
   environment
   本ガイドラインでは 3 つの文脈で用いる: 1）外部環境 / 2）計算機環境 / 3）運用環境

2. 外部環境
   external environment
   機械学習利用システムに対する入力の源となるシステム外部の実環境（physical environment）または情報空間（cyberspace）であって、その環境からシステムへの干

渉またはシステムから環境への干渉があるもの。

3. 開放環境
   open environment
   外部環境のうち特に利用者以外の人・自然などのモノの状況がそのシステムの動作に大きく影響を及ぼすようなもの。

4. 環境条件
   environmental condition
   外部環境の持つ状況の違いを識別するような特徴。機械学習品質マネジメントの観点からは、特に危害の状況やリスクなどが変化するような特徴に着目する。

5. 計算機環境
   computing environment
   機械学習要素（推論・予測プログラム）や訓練用プログラムなどを実行するソフトウェア実行環境。状況により、その基盤となるハードウェア環境や、オペレーティングシステム・ミドルウェアなどを含み得る。

6. 運用環境
   operational environment
   計算機環境と、それを用いる（人間の）運用体制などを含む。

7. 実運用環境
   in-operation environment
   機械学習利用システムが実際に実用に供される運用環境。

8. 推論実行環境
   runtime (computing) environment
   実運用環境中で、機械学習要素を含む機械学習利用システムを実際に運用する計算機・クラウド・IoT デバイスなどの計算機環境。

9. 開発環境

development environment
機械学習要素を構築・修正する計算機環境で、その環境における修正が実運用に直接影響を及ぼさないもの。また、その計算機環境を含む運用環境。

2.3.6　機械学習構築に用いるデータなどに関係する用語

1. 訓練用データセット
   training dataset
   反復訓練フェーズにおいて、機械学習モデルの訓練に用いるデータの集合。

2. 訓練用データ
   training data (instance)
   訓練用データセットに含まれるデータ。
   （参考）一般的には、「データ」は単数のインスタンスの意味でも、複数のインスタンス（インスタンスの集合）の意味でもどちらでも用いられる。本ガイドラインでは、この多義性を避けるため、「データ」は前者の意味にのみ用い、後者の意味には「データセット」を用いる。すなわち、「データセット」は、集合としての分布や総量などを議論対象とできるような、1 つの塊として扱う場合に用いる。「データ」は集合として捉える前の個々の、あるいは散在的なデータ点を表す場合に用いる。これらの間の関係は、「データセットに加える」「データセットに含まれる」など、集合と要素の関係の用語で記述する。

3. バリデーション用データセット
   validation dataset
   反復訓練フェーズにおいて、機械学習モデルの収束性などを評価するために用いるデータの集合。

4. テスト用データセット
   test dataset
   品質確認・検証フェーズにおいて、構築された機械学習モデルが所用の品質・性能を満たすことを確認する手段（の 1 つ）としてのテストに入力として用いるデータの集合。

5. 敵対的データ
   adversarial example(s)
   機械学習要素に入力すると想定・直感と異なる推論結果が出力されるよう、意図的に構成されたデータ。

6. 過学習
   overfitting
   訓練済み機械学習モデルが訓練データに過剰に適合し、訓練データ以外の入力データに対し望ましい出力を返せなくなること。

7. 属性
   attribute
   ML 要件分析において、環境条件の特徴を分析・分類する項目立ての各要素。

8. 属性値
   value
   ML 要件分析において分類した各属性に含まれる具体的な環境の特徴の種別。

9. 正解ラベル
   labels
   教師あり学習において、分類問題に用いられるデータに付随し、属する正しいクラスを示す識別子。

2.3.7 その他の用語

1. リグレッション
   (software) regression
   （参考）ソフトウェア工学分野で、ソフトウェアを改良した際、以前は期待通りに動作していた入力に対して、期待通りの動作をしなくなること。
   （注）機械学習・統計分析の分野では回帰分析の意味で用いられる。混乱を招くため、基本的には本ガイドラインでは使用しない。

2. SIL

   safety integrity level (SIL)

   IEC 61508 で定められる、製品の機能安全性の達成に関するレベル分け。

   [定義元: IEC 61508-1]

3. KPI

   key performance indicator

   機械学習要素の出力が機械学習利用システムを通じて達成する機能要件の目標達成度合いを、数値化して定量的に示す指標。

4. 継続的学習

   continuous learning

   運用期間中にデータの収集と追加的な機械学習の訓練を行い、随時・適時に訓練済み機械学習モデルの更新を行う運用の形態。

   （参考）下記の「オンライン学習」だけでなく、「オフラインでの追加学習」などを含む概念である。

5. オンライン学習

   online learning

   実運用環境において、開発環境での検証プロセスを経ることなく継続的学習とその結果の反映が行われるシステム実装の形態。

   （参考）オンライン学習を行わず、開発環境で追加学習を行い、検証プロセスを行ってから、ファームウェアアップデートなどの形態でモデルパラメータの更新を行う形態を、本ガイドライン中では「オフラインでの追加学習」と称している。

3. 機械学習利用システムの外部品質特性レベルの設定

本ガイドラインでは 1.5 節で述べたとおり、機械学習利用システムに内包される機械学習要素に対し 3 つの外部品質の特性軸を定義し、それぞれ下記の基準により品質レベルを設定する。開発における具体的な設定手順については、5.1.2 節で規定する。

3.1 リスク回避性

機械学習利用システムのリスク回避性のレベルについては、人に対する傷害などの人的リスクと、その他の経済的リスクに細分し、それぞれ表 1 及び表 2 により 7 レベルの「AI 安全性レベル」（AISL 4, AISL 3, AISL 2, AISL 1, AISL 0.2, AISL 0.1, AISL 0）に分類する [6]。

但し、表中の※に該当する項については、先に機能安全性規格 IEC 61508-1 [12]（または機能安全性に関する各応用分野毎の個別国際規格）を適用し、当該装置が IEC 61508-1 の SIL4～1（または同等と考えられる個別規格の安全性レベル）に相当する場合、そのレベルを同等の数値の AISL に読み替える。また、※に該当しない場合についても、IEC 61508-1 などを適用する場合には、同規格の SIL を AISL に読み替えて良いものとする。また、適用アプリケーションによって、発生頻度に応じて対応する SIL 等のレベルが異なるものについては、リスクアセスメントに基づいて適切なレベルを設定してもよい。

なお当面の間、AISL 4 および 3 に相当する極めて高いレベルの信頼性が機械学習要素に（直接的に）要求される場合については、現時点ではシステム全体の設計を見直すことなどにより対処するものとし、本ガイドラインでは対策基準を設定せず将来の検討課題とする。

表 1: 人的リスクに対する AI 安全性レベルの推定基準

想定される影響 ＼ 回避可能性	回避操作が不可能	人の監視により回避操作が可能	人による確認・手動操作が常に必要
複数人の同時死亡	※	※	※

[6] AISL の各レベルの表記については、IEC 61508（機能安全）規格の安全性インテグリティレベル SIL 4～SIL 1 と概ね対応させつつ、従来「SIL なし」とされているレベルをさらに 3 段階に分割するために、大小関係を保つために小数を用いて 0.2、0.1、0 というレベル表記を採用した。

単一の人の死傷	※	※	※
障碍の残る傷害	※	AISL 2	AISL 1
重傷	※	AISL 1	AISL 0.2
軽傷	AISL 1	AISL 0.2	AISL 0.1
軽傷(想定される被害者により容易に回避できる場合)	AISL 0.2	AISL 0.2	AISL 0.1
損害の想定無し	AISL 0	AISL 0	AISL 0

表 2: 経済リスクに対する AI 安全性レベルの推定基準

影響　＼ 回避可能性	回避操作が不可能	人の監視により 回避操作が可能	人による確認・ 手動操作が 常に必要
企業体としての存続等に著しい影響	(AISL 4)	(AISL 3)	AISL 2
業務の運営を揺るがす重大な損害	(AISL 3)	AISL 2	AISL 1
無視できない・ 具体的な損害	AISL 2	AISL 1	AISL 0.2
軽微な利益の逸失にとどまる	AISL 1	AISL 0.2	AISL 0.1
損害の想定無し	AISL 0.1	AISL 0	AISL 0

（参考）
AISL 0.1, 0.2, 1, 2, 3, 4 は概ね、IEC 62998 [16] の Sensor Performance Class の A～F（それぞれ AISL 0.1, 0.2, 1, 2, 3, 4 に対応）や、ISO 13849 [1]の Performance Level の PL_a～PL_e（それぞれ AISL 0.1, 0.2, 1, 2, 3 に対応）と対応していると考えられる。

3.2　AI パフォーマンス

AI パフォーマンスについては、以下の標準とし、サービス提供者が、または開発ステークホルダ間の協議により決定する。

- AIPL 2 (mandatory requirements):
  - 当該製品・サービスが一定の性能指標（正答率・適合率・再現率など）を満たすことが、製品・サービスの運用上の必須または強い前提である場合。
  - 受発注などの契約において、前記の一定の性能指標の充足が受入要件として明確に記載される場合。
- AIPL 1 (best-effort requirements):
  - 一定の性能指標が製品・サービスの目的として特定されているが、AIPL 2 に該当しない場合。

    特に、リリースまでの日程スケジュールが重視される場合、または品質をモニタリングしながら試験運用を行い漸次性能向上を行う運用が許される場合。
- AIPL 0
  - 性能指標が開発時点で特定されず、性能指標そのものを発見することが開発の目的となる場合など。
  - 所謂 PoC の段階で終了する開発を行う場合。

3.3　公平性

公平性については、以下を基準とする。

- AIFL 2 (mandatory requirements)
  - 法令・規則・社会的なガイドラインなどにより、一定の公平な取扱いを行う事があらかじめ要求・強く想定されている場合。
  - 当該製品・サービスがパーソナルデータを扱い、その出力が、個人の権利などに直接影響を及ぼす場合。

- AIFL 1 (best-effort requirements)
  - 当該製品・サービスが偏りを持たないことについて、明確な要件が特定できる場合。
  - 当該製品・サービスが内包する機械学習要素（ないし AI）の出力が公平であることを説明できないことが、システムの社会受容性などに影響する場合・運用の障害になる場合。

- AIFL 0
  - 当該製品・サービスに対する公平性の要求が存在しない場合。
  - 当該製品・サービスが潜在的に不公平・不均一であったとしても、性能とのトレードオフなどの観点から積極的に肯定されうる応用である場合。
  - 公平性の品質要求が、安全性などの面で「常に性能が高いこと」を求めるような場合。このような場合は、リスク回避性（またはパフォーマンス）の品質における、入力多様性への対応の要求と考える。
    （例えば、機械的損傷の検知のような応用において、特定の損傷の検知率が高くなった場合に、他の損傷に合わせて検知率を下げるべきでは無い。）

4. 機械学習利用システムの開発プロセス参照モデル

本章では、本ガイドラインが議論の前提として参照する機械学習利用システムの開発プロセス参照モデルについて述べる。

本章の参照モデルは、機械学習利用システムについて特定の開発方法を開発者に強制するものではなく、一般的な機械学習利用システムの内部構成要素や開発工程に参照可能な名称を付加し、それぞれの固有の構成や開発プロセスを本モデルと対比することにより、本ガイドラインが規定する推奨事項などを実際の開発工程と対応づける為のものである。

本ガイドラインは図 6（28 ページ）に示したとおり、機械学習要素（およびそれを含む機械学習利用システムの主要な部分）の開発工程を大きく 3 段階に分析する。

4.1 PoC 試行フェーズ

開発ライフサイクルの工程としての「**PoC 試行フェーズ**」は、システム開発の最初期段階において、システム全体の機能定義などに先行してシステムの達成できる性能の可能性や、品質劣化リスクやその原因となる状況などを特定するための準備段階として整理する。本ガイドラインの品質マネジメント上、PoC 試行フェーズにおける品質マネジメント活動は、本格開発フェーズにおける品質マネジメント活動のための重要な準備作業と整理する。この段階におけるデータサイエンスの観点からの分析活動は、後のリスク分析や要求分析と密接に関連し、その分析結果が後の本格開発フェーズにおいて利用されることも多い。特に、「要求分析の十分性」（6.1 節）を達成するためには、PoC 試行フェーズの取り組みが欠かせない。本ガイドラインでは、これらの活動の妥当性を最終的に本格開発フェーズの各段階で改めて確認することと整理する。これにより、PoC フェーズでは品質マネジメント活動そのものについても試行錯誤を制約無く自由に行うことができることになる。

4.1.1 試験運用を含む PoC フェーズなどの取扱い

また、一般には、単なるデータ収集や分析・試行的な訓練・モデル構築に留まらず、最終製品とは異なる利用状況（例えば有人の監視下）ではあるものの、実運用の環境で試行するようなものもひとくくりに「Proof of Concept」と呼ばれることがある。また、運用の段階

が複数にわかれ、品質要求が異なる運用状況へ連続的に移行していく場合もある。このような複数段階の実運用を伴う開発については本ガイドラインでは、それぞれの段階のシステムの構築段階に対して各々、本ガイドラインにおける「本番開発フェーズ」を含む全体の開発ライフサイクルに準ずるものとし、かつ早期の運用段階の成果すべてを後期の開発における PoC 段階の知見として取り扱う（図 11）。

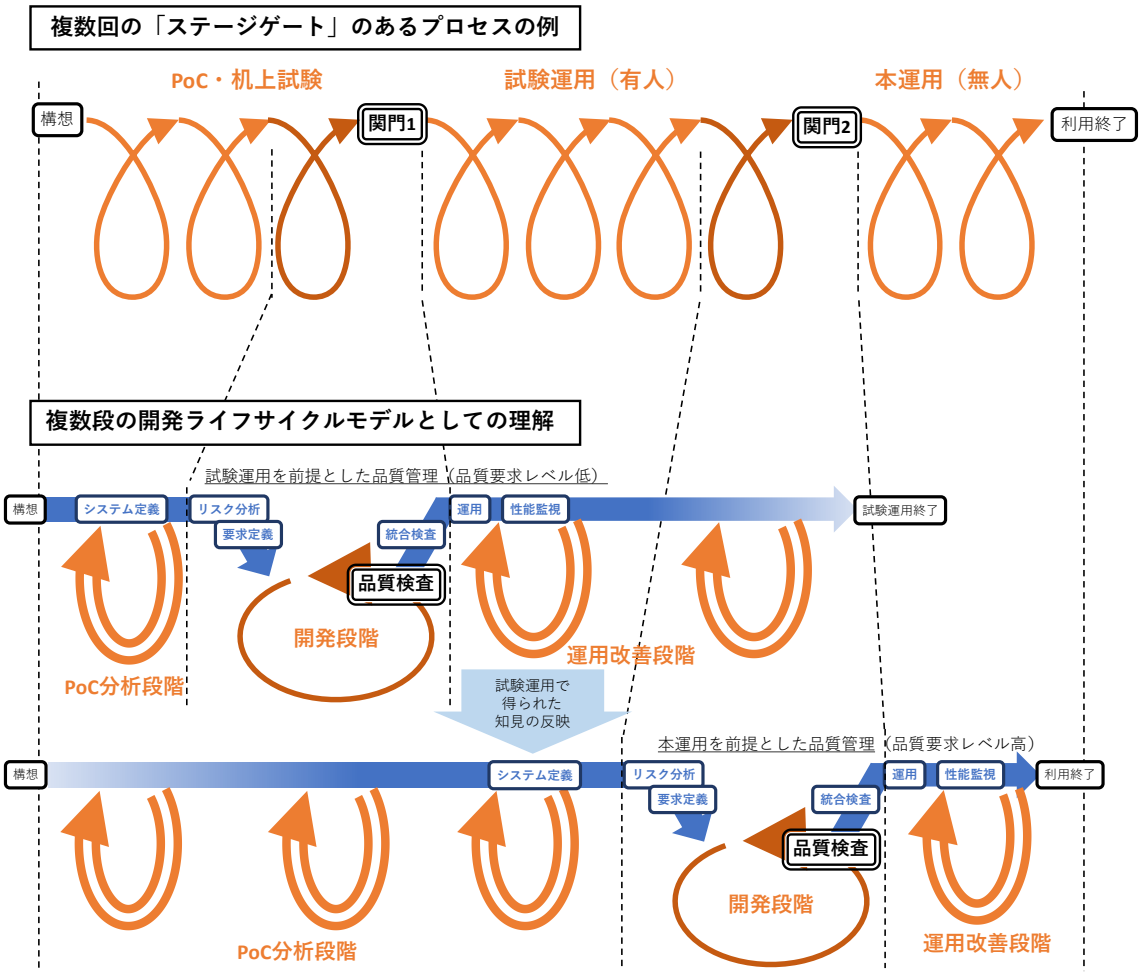

図 11: 複数の運用段階を伴う開発プロセスの取扱い（例示）

4.2 本格開発フェーズ

PoC 試行フェーズにおける取り組みによりシステムの機能要求が確定して以降、実運用に至る直前までのライフサイクル上の工程を、「**本格開発フェーズ**」とする。このフェーズには、従来のシステム開発の超上流工程に相当するシステム定義から、最終的な運用・出荷前の統合検査までの工程を含む。

本格開発フェーズにおいては、従来のウォーターフォール型の上流開発工程の一部と、機械学習要素に特有な反復型の開発工程を混合したライフサイクルをモデルとして定義する。より具体的には、従来「上流工程」とされてきた、システム全体の機能定義やリスク分析と構成要素の影響分析・分解、機械学習要素を含む各構成要素のシステム要求を決定し、出荷前の最終の統合検査におけるゴールを設定する一連のプロセスは、従来のウォーターフォール型モデルに準じて整理する。また、機械学習要素以外のソフトウェア・ハードウェアの構成要素については、下流工程についても従来のウォーターフォール型開発モデルや、同等の品質を確保できるアジャイル開発プロセス[7]を適宜適用するものと想定する。一方で、機械学習要素についての「下流工程」に当たる具体的な機械学習モデル開発のプロセスは、次節で改めて反復型の開発プロセスモデルを定義する。

4.2.1　機械学習モデル構築フェーズ

本格開発フェーズ中で、機械学習要素が対応すべきリスクや品質要件が特定された後、機械学習モデルを構築しその外部品質を（内部品質の指標を用いて）検査し、合格するまで反復する段階を、**「機械学習モデル構築フェーズ」**とする。

このフェーズをより具体的な工程に分解したモデルを図 12 に示す。このモデルの各工程も、各開発者それぞれ固有の開発プロセスを本モデルと対比することにより、本ガイドラインの記述と対応づける為のものであり、開発プロセスを一義的に制約する目的ではない。

[7] ここでいう「アジャイル開発プロセス」は、テスト主導型開発など、ウォーターフォール型の品質確保プロセスを代替するような品質確保活動を含み、その期待される品質面の効果をきちんと説明可能であるようなプロセスを想定しており、非ウォーターフォール型の開発の全ての形態を容認する趣旨ではない。

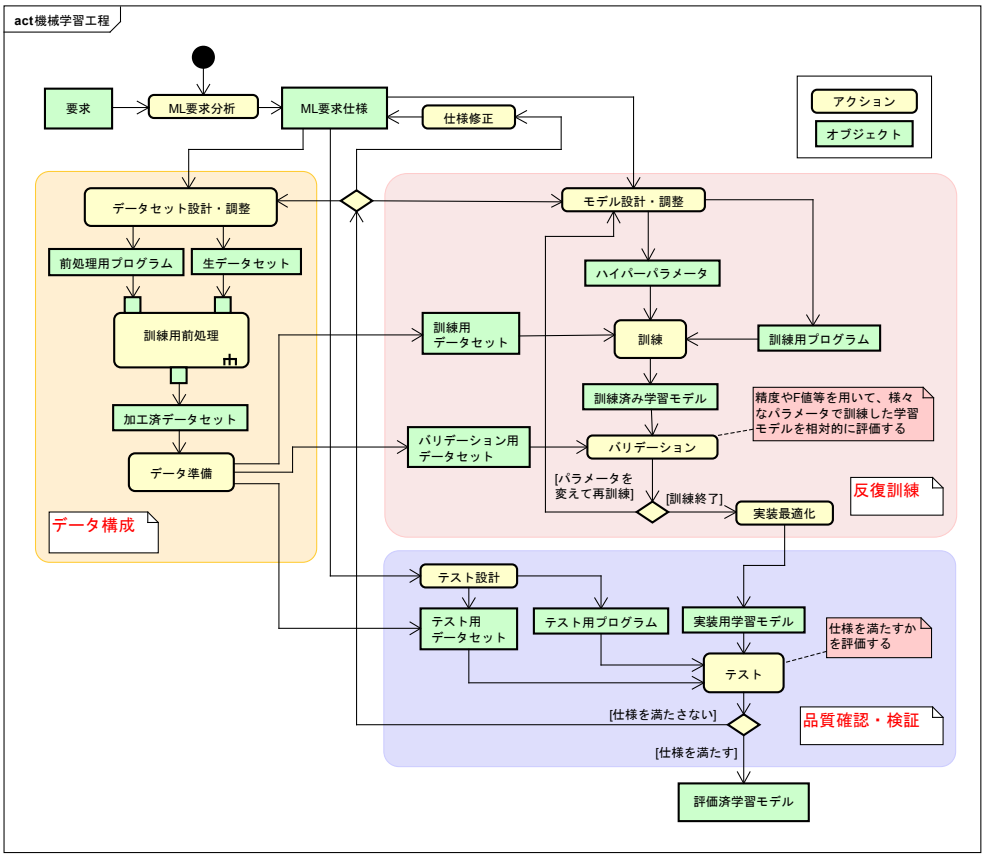

図 12: 機械学習構築の段階におけるプロセスモデル（例示）

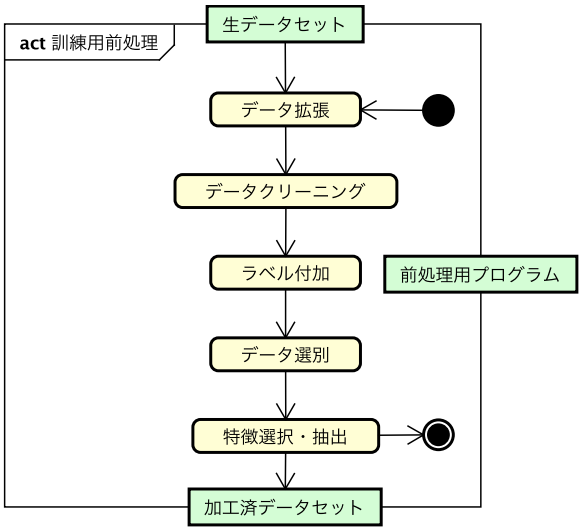

図 13: 訓練用前処理の一例

4.2.1.1　ML 要求分析フェーズ

機械学習要素に対する概念的な品質要求を、特に機械学習要素への入出力データデータの性質の整理として具体化し、機械学習モデル構築フェーズにおける具体的な構築への要求として再整理するフェーズを、「**ML 要求分析フェーズ**」とする。実際の機械学習利用システムの開発においては、しばしばデータの性質や具体的なデータセットそのものに基づいて性能要求（品質要求）が定められることも多いが、このような分析フェーズを明示的に置くことで、データ品質の具体的な管理方法や、他の構成要素との品質管理上の関係などを整理することができる。このフェーズは主に、6.1 節・6.2 節の内部品質に強く関係する。

4.2.1.2　訓練用データ構成フェーズ

訓練用データ構成フェーズでは、ML 要求仕様にしたがって、訓練用、バリデーション用、テスト用のデータセットを生成する。以下、図 12 左上の「訓練用データ構成」の流れにそって説明する。

4.2.1.2.1　データセット設計・調整ステップ

データセット設計・調整ステップでは、ML 要求仕様にしたがって、ケースごとに十分なデータを収集し、訓練に適したデータセットを生成するための設計を行う。このステップのアウトプットは、加工していないデータセットと前処理用プログラムとなる。なお、訓練後に訓練済み機械学習モデルが仕様を満たさない場合は、このステップに戻り、データセットの設計を調整することもある。

4.2.1.2.2　訓練用前処理ステップ

訓練用前処理ステップでは、4.2.1.2.1 節のデータセット設計・調整ステップで生成した生データセットを訓練に適したデータセットに加工する。図 13 に掲げたような、各処理を順不同で行う、並列に行う、繰返し行う場合などもある。

4.2.1.2.2.1　データ選別

大量の生データセットから、データセットの被覆性、均一性を考慮しながら、訓練と評価（バリデーションとテスト）に用いるデータセットを選別する。被覆性と均一性はトレードオフ関係にあるため、そのバランスは ML 要求仕様にしたがう。

4.2.1.2.2.2　データクリーニング

データセットに対して、本来の特徴を訓練できるように、ノイズ除去、データ整形、欠損値補完、外れ値除去などを行う。

4.2.1.2.2.3　データ拡張

訓練用のデータ数の確保や過学習の防止など、または、テスト用のデータ数の確保のため、データの変種を生成してデータセットに追加する。例えば、画像データでは、変種を生成するために、回転、縮小、反転、明度変更、背景の置換えなどの操作を行う。

4.2.1.2.2.4　ラベル付加

データセットに対して、ML 要求仕様にしたがい、正解（教師用）のラベル付けを行う。

4.2.1.2.2.5　特徴選択・抽出

モデルの訓練に有用な新たな特徴量を加えるために特徴抽出を行い、不要な特徴量を減らすために特徴選択を行う。例えば、特徴抽出ではフーリエ変換による周波数成分を計算したり、複数の特徴量の主成分を求めたりなどの技法を用いる。有用な特徴量が得られれば、多くの場合その数以上に不要な特徴量を特徴選択により削減できる。

4.2.1.2.3　データ準備ステップ

データ準備ステップでは、生成した加工済データセットを訓練用、バリデーション用、テスト用に分割する。反復訓練中に、訓練用とバリデーション用のデータセットを入れ替えることもある。

4.2.1.3　反復訓練フェーズ

反復訓練フェーズでは、ML 要求仕様にしたがって機械学習モデルを設計し、訓練用データ構成フェーズで作成したデータセットを用いて訓練とバリデーションを繰り返し行う。以下、図 12 右上の「反復訓練」の流れにそって説明する。

4.2.1.3.1　モデル設計・調整ステップ

モデル設計・調整ステップでは、ML 要求仕様にしたがって訓練に必要なハイパーパラメータ（学習モデルの構成、学習アルゴリズム、各種パラメータなどを含む）を設計するとと

もに、訓練用プログラムを作成する。バリデーションやテストの結果を受けて、ハイパーパラメータを調整することもある。

4.2.1.3.2　訓練ステップ

設計したハイパーパラメータをもとに、訓練用のデータセットを用いて、機械学習モデルを訓練する。

4.2.1.3.3　バリデーションステップ

訓練した機械学習モデルにバリデーション用のデータセットを入力し、学習モデルの評価指標（正解率、適合率、再現率、F 値など）によってモデルを評価する。一般には、複数のハイパーパラメータで複数の機械学習モデルを訓練し、それらを相対的に評価して、その妥当性を判断する。

4.2.1.3.4　後処理ステップ

後処理ステップでは、訓練済み機械学習モデルを実運用環境（推論用サーバやエッジデバイスなど）に合わせるためのモデル変換などを行い、実装用学習モデルを生成する。

具体的に行われる事例としては、

- 実運用環境（計算性能）に適したモデルの変換
  - 計算精度の変更
  - モデルの圧縮・高速化（蒸留、量子化など）
- 実運用環境に適したモデルの最適化（効率的な推論）
  - モデルのコンパイル（並列化、ベクトル化など）

などがある。

本ガイドラインの想定するプロセスモデル図においては、この後処理は反復訓練フェーズの直後に置き、品質確認・検証フェーズは実運用時に近い条件で行われることを想定しているが、実際の開発においては、機械学習要素単体での品質確認・検証フェーズのあとで、システム全体の構築の工程の一部として行われることもしばしばある。この場合、品質確認・検証フェーズで確認した品質が、実運用時に変化する（悪化する）可能性があることに注意し、場合によっては数値的同等性やシステム全体での検査段階などでの再検証が必要となることを想定しておく必要がある。

4.2.1.4 品質確認・検証フェーズ

品質確認・検証フェーズでは、ML 要求仕様にしたがってテストを設計し、実装用学習モデルの評価（テスト）を行う。以下、図 12 右下の「品質確認・検証」の流れにそって説明する。

4.2.1.4.1 テスト設計ステップ

テスト設計ステップでは、ML 要求仕様にしたがって、テストに必要なプログラムの作成やテスト用データの追加を行う。追加するテスト用データとしては、前処理のデータ拡張で行うような操作でデータを生成する場合や、訓練済みの学習モデルが誤推論しやすいデータを生成する場合もある。

4.2.1.4.2 テストステップ

実装用学習モデルに対し、テスト用データセットを用いた通常の指標（正解率、適合率、再現率、F 値など）による正確性の評価の他、データセットの各データ以外のデータ（ノイズを付加したデータなど）を用いた安定性の評価なども行う。評価結果として ML 要求仕様を満たさない場合は前のフェーズに戻り、学習モデルの調整、訓練用データセットの調整、ML 要求仕様の修正などを行う。

4.2.2 システム構築・統合検査フェーズ

本段階は機械学習構築の一部ではないが、手順としての順序の関係上ここに記述する。

理想的な開発状況であれば、機械学習要素に対する品質の要求は PoC 試行フェーズを通じて全て事前に整理され、その達成が機械学習構築フェーズまでで全て確認でき、その後のシステム全体の構築や統合テストの段階では問題が起こらないことが望まれるが、実際のシステム開発においては、複雑な実環境の要求分析が完璧でなかったり、他の構成要素と想定外の相互作用を起こしたり、他の構成要素の不具合の影響が波及するなど様々な要因で、いわゆる統合テストの段階で、機械学習要素の品質上の不具合が発見されることはしばしば起こる。また、特に大規模で複雑なシステムでは、事前のデータだけではどうしてもテスト用データとして不足し、統合後の実環境・実システムでの検査が不可欠である場合も多いほか、データ構成フェーズではどうしても事前に用意しきれないレアケースに対する検査

を、統合テストの段階で再現して行うことなども考えられる。

このような場合は、検査が失敗したときには ML 要求分析に対する部分修正などが起こり、機械学習モデル構築フェーズ全体がもう 1 周行われることになる。このような場合に膨大な作業の再発生を避けるためにも、機械学習構築の各段階においては、それぞれのフェーズでの品質管理活動内容をきちんと記録し、修正の影響などをきちんと把握できるようにしておくことで、部分修正を確実に行えることが望ましい。

4.3 品質監視・運用フェーズ

システムが実運用に展開された後に、品質を運用段階で継続的に監視し必要な修正を加えることで性能を維持し続ける為の活動を、「**品質監視・運用フェーズ**」として明確に位置づける。このようなフェーズは、ウォーターフォール型の V 字開発モデルでは狭義の開発工程の外側に置かれるが、例えば RAMS 規格などのシステムライフサイクルの考え方では既に存在するフェーズである。

特に機械学習利用システムにおいては、

- 事前データに基づく定性的な要求分析が実環境を捉え切れていないケース
- 事前データを用意した際の環境から、運用時の環境が変化しているケース

の双方の観点から、品質の運用時管理が必要な場合が多いと考えられる。

機械学習の多様な応用においては、実際に運用時に訓練済み機械学習モデルを更新する手段として様々なパターンが既に存在しており、単一の類型化では扱いきれないことから、本ガイドラインでは主なパターンとして、

① 開発環境で再訓練し、検査後に運用環境に展開するパターン

② 運用環境で自動的に追加学習し、自己適応するパターン

の 2 類型に分類整理する。その上で、6.9 節でそれぞれのパターンの対応方針を個別に整理することとする。

5. 本ガイドラインの適用方法

5.1 基本的な適用プロセス

本節では、想定する本ガイドライン適用プロセスを概観する。

本節では機械学習要素の構築に焦点を当てているが、説明上必要な範囲において、従来既に行われているであろう、システム全体の分析などの過程も含んでいる。そのため、開発担当者において、国際規格対応のために具体的な開発プロセスが既に構築されていて、その妥当性を説明できる場合には、その既存プロセスを基に本節で掲げる段階の必要なもののみを追加するなど、改変を行っても良い。

5.1.1 機械学習要素のシステム内での担当機能の特定

機械学習要素がシステム内で果たすべき安全性などの機能が、例えば IEC 61508 [12] などの従来型システム開発プロセスにおいて十分に特定できている場合、本項はスキップして良い。

機械学習システム全体の利用時品質が未特定の場合、以下のプロセスを通じてシステム全体の利用時品質と機械学習要素の外部品質を特定する。

5.1.1.1 安全性機能の事前検討・適用規格の確認

- 機械学習利用システムが一定以上（おおむね SIL1 以上・無視できない人身障害が想定される程度）の安全性機能を必要とするかをあらかじめ検討する。
- 安全性機能が必要な場合は、IEC 61508 または各応用分野の規格から適用すべきものを選択し、そのプロセスに従う。

5.1.1.2 システムの機能要件（目的・目標）の特定

- システムの利用目的・想定される利用環境の範囲、および達成すべき KPI を概要レベルで特定する。

5.1.1.3　システム利用のリスクシナリオ検討

- システムの機能要件が達成されない、またはシステムが利用目的や社会の要求に不適合となる状況の可能性を検討し、その場合に起こる被害その他のデメリットを列挙する。いわゆるリスクアセスメントに対応する。

5.1.1.4　システム全体の利用時品質特性の要求の検討

- 何らかのシステムの品質メトリクスを用いて、システムの利用時に要求される品質を検討する。
  - 他に適切な選択肢が無い場合の手段の一例としては、3 章で掲げる 3 つの外部品質特性軸を利用時品質に流用し、前節で検討したデメリットのいずれに当たるかを判断し、それぞれの深刻度を 3 章各節の基準で判断し品質レベルに対応づける。
  - 3 つの軸ごとに、最大のレベルを特定し、システムの達成すべき利用時品質レベルとする。
- 但し、5.1.1.1 で従前の安全性規格を採用することとした場合には、物的損害に対するリスク回避性レベルは従来規格の該当する品質指標（機能安全性など）による。

5.1.1.5　システムの構成要素の設計・機能要件及び利用時品質特性の達成に寄与する要素の特定

- システムの構成要素の組み合わせを設計し、機械要素や従前のソフトウェアなどで達成される機能と、機械学習要素により実現される機能を分別する。
- また、利用時品質特性の達成がシステムのどの構成要素に依存するかを分析する。

5.1.2　機械学習要素の外部品質達成要求レベルの特定

- 機械学習要素が具体的に利用時品質に寄与すべき外部品質の達成レベルを、以下の手順により特定する。
- リスク回避性のうち物的・人的損害リスクについて、従来の機能安全性規格を適用する際は、従来の規格による分析を優先し、同規格による機械学習要素がソフトウ

ェアとして達成すべき機能安全性レベルを、機械学習要素のリスク回避性の達成要求レベルに読み替える。

- その他のケースについては、以下の通りとする。
  - 基本的には、システム全体に要求される利用時品質の特性レベルを、機械学習要素に要求される外部品質の達成要求レベルとする。
  - 機械学習要素と並列・直列に処理されるソフトウェア要素（図 9）により、機械学習要素の望ましくない出力に対して監視・補正（出力の上書き修正）が行われる場合で、かつ同ソフトウェア要素が従前の手法で十分な品質を確保できると評価できる場合、機械学習要素への品質特性レベルへの要求は、システム全体のレベルから 1 段階軽減させたものとする。
  - システム全体に要求される品質特性レベルの達成に、機械学習要素が全く寄与・干渉しないと認められる場合、機械学習要素には当該品質特性の達成レベルを設定しない（レベル 0 とする）。

5.1.3　機械学習要素の内部品質の要求レベルの特定

- 6 章の各節に掲げる内部品質特性の各項目について、それぞれの特性の要求レベルを、同節内に掲げる対応関係に従い、前項の利用時品質の特性達成要求レベルから導出する。

5.1.4　機械学習要素の内部品質の実現

- 前項の内部品質特性の要求レベルを実現する方法を、8.2 節の各小項目に従って検討する。
- 各項に掲げられた技術・プロセスや同等と考えられる技術により、各特性の実現を担保する。

5.2　（参考）AI 開発の依頼

本節の内容は参考（informative） である。

機械学習要素開発に必要な、前節までで述べたフェーズやプロセスを遂行する作業は、単

一の組織では遂行しきらず、作業依頼（受発注）が発生する場合も多い。本節では、開発依頼者と開発協力者が、AI 特有の側面への対応を含め協力して作業推進するうえで留意すべき点を述べる。なお、知財面での権利帰属問題や損害賠償の分担など「契約」に関しては 10.1.1 節 経産省 AI 契約ガイドラインも参考にされたい。

5.2.1 探索的アプローチへの対応

機械学習要素開発においては、開発依頼者から開発協力者への依頼開始時に明確な KPI や受け入れ条件の定義が困難な場合も多く、探索的段階型開発アプローチが必要となる。このアプローチは、いわゆるアジャイル開発を、厳密なコスト・品質が要求される製品開発プロセスに適用しようとした際に発生しがちな「品質管理が難しい」「必要費用が分からない」といった問題と類似する為、企業向けアジャイル適用のプラクティスが有益と考えられる。

たとえば、 4 節で述べた PoC フェーズは、DA (Disciplined Agile)[8] でいう「Inception」フェーズと見做し、Inception で推奨されている成果物（テスト戦略や、基本アーキテクチャ決定など）を参考に、AI の PoC フェーズの目的（出口条件）や、次フェーズへ引き渡されるべき成果物を開発依頼者、開発協力者の双方で合意形成し、ステージゲートにて双方にて確認する、といった施策が取り得る。このように PoC ステージゲートの目標設定は大変重要な意味を持つ。 また、特に AI 開発を念頭に置いた場合、下図に示すとおり PoC フェーズの活動を、「要求明確化」に関するものと、「実現可能性確認」（フィージビリティスタディ）に大別し、その二つの視点から、

- どんな成果物が次フェーズに移行するうえで条件とすべきか
- 開発協力者から開発依頼者へ成果物として引き渡されるか

を検討する事で、より抜け漏れ防止、またその後のフェーズの適切な推進へ繋がる。

[8] 企業がアジャイル開発を実際に取り入れるためのベストプラクティスをまとめたもの。2019 年 PMI (Project Management Institute) に獲得された。https://disciplinedagiledelivery.com/

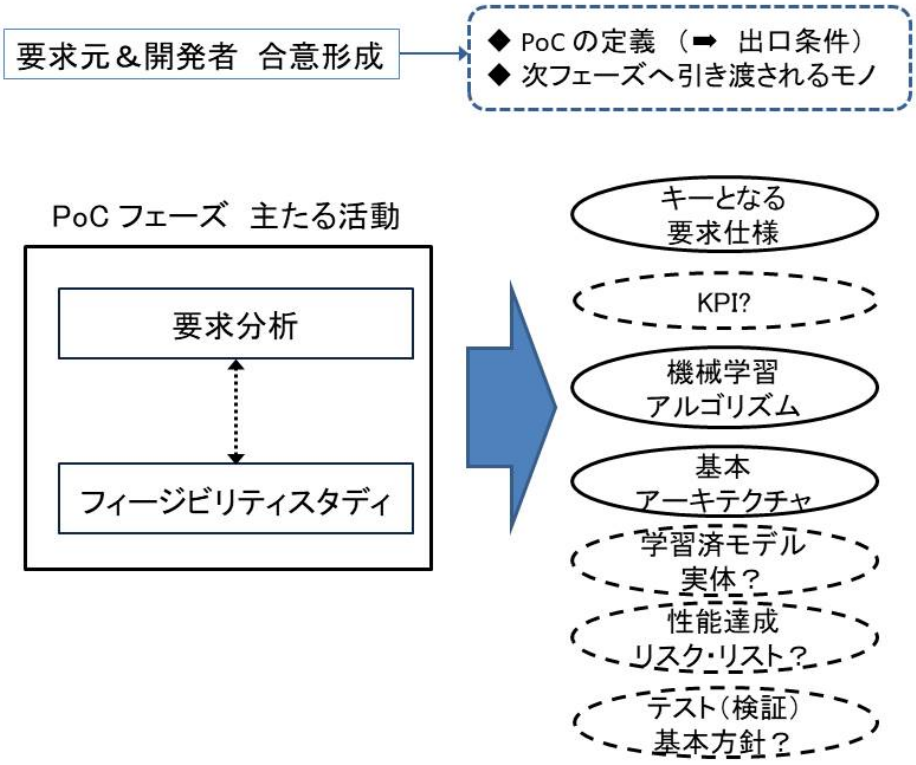

図 14: PoC フェーズの活動と成果物イメージ

5.2.2 各工程における作業内容の明確化

5.1 節で述べたプロセスに基づく作業ごとに、開発依頼者、開発協力者双方が担うべき役割を明確化する事が、双方のリスク対策になるとともに、為すべき作業の抜け漏れ防止、結果として機械学習要素の品質確保に繋がる。

たとえば、ゼロから作成するオーダーメイドなＡＩ開発を想定した場合の一例を以下に示す。

表 3 役割定義表のサンプル

各ステップ	開発依頼者	開発協力者
1.安全性適用規格確認	実施	確認
2.システムの機能要件	○○からの要件をもとに、自然文章にて提供 利用時品質特定の定性的な表現を含む	レビューし、目的・目標の明確化を図る。

3.リスクシナリオ	類似製品○○のリスクリストの用意 制約条件の提示	○○手法にて分析し、提示。必要に応じ更新する。
4.利用時品質特性特定	優先度、制約条件を提示する(達成必須な品質特性など)	三つの品質特性について目指すべき品質レベルを提示する。
5.システム構成要素設計	レビュー	遂行 レビューの便宜上○○を提供する
6.機械学習要素品質達成要求レベル特定	レビュー	レベル提案
7.機械学習要素の内部品質要求レベルの特定	レビュー	内部品質管理・保証の方法(テスト手法など)を検討し、目標とするレベルを提示。
8.規格学習要素の内部品質の実現	データセットの提供 テストレポートで確認	７で検討した方法の遂行

プロジェクトにより、行う内容の詳細は異なり、また繰り返し実施される場合一度定めた内容が不変とは限らない。

（例１）
現実解として、「この訓練用データにて学習させること」趣旨での作業依頼となる場合もある。この場合は６節で述べるデータ関連内部品質分析を実施した結果、追加の訓練用データが必要になる可能性は少なくない。　このリスクを認識したうえで、表３の７の内容を考え、また必要な段取り（開発依頼者、開発協力者双方が何をするか）を予め合意することが望まれる。

（例２）
当初、開発依頼者認識に機械学習要素が達成すべき品質への「理解不足」や「曖昧さ」が存在する場合、表３に記載したように、開発依頼者が利用時品質の側面から「優先度、制約」を適切に提示し、開発側が相当するレベルを定義することが出来ない。　こ

の場合は、最初の PoC フェーズは、利用時品質については、開発側が(システム設計の概要を踏まえたうえで、ボトムアップに)初期アイデアを出すことが有り得る。
　その後の本格開発フェーズにおいては、開発依頼者から利用時品質への提示を想定した役割定義表に変更をしていく。

PoC フェーズの場合において、表３の各工程ごとの作業内容は変わってくるが、工程が丸ごと不必要になるわけではない。いっぽう本格開発フェーズにおいては、前述したようにステージゲートの目標を適切に設定しクリアしているのであれば、通常ステップ８の中でのループ（繰り返し）で済むことになる。

しかしながら、プロジェクトによってはステージゲートの目標を落とさざるを得ない、すなわち、PoC フェーズと本格開発フェーズの区切りが実態としてあまり無い場合もあり得るが、この場合は品質管理の見通しが立たないリスクをとるのは開発依頼者になり、受け入れるかどうか、また作業分割をどうするかを含め通常開発依頼者の判断に委ねられる。

5.2.3　作業分担の詳細分けにあたって留意すべき点

前節で述べた開発ライフサイクル中の作業具体化＆取り決めをする際に、同時に留意すべき事項を、IPA/SEC「つながる世界の品質確保に向けた手引き」［126］を参考に AI 視点にて抽出・検討した結果を以下に述べる。

1）説明責任が果たせる体制・仕組み

AI 開発においては品質に関するエビデンスがプロセス視点にならざる得ない部分も多いため、その定義や承認者の明確化が必要となるうえ、さらにはプロセスに内在するリスクも存在する。たとえば、

- 訓練用データの準備について、生データの用意は発注者と定義するだけでは不足であり、前処理（クリーニングなど）は誰がどう行うのか、その目標を達成したことのエビデンスは誰がどう残し確認するのか？

などが例としてはあげられる。いずれの場合も、時間・費用制約がある中で、「双方が納得できる妥協・論理」を、事前に検討し、定めることが重要となる。

2)テスト

モデルの訓練実施までは受注側に完全に任された場合でも、「品質確認・検証フェーズ」

においては発注側として内容に踏み込んだ理解が通常求められる。そのためには、エビデンスの定義はもちろん、テストそのものの方法・スコープ、あるいは実効性・効率性の面で何を断念するのか、など可能な限り受発注時に納得感を持てるべく、取り決めが望まれる。この際、受発注両者ともに、「最終的な品質向上」を目的に協力をする事が重要であり、たとえば、提示されたテスト用データセットを訓練に用いるなど「検証クリアのみを重視した行為」は、無論避けねばならない。

3)運用時の品質管理

モデル更新の即時性如何によらず、4.3「品質監視・運用フェーズ」で述べたとおり、機械学習はその性質上、リリース後も継続的な品質管理は欠かせない。したがって、システムの機能要件に応じた最適な運用時品質管理方法の設計・実装を作業スコープに含める必要がある。

具体例としては、性能評価用データや、開発時には把握しきれない環境データなど、「取得が必要なデータ」の特定と、取得・保存の仕組みの実現に関する取り決めがあげられる。

5.3　差分開発などにおける留意点

機械学習利用システムに限らず、ソフトウェア要素を利用したシステムやサービスにおいては、既存のソフトウェア部品の再利用が頻繁に行われる。機械学習においては特に、一定のデータを訓練済みのモデルを基に、追加学習を施してカスタマイズするような応用が行われることがある。このような再利用に関する品質の保証には取扱いの難しさがある。

まず基本原則として、従前からの機能安全性の確保においても、本ガイドラインの対象とする機械学習の品質確保においても、ソフトウェア部品を再利用したシステムの品質については、あくまでその新しいシステムが利用される状況（文脈・コンテキスト）において品質を保証する方針が、利用状況の分析から機能要件定義・実装から検査にいたるまで、一貫して整合している必要がある。この一貫性を実現する方法としては、例えば以下のような方法が考えられ、システムの要求する品質レベルや部品の提供された状況などに応じて選択すべきである。

1. 再利用元の機械学習要素を構成したデータセット（旧データセットと呼ぶ）の精査を含め、新しいシステムの文脈において新たに品質管理プロセスを立ち上げ、元の要素とは独立して品質管理を行う。

2. 再利用元の機械学習要素について本ガイドラインと同等な品質管理活動が行われ、新しいシステムの要求以上の品質管理が行われていることと、特に「要求分析の十分性」について、想定する利用状況が旧要素の想定状況の部分集合（同じ場合を含む）であることが明確に確認できる場合には、元の品質管理に関する記録類を参照し、必要に応じて追加学習による品質の低下がないことなどを確認する。

   元の要素についての品質管理が十分に行われ記録されていることが前提となるため、当該部品が当初より品質を意識していない場合には適用が難しい。また、利用状況の分析は暗黙に一定の利用文脈を前提としてしまうことが多いため、包含性の判断には困難がある場合もありえる。

   また、機械学習要素を汎用部品として提供する開発者は、あらかじめこのような再利用を想定して品質管理活動を行っておき、特に前提とした品質レベルを明確にしておくことで、再利用性を向上させることができる。

3. 比較的低い品質要求レベルの応用においては、既存データセットを詳細不明なブラックボックスとした前提で、テストフェーズなどに重点を置いた品質管理活動を検討する。具体的には例えば、
   - 6.1「要求分析の十分性」および 6.2「データ設計の十分性」については、新システムで新たに分析を行い目標を設定する。
   - 6.3「データセットの被覆性」6.4「データセットの均一性」については、新たな要求分析に基づいたテスト用データセットを新たに用意し、テストフェーズにおいてこれらの性質の一定の達成を確認する。
   - 6.6「機械学習モデルの正確性」6.7「機械学習モデルの安定性」については、追加学習を行う場合にはその際の指標を用い、新たな要求分析に基づいて追加された訓練用データセットでバリデーションフェーズ・テストフェーズで評価を行う。追加学習を行わない場合には、新たな要求分析に基づいて、テストフェーズまたはシステム全体の結合テスト以降のフェーズで検証を行う。

   但し、少なくとも現在の 6 章・7 章の記述の想定では、訓練に用いたデータセットの分析を行わずに訓練結果に対するサンプリング的な検査のみを用いて十分な品質管理活動を行うことは困難であると考えている。できるだけ旧データセットなどに関する詳細な情報を入手し、前項のようなホワイトボックス的なアプローチと併用することが、現時点では現実的と考えられる。

6. 品質保証のための要求事項

本章では、主に「リスク回避性」「AI パフォーマンス」の 2 つの利用時品質の達成に対応して、機械学習要素の内部品質の管理を具体的に行うための特性として、以下の 9 つの内部品質を品質管理の特性軸として設定する。この設定に至る分析については、11.1 節にその概要を述べる。「公平性」については、本章に加えて第 8 章に公平性特有の要求事項を整理する。

6.1 A-1: 問題領域分析の十分性

6.1.1 基本的な考え方

1.7.1 節で述べたとおり、機械学習利用システムの実世界での利用状況について、十分に要求分析が行われ、その分析結果が想定される全ての利用状況を被覆していることを、「問題領域分析の十分性」とする。

要求分析は主にリスク回避性（安全性など）を要求される従来のソフトウェアの構築の超上流段階において重要とされている。本ガイドラインが想定する機械学習実装における要求分析の目的は、

① 主にリスク回避性を要求される応用において、実世界の利用状況の中でリスク対策が必要な状況を十分に特定すること、
② 主に公平性が要求される応用において、不公平であってはならない比較対照集合としての属性を十分に特定すること、
③ AI パフォーマンスを含む全ての性能要求に対して、十分な訓練用データセットやテスト用データセットが実世界を十分に網羅的かつ妥当に抽出したものであることを確認するために必要な、実世界の分析を与えること

の 3 つが主なものと考えられる。

この「要求分析の十分性」は、端的には機械学習利用システムに限らずソフトウェアによる制御を伴う機器やサービスでは必ず要求される性質である。ただし、機械学習利用システムにおいては特に、この段階でシステムの利用状況に関する見落としがあると、データの収集から実際の訓練過程・テスト工程に至るまでほとんどの段階で誤りに気付く機会がなく、

実環境での最終的なシステムテスト段階、または実際の運用段階において初めて発生する誤動作の原因になる可能性が高い。

一方で、全ての利用状況の詳細をその微細な差異に至るまで全て事細かに分析することを原則論として徹底すると、分析結果をそのまま通常のプログラムとして実装することができ、そもそも機械学習技術を用いるメリットが皆無となる。これは裏を返すと、このような網羅的かつ徹底的な分析が事前にできないような応用のシステムであるからこそ、何らかの形で機械学習により知識獲得を行わせたいという需要があることになる。

このような２つの観点から、機械学習利用システムにおける「要求分析の詳細度」の適切な設定は、品質確保と実現性・実装の効率性の両面から極めて重要であると考えられる。これは、「何を人が分析するか」「何を機械学習に獲得させるか」の判断に対応していると考えられる。この 2 つのバランスを適切に維持することが、機械学習品質管理における要求分析の重要なゴールとなる。

6.1.2　具体的な取扱い

本項の内部特性軸のゴールは、

- 機械学習要素が対応すべき要求内容の対象を明らかにすることと、
- 機械学習要素が対応すべき範囲の限定を明示すること

の２点となる。

6.1.2.1　属性（特徴の軸）及び属性値（具体的な特徴）の列挙抽出

まず初めに本ガイドラインでは、具体的な特徴として抽出しうる実世界の入力を、以下の観点で整理する。これ以降、その整理されたそれぞれの観点を「属性」と、その観点に属する具体的な特徴の種類を「属性値」と、それぞれ称することとする。
ここで、「属性」の候補として考えられる特徴としては、以下のようなものが挙げられる。

1. 機械学習要素に機能要件として要求される出力の違いを特徴付ける属性。
   教師あり機械学習においては、「教師ラベル」が異なるもの。
   例えば、数字の文字認識では「0 から 9」までの 10 通りの区別、あるいは異状検知などにおいては、異状の有無などが該当する。
2. 出力が同一であっても、機械学習要素に機能仕様レベルで明示的に対応が要求される

属性。

たとえば、文字認識における「1 と ℓ 」や「1 と /」などが、「どちらも正しく認識すること」と仕様で定められている場合が相当する。

あるいは、自動運転において回避すべき物体として「歩行者」「自転車」「交差交通の自動車」などが指定されている場合に、それぞれの物体が存在する状況などもこれに当たる。

あるいは、公平性の要求される人事採用スコアリングにおいて、例えば「性別」「年齢層」なども本項に該当する。

3. 機械学習要素に期待される性能について、実運用時の性能劣化リスクが大きく異なる可能性の高い属性。

   たとえば、屋外の物体認識において、「天候」「昼夜の別」「逆光・順光」「移動速度」などが考えられる。

4. 機械学習の特性上、期待される出力が同一であっても、学習結果のモデルが共通性を見いだしづらい・別のものとして内部的に認識する可能性のある要素。

   例えば、屋外の物体認識において、昼間と夜間では物体の異なる特徴を捉える可能性や、交通信号機の縦横の区別などが挙げられる。

   あるいは、手書き数字における「1 と /」のような形態の相違はその筆記者の出身国などにより大きく異なるいくつかの特徴に分類され、異なるタイプを全て認識させるためには、たとえ機能仕様で明示されていない場合であっても、異なる母集団として少なくとも訓練用データセットとテスト用データセットに意図的に含めることが必要となる。

5. 学習用にデータを収集する際に、「均等に集めてくる」方法をプロセスとして定めることが難しい特性。

6. 上記のような差異はないが、人が容易に把握できる対象の差異。例えば、動物の種類の違いや、皮膚の色など。

7. 人が言語で特徴を認識・説明できない対象で、かつ一方で十分に偏りのないサンプル抽出が手続き的に可能であるもの。例えば、数字の 8 のありとあらゆる書き方を事前に分析することは困難であるが、人も普段意識して書き分けていない場合には、十分に大量のサンプルを無作為的に用意すれば、偏りのない標本抽出になっていると期待できる。

8. 差異を機械的に網羅補完することが極めて容易なもの。例えば、画像の並行移動や色の変化、一定の範囲での回転拡大などは、要求仕様が一旦定まれば画像処理により機械的

にサンプルを生成することが可能であり、意図的に訓練用データやテスト用データを合成することができる。

システム全体について分析された要求などからこれらの属性の候補を列挙した上で、それぞれのリスクの高さや、機械学習を利用する応用の特性（PoC 試行の結果を参考にする）に応じて、それぞれの特性を属性として抽出して開発者による管理の対象とするか、抽出せずに「機械学習に任せる」かを検討する。一般論としては、上記の列挙で始めの方、1〜3 に掲げたような特性ほど、属性として認識する必然性が高いと考えられる。また 7〜8 項のような特徴は、最終的には属性とせず、ほぼ確実に機械学習に「任せる」ことが妥当であると考えられる。ただし、機械学習に「任せた」属性については、後に性質 3「データセットの被覆性」においてその属性に対応するデータ種別の多様性を検討することになるので、このことも踏まえた分析が必要である。

6.1.2.2　除外する属性組み合わせの検討

また、本項では同時に、「あり得ない属性値の組み合わせ」についても検討を行う。これは、例えば「（東京地方での）夏の雪」のような、機能要件として排除すべきレアなケースを、例えばごく稀に起こる「冬の積雪」のような、機能として対応すべきレアケースと峻別することになる。このような整理は極めて重要で、この段階で区別をしておかない限り、あるデータ欠損が妥当なものか望ましくないものかを判断するすべがなく、この後の品質管理において「品質管理手法自体の品質」を説明することができなくなる。

具体的には、属性とそれに対応する属性値を列挙した後に、システムへの要求などに基づきあり得ない属性値の組み合わせを書き出すことになる。

6.1.3　品質レベル毎の要求事項

要求分析の十分性については、各外部特性のレベルに応じて、以下の 3 レベルの取扱いを要求事項とする。

外部特性レベルと本内部特性の要求レベルの対応は以下の通りである。

「リスク回避性レベルについて」

AISL 0.1 → Lv 1 以上

AISL 0.2 → Lv 2 以上

AISL 1 → Lv 3

AISL 2～4 → Lv 3 に追加すべき要求について、今後検討する。

「AI パフォーマンスについて」

AIPL 1 → Lv 1 以上

AIPL 2 → Lv 2 以上

「公平性について」

AIFL 1 → Lv 1 以上

AIFL 2 → Lv 2 以上

上記の内部特性の要求レベルごとの要求事項は以下の通りである。

- Lv1
  - 主要な品質低下リスクが発生する原因について検討を行い記録する。
  - その検討結果に基づき、データの設計を行い必要な属性などに反映する。
- Lv2
  - システム全体での利用時品質低下リスクとその影響について、工学的に一定の網羅性をもつ分析を行い、文書として記録する。
  - それぞれのリスクについて対策の要否を分析し、機械学習要素への入力においてそのリスクに対応する特徴となる属性について分析を行う。
  - また、応用に即した機械学習要素の入力をもたらす環境の特徴について、機械学習の容易さなどの分析を行い記録する。
  - これらの分析結果に基づいて属性と属性値のセットの検討を行い、その決定の経緯を記録する。
- Lv3
  - Lv2 に加えて、以下の活動を行う。
  - システムの利用環境の特徴量として捉えるべき要素について、過去の自己・他者の検討結果などの文献調査を行い、必要な集合の抽出に至る検討経緯を記録する。
  - システム全体の利用時品質低下リスクについても、そのシステムの応用分野に即した過去の検討結果などを調査し、取捨選択の経緯も含めて検討経緯を記録

する。

- また、システム全体の利用時品質低下リスクについては、Fault Tree Analysis などの工学的分析を用いた抽出も行い、その結果を記録する。

6.2 A-2: データ設計の十分性

6.2.1 基本的な考え方

前項の要求分析の十分性を前提として、システムが対応すべき様々な状況に対して十分な訓練用データやテスト用データを確保するためのデータ設計の十分な検討を、「データ設計の十分性」として要求する。より具体的には、訓練用データセット準備以降、テスト工程までの段階で着目する属性値の組み合わせの数や内容についてこの段階で検討を行う。

理想的な状況として、例えば、前項で十分と考えられる属性の組み合わせに対して、全ての属性値の組み合わせ（属性の直積）に対応する十分なデータがあれば、実世界の全ての状況を網羅していると言うことができる。しかし、現実のシステム開発においては、属性の数が 10 以上になることは十分に考えられ、属性値の組み合わせの数が 1 万～100 万程度になることもよくある。このような場合に適切な「網羅性」を考え「設計する」ことが、本項での品質管理の要点となる。

実際に品質管理を行う上では、2 つの観点に着目することが重要となる。1 つは、誤動作・誤判定などを引き起こす可能性のある属性の組み合わせは、確実に訓練や検査段階で対応しておく必要がある。また同時に、実装する機械学習利用システムが運用時に遭遇しうる全状況に対しても、訓練の品質と成果物の品質の双方の観点からできるだけ網羅する必要がある。

このような問題は、従来のソフトウェア開発において「テスト設計」の課題として取り組まれてきていたものであるが、テストだけでなく実装工程もデータセットを元に行う機械学習においては、同じ課題が実装工程において必要になっていることがユニークな点である。

従来のソフトウェア工学ではこのようなテスト設計の組み合わせ爆発にはいくつかの現実的な解決策が既に示されており、本ガイドラインでは、この既存の知見を機械学習応用に適用することで、現実的かつ十分な訓練や品質検査を行うことを目指している。

6.2.2　具体的な取扱い

まず、認識すべき属性が極めて少なく、全ての属性の属性数の積（前項のあり得ないケースを差し引いたもの）が高々10～20 程度であれば、これらの組み合わせをそのまま「ケース」とよび、後の段階でテスト用データセットや訓練用データセットに全てのケースが含まれることを確認することにする。

一方で、これらの属性の積をそのまま用いたのでは組み合わせ数が多すぎる場合や、特にレアケースにおいて十分なデータを得ることができない場合には、属性値のうちのいくつかを取りだした組み合わせを一定の基準で抽出して「ケース」として設定し、これらの組み合わせを網羅することとする。例えば、1.7.1 節の例 1 に掲げた交通信号機の例においては、例えば「昼間の緑」「昼間の黄」「夜間の赤」のほか、「昼間の雨の状況」「夜間の雨の状況」「雨の中での赤信号」などの組み合わせを抽出し、それぞれをケースと呼び、必ずテスト用データセットにこれらの組み合わせに該当するデータが含まれるようにすることで、完全な属性全ての網羅が不可能な場合でも、例えば「夜間の雨」のような高リスクな事例が全くデータセットに含まれていないことや、「赤信号」を全く学習させていないといった事例を排除して、ある程度の「網羅性」を得ることを目指す。なお、このような「間引いた網羅性」を考える場合には、「昼間の雨の赤信号」のデータは、「昼間の雨」「昼間の赤信号」「雨の赤信号」など、複数のケースに同時に含まれることになる。

さらに、高い品質を要求される応用では、これらの抽出作業に数学的な「網羅性基準」を導入することで、完全性をより具体的に保証することにする。ソフトウェア工学のブラックボックステストにおいては、ランダムサンプリングのサンプル率の他にも直交表やペアワイズテストなどの手法とそれに基づく「網羅性基準」が知られており、応用毎に適切な手段を選択することになる。

6.2.3　品質レベル毎の要求事項

要求分析の十分性については、各外部特性のレベルに応じて、以下の 3 レベルの取扱いを要求事項とする。

外部特性レベルと本内部特性の要求レベルの対応は以下の通りである。

「リスク回避性レベルについて」

AISL 0.1 → Lv 1 以上

AISL 0.2 → Lv 2 以上

AISL 1 → Lv 3

AISL 2〜4 → Lv 3 に追加すべき要求について、今後検討する。

「AI パフォーマンスについて」

AIPL 1 → Lv 1 以上

AIPL 2 → Lv 2 以上

「公平性について」

AIFL 1 → Lv 1 以上

AIFL 2 → Lv 2 以上

上記の内部特性の要求レベルごとの要求事項は以下の通りである。

- Lv1
  - 主要なリスク要因に対応する属性について、それぞれに対応したケースを設定すること。
  - さらに、複合的なリスク要因については、その組み合わせに対応したケースを設定すること。
  - また、特に重要と考えられる環境要因の差異に対する属性を抽出し、大きなリスクの要因との組み合わせに対応するケースを用意すること。
- Lv2
  - Lv1 の要求を全て満たすこと。
  - 特に重要と考えられるリスク要因については、原則として pair-wise coverage の基準を満たすこと。具体的には、「その原因の組み合わせの属性値」と、「その属性値の属する属性以外の全ての属性について、属性に含まれる属性値を 1 つずつ個別に選択したもの」の組み合わせのケースを含むこと。
- Lv3
  - 工学的な検討に基づき、属性の網羅基準を設定し、その網羅基準を満たす属性値の組み合わせの集合をケースとして設定すること。
  - 網羅基準の厳密さ（pair-wise coverage, triple-wise coverage など）は、システムの利用状況やリスクの重大さなどを加味して設定されること。必要な場合には、個別のリスクに応じてリスク毎に基準を個別に設定することも考えられる。

6.3 B-1: データセットの被覆性

6.3.1 基本的な考え方

前項で基準を定めて網羅したそれぞれのケースに対して、それぞれのケースに対応する入力の可能性に対して抜け漏れなく、十分な量のデータが与えられていることを、「データセットの被覆性」と称する。

通常のソフトウェアの開発においては、ソフトウェアの動作が依存する全ての実世界の特徴の子細については、いわゆる要求分析フェーズから実装フェーズまでの少なくともいずれかの段階で把握され、最終的にプログラム内の条件分岐や計算式などとして反映されることになるが、機械学習要素の構築においては、6.1 節で述べたとおり、ある程度以上の子細な状況の差異は ML 要求分析の属性や訓練時の「正解ラベル」などとして把握されずに、学習に用いるデータセットの分布として、機械学習の訓練フェーズを通じて最終的な動作に反映されることになる。このような「属性値」として特定されていない子細の特徴について、不足したデータによる不適当な学習の振る舞いが起こらないことを保証するのが、本特性軸を設定する目的である。

6.3.2 具体的な取扱い

本項の目的は主として「量」と「入力の状況に対して網羅的であること」の 2 点があるが、属性としてその内部の差異が特定されていない特徴がある以上、後者の網羅性は一般的には、データを収集・処理するプロセスにその多くを依存せざるを得ないと考えられる。一方で、6.2 節で網羅性基準を導入している場合には、それぞれのケース毎に、「ケースに含まれない」属性が偏り無く分布していることを検査することは十分に考えられる。

また、量に関しては基本的には前節でのケースの取り方の粒度を適切に取ることが重要となるが、例えば発生頻度の低いレアケースでは特定のケースのデータがどうしても十分に得られない場合も考えられる。その場合には、例えば特定のデータの欠落したケースに対しては「データセットの被覆性」を部分的に諦め、より緩い網羅性基準でカバーをした上で、システム全体のテストなどにおいてその対応状況を評価するなどの対応が必要になるなど、開発プロセス全体での対応となる場合も考えられる。

6.3.3　品質レベル毎の要求事項

外部特性レベルと本内部特性の要求レベルの対応は以下の通りである。

「リスク回避性レベルについて」

AISL 0.1 → Lv 1 以上

AISL 0.2 → Lv 2 以上

AISL 1 → Lv 3

AISL 2～4 → Lv 3 に追加すべき要求について、今後検討する。

「AI パフォーマンスについて」

AIPL 1 → Lv 1 以上

AIPL 2 → Lv 2 以上

「公平性について」

AIFL 1 → Lv 2 以上

AIFL 2 → Lv 3 以上

（公平性を損なう要因で重要なデータの偏りに対応するためには、
この品質特性が重要となることを考慮している。）

上記の内部特性の要求レベルごとの要求事項は以下の通りである。

- Lv1
  - テスト用データセットの取得源や方法を検討し、応用の状況に対して偏りがないことを期待できるようにすること。
  - 各ケース毎に、元データから偏りのないサンプル抽出などを行い、偏りがないことを期待できるようにすること。
  - これらの偏りを入れないために行った活動について、記録を行うこと。
  - 分析した各ケースについて訓練用データおよびテスト用データが十分に存在することを、訓練フェーズやバリデーションフェーズなどで確認すること。
  - ケースに対して訓練用データが十分に取得できない場合には、網羅基準を見直して緩めた上で、当初の基準に照らして個別にシステム結合テストなどで確認すべきことを記録しておくこと。
- Lv2

- Lv1 に加えて、以下の取り組みなどを行うこと。
- 各属性値または各ケース毎に、およその出現確率の想定を把握すること。
- 取得できたデータがその分布から外れていないことを確認すること。
- 各ケース毎に、中に含まれるデータの被覆性について、取得方法以外の何らかの積極的な確認を行うこと。
- 例えば、各ケース毎に、そのケースに含まれない属性がある場合、その属性に関する分布を抽出して、著しい偏りがないことを確認すること。

・ Lv3

- Lv2 に加え、各ケース毎に、中に含まれるデータの被覆性について、一定の指標を得ること。
- 例えば、特徴量抽出などの技法を用いて、ケース組み合わせに含まれる属性値以外のデータ間相関がないことなどを確認すること。
- あるいは、各ケース毎の、ケースに含まれない属性の分布について、あらかじめ想定される分布を検討し、相違について分析を行い記録すること。

6.4 B-2: データセットの均一性

6.4.1 基本的な考え方

一方で、上記の「ケース毎の被覆性」の評価だけでは、データセット全体が入力データの表現する環境全体に対する良いサンプリングになっているとは必ずしも言うことができない。ケース毎の発生確率が大きく異なる場合に、ケース毎にサンプルを準備しただけでは、全体としては大きな偏りを持ったデータセットが生成され、特に AI パフォーマンスの観点での性能を大きく損なう可能性がある。一方で、とりわけ発生頻度の少ないレアケースに対する性能が要求される場合には、入力全体にわたって偏りない均等なデータを現実的な量で用意することと、レアケースに対して十分な量のデータを用意することは一般に両立しない。例えば、百万分の一の頻度で発生する事象に 100 件の訓練用データが必要な場合、不偏な全データの件数が一億件となることは一般に許容できないであろう。このような観点から、本項の均一性は、前項の被覆性と場合によっては相反して適切な妥協が必要となることが考えられる。

一般論として、リスク回避性が強く求められるケースでは、正しい判断で回避すべきリス

クのある属性値の組み合わせに対して十分な訓練用データがあることが求められるであろうが、特にそのようなリスクが稀に発生する場合、その稀なケースにおける十分な「データ量」で他の全てのケースを訓練しようとすると、必要なデータ量が膨大になる可能性がある。このような場合、特に稀なリスクケースを「重点的」に訓練することは十分に考えられる。

一方で、全体的な性能（AI パフォーマンス）が求められる場合には、稀なケースを実際の発現確率以上に重点的に訓練することで、却って他のケースにおける推論精度が劣化し、全体としての平均性能を悪化させる可能性もある。このような場合には、前節の詳細なケースに対する被覆性を深く追求することは、必ずしも適切で無い可能性がある。

また、公平性が強く求められる場合には、「どのような公平性」が求められているかに依存して、ケース間で人工的に同等な学習をさせるべきか、抽出された訓練用データセットの分布に即して無作為に学習をさせるべきかが変わってくる可能性がある。

6.4.2　具体的な取扱い

基本的な考え方そのものは、前 6.3.2 節の被覆性において、「全体」というケースに着目することと変わらない。データセット全体を取得するプロセスに偏りが生じないよう配慮しつつ、個々の属性値の発生頻度などを適宜監視することとなる。

むしろ基本的な考え方で述べたとおり、本項では前節の網羅性とどのように両立させるかの検討やデータ設計が重要になると考えられる。

6.4.3　品質レベル毎の要求事項

外部特性レベルと本内部特性の要求レベルの対応は以下の通りである。

「リスク回避性レベルについて」

AISL 0.1 → Lv S1 以上

AISL 0.2・1 → Lv S2 以上

AISL 2〜4 → Lv S2 に追加すべき要求について、今後検討する。

「AI パフォーマンスについて」

AIPL 1 → Lv E1 以上

AIPL 2 → Lv E2 以上

「公平性について」

AIFL 1・2 → Lv E2 以上

上記の内部特性の要求レベルごとの要求事項は以下の通りである。

- Lv E1
  - 前節「データセットの被覆性」Lv 1 に同じ。
- Lv E2
  - 前節「データセットの被覆性」Lv 2 に同じ。但し、想定する出現確率については想定事象の全集合に対して比較する。
- Lv S1
  - 前節 L1 で検討したケース毎のデータ量に関して、リスクに対応するケースにおいて十分なデータ量が存在することを明示的に確認すること。
  - 訓練用データの全体集合の量、レアケースの出現確率を比較して、レアケースのデータが訓練に不足する場合には、レアケースの学習を重点化することを検討すること。但し、特に Lv E2 が要求される場合には、重点化に伴い他のケースの学習が弱化することの、システム全体の品質への影響について必ず検討を行うこと。
- Lv S2
  - Lv S1 に加え、リスク事象毎・ケース毎の出現確率の想定に基づき、各ケースのデータ量を事前に見積もり設計すること。

6.5 B-3: データの妥当性

6.5.1 基本的な考え方

「データの妥当性」(Adequacy of data) は、機械学習訓練やテスト工程などで使われるデータに誤りや不適切なものが含まれないことを意味し、様々な観点を内部に含む。

SQuaRE (ISO/IEC 25012) [8] で定めるデータ品質は、データ固有の観点と、使われるシステムに依存した観点で分類すると、以下のようなものになる。そのうち、下線を付した項は、特に機械学習の構築元データとして重要性が高いと考えられる。

- データ固有の品質特性

- ➢ ①正確性（Accuracy）
- ➢ ②完全性（Completeness）
- ➢ ③一貫性（Consistency）
- ➢ ④信憑性（Credibility）
- ➢ ⑤最新性（Currentness）

・ データ固有の視点及びシステム依存の視点の双方からのデータ品質特性

- ➢ アクセシビリティ（Accessibility）
- ➢ 標準適合性（Compliance）
- ➢ 機密性（Confidentiality）
- ➢ 効率性（Efficiency）
- ➢ ⑥精度（Precision）
- ➢ ⑦追跡可能性（Traceability）
- ➢ 理解性（Understandability）

・ システム依存の視点からのデータ品質特性

- ➢ 可用性（Availability）
- ➢ 移植性（Portability）
- ➢ 回復性（Recoverability）

また、機械学習のプロセスの観点から見ると、

A）データ選択適切性:

データセットの被覆性・均一性（B-1、B-2）で固定されたポリシーに対して、データセット中のある 1 つのデータが妥当なものである

- ➢ 例えば、測定ミスを含むデータや、外れ値として除去すべきデータでない

[① 正確性 ② 完全性 ④ 信憑性 ⑤ 最新性 ⑥ 精度 ⑦ 追跡可能性]

B） ラベリングの適切性:

データセット中のある 1 つのデータに対して、データセット準備段階で付加された情報が適切なものである。

[② 完全性 ③一貫性 ④ 信憑性]

の 2 つの要素があり、SQuaRE の特性とは概ね角括弧内のような対応と考えられる。

人工知能分野のデータ品質としてしばしば指摘され、またセキュリティの観点でも重要であることが多いデータ源が真正であること、「真正性 Authenticity」は、上記の品質特性における「④信憑性」「⑦追跡可能性」と対応づけられると考えられる。

6.5.2 具体的な取扱い

データの妥当性は最終的には「良い機械学習要素が得られること」を目指すものであるが、品質マネジメントの観点では基本的に、内部品質特性 A-1～B-2 までで定められたシステムの期待するデータへの要求に対して評価されるものとして考える。

本品質特性についての具体的な評価においては、データそのものの検査や構築プロセスの管理などにおいて様々な観点を評価する必要がある。

6.5.2.1 ラベリングのポリシーの統一・精査

教師データとして付与するラベルそのものは A-1 の問題領域分析の完全性までで特定されているが、実際のデータに対しては様々な揺らぎや迷い、曖昧さなどが生じうる。例えば、画像の特徴検出において、ラベルとして抽出すべき物体のサイズや距離、重なり合う物体の遮蔽状態の扱いなどは、機能要件との関係から明確にしておく必要がある。このような観点を作業者で統一しておかないと、ラベルの揺らぎが訓練工程における精度の低下やテスト工程における不正確な検査に繋がることが有り得る。

さらに、PoC 段階の繰り返し試行や精度不足による手戻りなどで、要件が拡張された場合には、ラベリングそのものを拡張・修正する必要があるが、一般に再ラベルの作業はコストが高く付く上、ここでも作業者の間でラベルのブレが生じる場合もある。更に、追加でデータを取得する場合には、データそのものの環境条件が統一できない場合も有り得る。

このような観点から、PoC のなるべく早い段階で十分な検討を行い、ラベリングのポリシーをなるべく詳細に固めておくことと、それを文章化して記録しておくことは、品質管理の追跡性の観点からも重要であると考えられる。

6.5.2.2 データセットの整合性チェック・再チェック

機械学習システムの構築においては、既存の測定済みデータ群や、ラベリング済みのデータセットを用いることが有り得る。しかし、データの妥当性は要件との整合性において評価されるものであるから、機能要件や前提となる使用環境が変われば、既存のデータの妥当性は再評価がされなければならない。また、またデータ収集やラベリング作業を外注する場合には、品質管理の観点から受入検査を行う必要がある。このような検査をどのように行うかなどについては、プロセス構築以前の段階から十分な事前検討を要する。

6.5.2.3　ロングテールの扱いと、計測ミス・外れ値の判断

あるデータが他のデータから傾向として外れていること自体は、統計的な分析などを用いてある程度自動的に篩をかけることは可能である。しかし、そのような希少なデータをロングテールのような意味のある訓練対象として採用すべきか、あるいは外れ値・測定ミスの棄却すべき値とするかは、問題の性質と個別のデータの内容により変わりうるし、リスク回避性と AI パフォーマンスの外部品質の優先度によっても変わりうる。この点についても、明確なポリシーないし、迷ったときの判断プロセスを決めておかないと、データ選択やラベリングのブレが生じる。

6.5.2.4　データ汚染への対応（セキュリティ・信憑性）

訓練データやテストデータに意図的な誤りや偏りを含めうる場合や、測定環境に悪意の改変（センサーへの干渉など）が有ると、最終的なシステムの機能性に重大な影響を与える場合が有り得る。このような誤りはテストデータに汚染があるとテスト工程でも検知できず、また一般にはシステムテストなどでも十分に防ぐことができないことが有り得る。

そのような観点から、データそのものの情報セキュリティ的保護は当然として、データ取得環境の物理的なセキュリティや多様性などについて、プロセス管理的な観点から担保が必要で、またそのような品質担保活動の記録を残しておくことも重要である。

また、特に外部からデータを調達する場合において、現時点においてはデータ汚染を訓練以前に検知できる汎用の方法は存在しない。そのような観点から、一定以上の品質を求められるシステムにおいては、現時点ではデータセットそのものや提供主の信用性や追跡可能性などに、データ汚染対策を依存しなければならない場合がある。

6.5.2.5　最新性

機械学習応用においてしばしば、訓練データ取得から時間が経過するにつれ、性能が劣化していくことがある。訓練データの最新性の管理は、このような品質劣化に対する予防として重要である。一方で、最新性の要求はしばしば、訓練に使用可能なデータの量と対立することがあり、特にレアケースへの対応が必要な場合において、網羅性との間でトレードオフが生じうる。このような観点から、最新性についてのポリシーはきちんと事前に検討するか、あるいは PoC 段階などで妥当な要求水準をきちんと洗い出す必要があると考えられる。

6.5.2.6　プロセス・体制・仕組み

以上述べたように、データの妥当性の担保においては、具体的な個々のデータの取扱いだけでなく、構築の全体プロセスや監査も含めた体制などに依存しなければならない要素が多々ある。また、実際の機械学習システム構築においては、データ妥当性の判断を行う工程で具体データを扱うことで、要件定義などの詳細化に繋がり前段の内部品質特性 A-1 などへ波及することがしばしば有り得る。このような作業は本ガイドラインでは PoC 段階での試行錯誤と整理しているが、このような循環的な作業を生じた場合に、最終的にきちんと整合性が取れているかの検証は特に重要で、しばしば見落としを生じうる一方、データ検査の全てを毎回の試行で 1 からやり直すことも現実的ではない。このような観点から、開発中のポリシーの変更管理等も含めて、きちんと工程管理を行う体制・仕組み作りが、品質管理には求められると考えられる。

6.5.3　品質レベルごとの要求事項

外部特性レベルと本内部特性の要求レベルの対応は以下の通りである。

「リスク回避性レベルについて」
　　AISL 0.1 → Lv 1 以上
　　AISL 0.2 → Lv 2 以上
　　AISL 1 → Lv 3
　　AISL 2〜4 → Lv 3 に追加すべき要求について、今後検討する。
「AI パフォーマンスについて」
　　AIPL 1 → Lv 1 以上
　　AIPL 2 → Lv 2 以上
「公平性について」
　　AIFL 1 → Lv 1 以上
　　AIFL 2 → Lv 2 以上

上記の内部特性の要求レベルごとの要求事項は以下の通りである。

・　Lv1

- 全般:
  - ✧ データの出所が問題に対して適切かどうかをきちんと検討・確認すること。
  - ✧ ラベリングのポリシーを整理すること。
  - ✧ ラベリング・外れ値除去の判断基準を事前に検討しまとめておくこと。
  - ✧ 与えられたデータに照らして判断基準が妥当かどうかを判断し、場合によっては基準の見直しと再チェックを行う事。
  - ✧ ラベル付データを用いる場合には、既存ラベルの妥当性の事前検討を行い、必要に応じて事前テスト等で確認すること。
- ラベルの揺らぎ:
  - ✧ 作業者間で統一した基準を定めてラベルの判断をおこなうか、ダブルチェックを行う事。
- データ汚染:
  - ✧ データ源への汚染の影響と可能性を検討すること。
- 最新性:
  - ✧ データセットに不適切な時期のデータが含まれないことを、問題特性に合わせて事前に検討すること。

・ Lv2

- Lv1 に加えて以下の対応を取る。
- 全般:
  - ✧ データの準備段階をきちんと品質管理プロセスに組み込んで管理すること。
  - ✧ データを外部調達する場合、データの準備方法・処理方法・品質管理プロセス・セキュリティ管理を要件に組み込むこと。
- ラベリングのポリシー:
  - ✧ 作業者によるラベリングのばらつきを除去するための管理プロセスを構築すること。
  - ✧ データ属性の定義が変更されたときのラベルの変更管理をプロセス化すること。
- ラベルの揺らぎ:
  - ✧ 許容されるラベルの揺らぎの範囲を事前に検討し文書化すること。
  - ✧ 作業中に生じた揺らぎに関するラベルの判断を記録すること。
- データ汚染:
  - ✧ 可能な範囲で訓練データの検査を行う手法を検討すること。

- ✧ Adversarial Example への対処をおこなうこと。
- ✧ 異常を実行時に検知する設計を検討すること。
- ✧ 詳細については、第 9 章のセキュリティを参照。

➢ 最新性:

- ✧ データの準備段階をきちんと品質管理プロセスに組み込んで管理すること。

➢ 事後検査:

- ✧ 可能であれば、入力の影響度分析・ニューロンの発火状況その他の内部的な情報の分析の適用を検討し、可能な範囲で明らかな誤りを手動で排除すること。

・ Lv3

➢ Lv2 に加えて以下の対応を取る。

➢ ラベリングのポリシー:

- ✧ ラベル設計の影響によるリスク分析を行い記録すること。

➢ ラベル・データ除去の確認:

- ✧ 外注時に受入検査でのダブルチェック、または監査プロセスの事前設定・検査を行うこと。

➢ データ汚染:

- ✧ データ汚染によるリスク分析を行い記録すること。

6.6 C-1: 機械学習モデルの正確性

6.6.1 基本的な考え方

学習データセットに含まれる入力に対して、機械学習要素が期待に反しない反応を示すことを、「モデルの正確性」と称する。

6.6.2 具体的な取扱い

基本的には主にバリデーションのフェーズで訓練用データセットに対する収束性をみて判断し、テスト用データセットでその達成を確認することになるが、入力値に確率的な広がりやノイズが含まれるような現実的な場合には、特に出力値が連続的である場合に「100%

の再現性達成」がゴールとならない（むしろ過学習を起こしている状況に対応する）場合が有り得る。

そのため、データの量を変化させたり、いわゆる交差検定の手法を用いるなど、Accuracy などの指標の相対的な振る舞いを見て、学習の達成度を評価する必要が一般にある。

具体的な手法は応用に応じてデータサイエンティストが選択し、きちんとその選択の背景を説明できるようにする必要がある。いくつかの具体的に適用可能かもしれない技術については、7.6 節に例示する。

6.6.3　品質レベル毎の要求事項

外部特性レベルと本内部特性の要求レベルの対応は以下の通りである。

「リスク回避性レベルについて」

AISL 0.1 → Lv 1 以上

AISL 0.2 → Lv 2 以上

AISL 1 → Lv 3

AISL 2〜4 → Lv 3 に追加すべき要求について、今後検討する。

「AI パフォーマンスについて」

AIPL 1 → Lv 1 以上

AIPL 2 → Lv 2 以上

「公平性について」

AIFL 1 → Lv 1 以上

AIFL 2 → Lv 2 以上

上記の内部特性の要求レベルごとの要求事項は以下の通りである。

- Lv1
  - テスト用データとして必要なデータ量を PoC 仮定や過去の経験から導き出し、「データの被覆性」を満たす抽出プロセスを通じて用意すること。
  - 訓練用データセットについても上記に準じた取扱いとする。ただし、データの分布の取り方については違う方法を採用して良い。
  - テスト段階において一定量の誤判断を許容する場合（false negative/false positive

で扱いを変える場合を含む）については、その判定基準を合理的に事前に決定し、記録しておくこと。

- 公平性が要求される場合には、予め公平性の比較手段を定めておくこと。対照テストの結果による場合には、その合格基準を予め定めておくこと。

・ Lv2
- Lv1 に加えて以下の対応を取る。
- 正解率（Accuracy）などのバリデーション段階での合否判定についても、その合理的な判定基準を事前に決定し記録しておくこと。
- 実データでのテストと、可能な範囲でのデータ変形などでの機械的な増量テストを同時に行うこと。
- 可能であれば、入力の影響度分析・ニューロンの発火状況その他の内部的な情報の分析の適用を検討し、可能な範囲で明らかな誤りを手動で排除すること。

・ Lv3
- Lv2 に加え、学習成熟状況の内部確認手段などを事前に検討すること。
- 結合テスト以降のシステム全体での検証計画と機械学習要素のテスト計画の対応を明示すること。
- 特にリスクが大きいケースを中心に、システムレベルでのテスト時の機械学習要素の要件との対応をテスト計画に反映し、その被覆状況を監視・確認すること。

6.7 C-2: 機械学習モデルの安定性

6.7.1 基本的な考え方

機械学習モデルの安定性とは、データセットに含まれない入力に対して、機械学習要素が期待通りの反応を示すことである。低い安定性は、未知の入力に対する予測性能の低下（AI パフォーマンスの低下）や、潜在する重大な誤判断によるリスクの増加（リスク回避性の低下）をもたらす。そのため、特に安全性が要求される場合の安定性の評価は重要である。例えば、安定性に関しては次の 2 つの問題がよく知られている。

・ 訓練用データセット以外の入力に対して大きく外れた推論をする問題：
このような問題は、例えば、訓練用データセットに対する過度な適合（過学習）によ

って生じることがある。

・　意味を変えない程度の入力の微小変化に対して出力が大きく変動する問題：
これは機械学習特有の問題として知られている。微小変化は自然界のノイズ（例えば、カメラレンズの汚れ）による場合（擾乱）の他、敵対的データと呼ばれる意図的な攻撃の場合（摂動）もある。攻撃には、サイバー空間のデータ改ざんや物理世界の対象物への細工（例えば、道路標識への微小なシールの貼付け）などがある。

6.7.2　具体的な取扱い

安定性は機械学習ライフサイクルの次に示す 3 つのフェーズとの関係が強く、安定性の目標達成に向けて、主にこれらのフェーズで安定性の評価と向上を行う。そのために適用可能な技術については 7.6.2 節で例示する。

・　反復訓練フェーズ：訓練用データセットとバリデーション用データセットの分離、訓練過程の監視などによって、訓練用のデータセットへの過剰な適合を回避する他、入力の微小変化の出力への影響評価などによって、ノイズに対する耐性を向上させる。
・　品質確認・評価フェーズ：テスト用データセットによる正解率や適合率などの評価の他、入力の微小変化に対する出力の変動の評価を行う。
・　品質監視・運用フェーズ：運用時に誤推論しやすい入力を検知して排除する。

6.7.3　品質レベル毎の要求事項

機械学習モデルの安定性を確認できるように、その安定性を向上するために適用した技術とその評価結果を記録することが求められる。特に、各レベルにおいて記録すべき事項を以下に示す。ここで、近傍データとは、元データに微小変化を加えて生成されるデータのことである。適用可能な具体的な技術については 7.6.2 節で説明する。

外部特性レベルと本内部特性の要求レベルの対応は以下の通りである。

「リスク回避性レベルについて」

AISL 0.1 → Lv 1 以上
AISL 0.2 → Lv 2 以上
AISL 1 → Lv 3

AISL 2～4 → Lv 3 に追加すべき要求について、今後検討する。

「AI パフォーマンスについて」

AIPL 1 → Lv 1 以上

AIPL 2 → Lv 2 以上

「公平性について」

AIFL 1 → Lv 1 以上

AIFL 2 → Lv 2 以上

上記の内部特性の要求レベルごとの要求事項は以下の通りである。

- Lv1：安定性向上のために適用した技術を記録すること
  - Lv1 では、過学習を防止するために広く利用されている交差検証や正則化などの技術の適用が推奨される。
- Lv2：近傍データによる安定性の評価結果を記録すること
  - Lv2 では、データセットの各データの近傍に対する安定性を評価することが求められる。例えば、近傍の敵対的データによる攻撃を防御する技術の適用が推奨される。敵対的データを生成して安定性を評価する技術、敵対的データの攻撃を受けにくくする訓練技術、敵対的データの動的検知技術などがある。これらの技術を適用することは容易ではないが、現在、そのための実用的な開発・評価するための環境整備が進められている。
- Lv3：データセットの各データ以外の安定性を保証すること
  - レベル 3 では、データセットの各データ以外のデータに対して一定の安定性をもつことを保証することが求められる。例えば、近傍には敵対的データが存在しないことを保証する技術や汎化誤差上界を確率的に保証する技術などがある。これらの技術はまだ研究段階であるが、将来的にはレベル 3 での適用が期待される技術である。

6.8 D-1: プログラムの信頼性

6.8.1 基本的な考え方

訓練用プログラムや予測・推論プログラムなど、訓練済み機械学習モデルを除く純粋なソフトウェア部分（C 言語などで記述された従来のプログラム部分）が、ソフトウェアプログラムとして仕様通り正しく動作することを、「プログラムの信頼性」と称する。アルゴリズムとしての正しさの他、メモリリソース制約や時間制約の充足、ソフトウェアセキュリティなど一般的なソフトウェアとしての品質要求がここに包含される。

組み込み機器などにおいて、MPU などの電子的ハードウェアが開発に含まれる場合には、それらの信頼性の確保も本項に準ずる。

機械学習の応用においては、学習訓練及び実運用の双方において、いわゆるオープンソースの実装を用いることが非常に多い。世に知られるオープンソース実装には極めて多数の利用者がいてその品質が間接的に評価されているものもある一方で、品質がまだ十分に検証されていないものや、性能向上のためのバージョンアップが頻繁で新たなバグ混入のリスクが高いものなどもあり、最終的には開発者の責任において十分な品質が確保されることが求められる。

また、一般にソフトウェアの正しさの検査は実際の動作環境と同一性を確認できる環境で行うことが大前提であり、動作環境が異なる場面ではその動作環境の差異の影響をきちんと評価する必要がある。しかし、機械学習の応用においては特に、訓練などにもちいる環境（例えばクラウドや GPU 付き計算機環境）と実際の動作を行う環境（例えば組み込み計算機）が異なり、数値的な計算の振る舞い（浮動小数点演算の精度など）ですら変化する場面も多く見受けられるほか、さらにモデルの圧縮など数値的処理そのものを変化させることも稀にある。実際の機械学習実装において、これらの環境変化などが予測精度などに影響を与えることがあることも既に知られており、慎重な取扱いが必要となる。

6.8.2 具体的な取扱い

純粋なソフトウェアとしての品質確保そのものは、従来の計算機応用などでの品質管理手法を適宜適用する。

オープンソース実装の利用に関しては、その応用の信頼性の重要性や事故リスクの重大

性などに鑑み、次に掲げるように適切な品質確保手段を講じることとする。

また、実行時環境が訓練工程の環境と異なる場合（モデル圧縮などを行う場合も含む）には、少なくともテストフェーズにおいては実行時環境と同じ計算（精度やアルゴリズム・ソースコードなど）を再現するソフトウェアを用いて検査することを、本来あるべき姿として本ガイドラインでは推奨する。そのような扱いが困難な場合には、実機でのシステム全体の検査工程において、品質の劣化が許容範囲内であることを追試することが必要と考えられる。

6.8.3　品質レベル毎の要求事項

「リスク回避性レベルについて」

AISL 0.1 → Lv 1 以上

AISL 0.2 → Lv 2 以上

AISL 1 → Lv 3

AISL 2～4 → Lv 3 に追加すべき要求について、今後検討する。

「AI パフォーマンスについて」

AIPL 1 → Lv 1 以上

AIPL 2 → Lv 2 以上

「公平性について」

AIFL 1 → Lv 1 以上

AIFL 2 → Lv 2 以上

- Lv1
  - 利用するソフトウェアについては、信頼できる実績を持つソフトウェアなどを選定し、その選定経緯を記録すること。
  - 選定したソフトウェアについて、その欠陥の発見などを運用期間中モニタリングし、必要に応じて修正などの措置をとること。
  - 学習からテストフェーズに至るまでの環境と、実用段階で用いる環境の相違について、その影響などをあらかじめ検討しておくこと。
- Lv2
  - 利用するソフトウェアについて、検査・実験などによりその信頼性を自己評価すること。

- 可能な場合には、SIL 1 相当のソフトウェア信頼性を得られたソフトウェアを用いること。
- システムの運用期間中のソフトウェアの健全性の維持に関する保守体制を必ず構築すること。
- バリデーションおよびテストフェーズにおいては、原則として実用段階で用いられる計算環境（浮動小数点精度・モデル規模など）を模倣した環境でバリデーション・テストを行うこと。または、テスト済み学習モデルと実用環境での学習モデルの動作の一致性について、何らかの検証を行うこと。

・ Lv3

- SIL 1（またはシステムの要求する SIL レベル）のソフトウェア品質の確認を必ず行うこと。
- 実用環境の計算環境での学習モデルの振る舞いに基づくテスト（または形式検証など）を必ず行うこと。
- また、そのモデルと実用環境での動作の一致の確認を、結合テスト以降の段階で必ず行うこと。

6.9 E-1: 運用時品質の維持性

6.9.1 基本的な考え方

運用開始時点で充足されていた内部品質が、運用期間中を通じて維持されることを「品質の維持性」と称する。システム外部の動作環境の変化に十分に追従できることと、その追従のための訓練済み機械学習モデルなどの変更が品質の不用意な劣化を引き起こさないことの 2 点を包含する。

具体的な実現方法は運用の形態、特に追加学習・再訓練の行われ方に強く依存する。本ガイドラインでは、追加学習・再訓練の形態として、次の 2 つのパターンを想定する。

(a) 追加学習後に、サービス提供中の実システム（運用環境）の外（開発環境など）における品質検査を経て、明示的なソフトウェアの更新などを通じて初めて運用環境に追加学習結果が反映される場合。現在想定されている自動運転などのソリューションは概ねこちらのパターンに属する。

(b) 運用中にリアルタイムに訓練済み学習モデルの更新がおこなれ、運用環境で人手を介

した検査を介することなく更新処理が完結する場合。ストリームデータ処理などで用いられるオンライン学習や、その他にも開発・運用一体の案件での言語認識や会話応用などでもこのパターンが見られる。

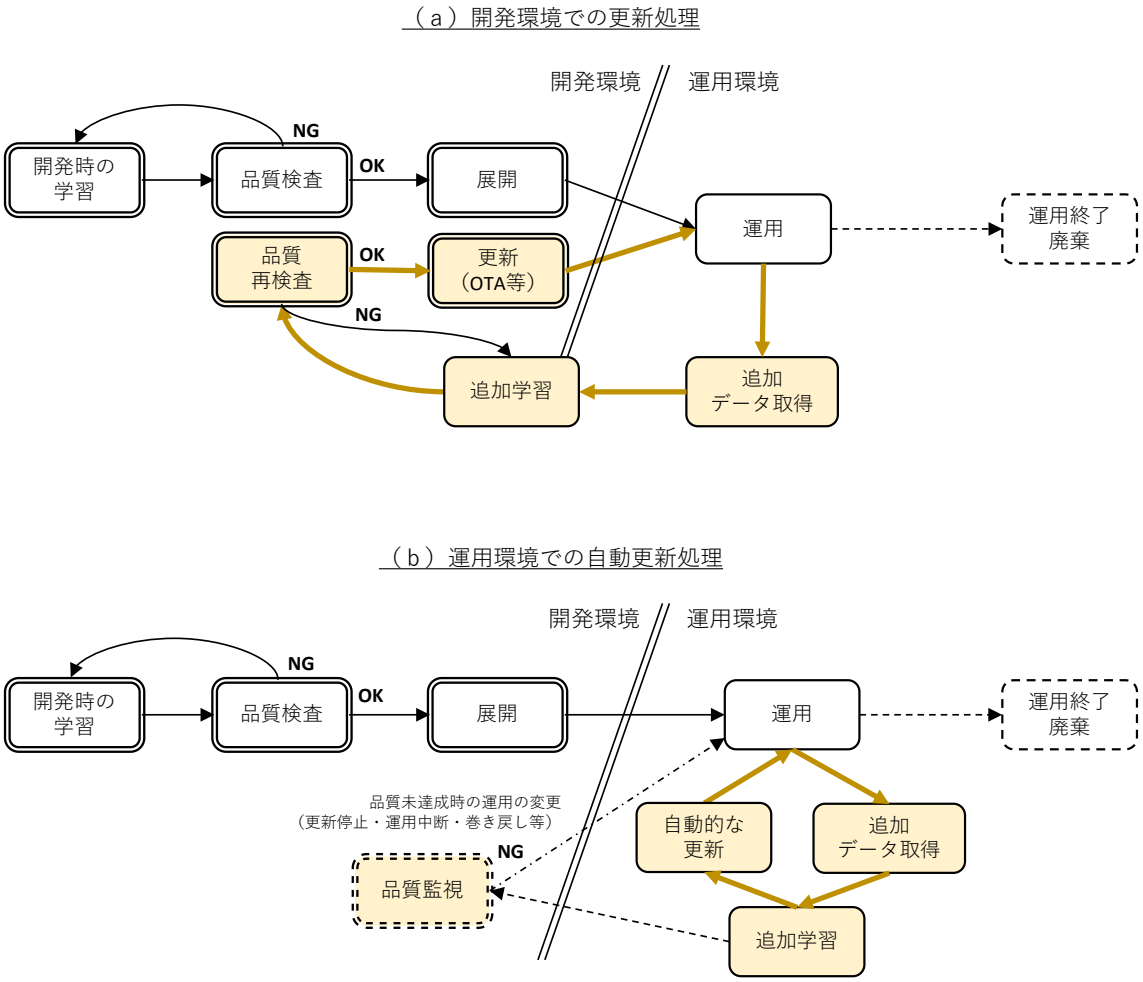

図 15: 機械学習要素の運用中の更新形態

(a) の開発環境などでの更新パターンでは、上図に掲げたように品質検査が必ず更新前に行われ、基本的には初回開発時のテストフェーズと同様の内容で検査を行えば、一定の品質は確保できると考えられる。一方で（b）の運用環境での更新パターンでは、更新そのものは全自動で行われるため、品質が劣化したモデルがそのまま運用に反映されるリスクが高い。このような場合には、品質のモニタリングの仕組みや、劣化した際の対処までを運用体制にあらかじめ組み込んでおくことが重要になる。

また、運用時品質の維持性においては、全体としての性能の劣化の他にも、特定の入力に

ついて、更新前に正答を導いていた機械学習要素が更新後に誤判断を行う[9]問題についても配慮が必要な場合がある。本来は全体として 1.5 節の外部品質が向上ないし維持できていれば必要十分であると考えたいところであるが、実際の産業の現場においては、一旦正しく実装され動作したものが、後日正しく動作しないことについて大きな抵抗感がある。一方で、機械学習システムの特性上追加学習は従来の入力に対して異なる出力値を導くことが必然的であることから、あらかじめ運用方針として取扱いを検討しておく必要があろう。

また、本ガイドラインの直接の対象外ではあるが、追加学習に用いた実環境のデータについての契約やプライバシーなどの法的位置づけも、運用の検討においては配慮しておく必要がある。品質管理の観点からは、追加学習も含め学習訓練に用いたデータは全て保存され、必要に応じて品質検証やモニタリング・上述した過去に正解した事例での性能劣化の確認などのために用いることができることが望ましいが、実際の応用においては、特にプライバシーなどの観点からデータの取扱いに制約が掛かることも想定される。機械学習利用システムの設計時において品質管理の前提としたデータが、運用時に入手できないこととなった場合には、システムの品質担保のロジック全体が崩壊することになりかねない。そのため、入手可能・保存可能なデータの特定は、運用体制の重要な検討要素となる。

6.9.2　具体的な取扱い

システムの更新時に、開発者などによる品質検査が行われるか否かにより、前記した 2 パターンに分けて取り扱う。

（a）更新前に品質検査プロセスを経る更新処理の場合

- 更新頻度の見積もり または 更新の必要性の判断の基準を事前に検討する。
- 運用時に必要となる更新プロセスについて、設計段階で分析と概要の想定を行い、運用開始時までに具体的な手順を設計する。少なくとも、下記のような点に留意が必要と考えられる。
  - ➢ 運用環境から取得可能なデータと、更新を行う開発環境などへの収集・展開方法。

[9] ソフトウェア工学分野ではしばしば「レグレッション」とも呼ばれる。とりわけ本節で述べている内容はいわゆる「レグレッション・テスト」に相当する。しかし、機械学習の分野では同じ意味の英単語を全く違う文脈で用いるため、カタカナの「レグレッション」は全く違う意味をもつ。そのため、本文書では基本的にはどちらの意味でも用いないこととする。

- 追加訓練用データの前処理・フィルタ・ラベル付けの方法。
- 追加訓練・モデル更新時に利用するデータ範囲。特に、時間経過で環境の変化が想定される場合、どのように古いデータを除去するか。

・ 更新時の品質検査の方法、特に更新の可否の判断基準（または意志決定の方法）について、運用開始時までに検討する。
・ 個別の正解事例について品質が悪化する場合の運用上の取扱いについて、応用によってはあらかじめ決定しておく必要が有る可能性がある。

(b) 開発環境での品質検査プロセスを経ない更新処理の場合

・ 追加学習による顕著な品質劣化が起こる可能性と、発生した場合のシステムへの影響波及の範囲について、あらかじめ想定をする。
・ その影響が許容できない可能性がある場合には、追加学習による学習モデルの品質劣化が発生させる、システム全体へのリスクを許容範囲に収める技術的または運用上の仕組みを検討し、運用に組み込む。例えば、以下のような方法が考えられる。
  - 技術的に出力値域を限定する。事前に想定される正常範囲を逸脱しないよう、周囲のシステム構成要素（ソフトウェアなど）で強制する。
  - 運用環境外部からの性能監視により、学習の巻き戻しや、運用の停止・中断などを行う。
・ 運用環境・運用システム内部において、更新前の品質監視ができる可能性があると、品質管理に有用と考えられる（図 16）。

（b－2）運用環境での品質監視の可能性

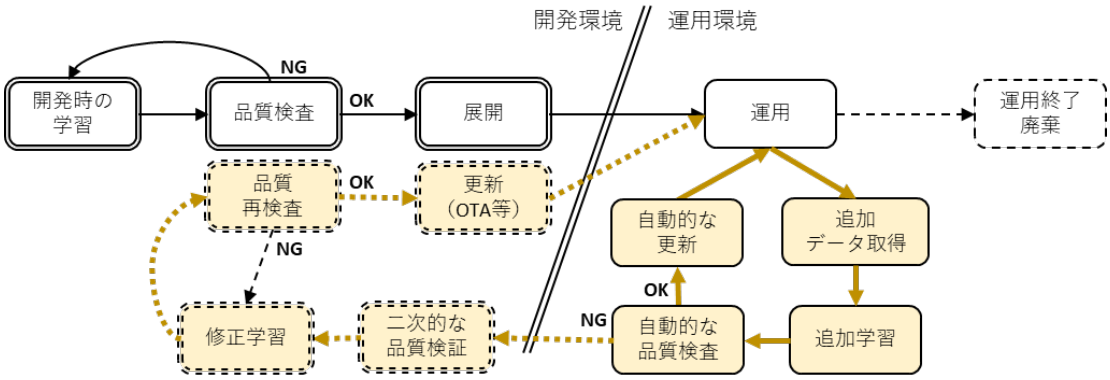

図 16: 運用環境での自動的な品質監視の可能性

6.9.3　品質レベル毎の要求事項

外部特性レベルと本内部特性の要求レベルの対応は以下の通りである。

「リスク回避性レベルについて」

AISL 0.1 → Lv 1 以上

AISL 0.2 → Lv 2 以上

AISL 1 → Lv 3

AISL 2～4 → Lv 3 に追加すべき要求について、今後検討する。

「AI パフォーマンスについて」

AIPL 1 → Lv 1 以上

AIPL 2 → Lv 2 以上

「公平性について」

AIFL 1 → Lv 2 以上

AIFL 2 → Lv 3 以上

上記の内部特性の要求レベルごとの要求事項は以下の通りである。

- Lv1
  - 外部環境変化によりシステムの品質が著しく失われたときの対応について、あらかじめ検討しておくこと。
  - オンライン学習を行う場合には、予想外の品質の低下がもたらす影響についてあらかじめ検討しておき、必要な場合には動作範囲の限定などのシステム的な対応を取ること。
  - オフラインで追加学習を行う場合には、前 7 項に準じた品質管理を行うこと。
- Lv2
  - システムの利用状況が許す範囲において、システムの品質について、動作結果との対照などから品質劣化・誤判断のモニタリングを行うこと。モニタリングにおいては、プライバシーなど製品品質以外の要因を十分に検討すること。
  - オンライン学習を行う場合には、追加学習結果を何らかの方法で定常的にモニタリングすること。モニタリングの結果で性能要求からの逸脱が判明した場合には、直ちに対処を行うことができること。

- オフラインでの追加学習を行う場合には、システム開発段階で用いたテスト用データセットでの「性能劣化の回帰テスト」を行い、更新前に品質が失われていないことを確認すること。必要な場合には、システム開発段階と同等の手法でテスト用データセットの更新を行うこと。

・ Lv3

- プライバシーなどと両立するシステム品質の監視手段を、運用体制を含めて必ず構築すること。
- オンライン学習を行う場合には、追加学習結果をシステムに反映する前に、システム内部で一定の品質確認を行う仕組みを実装し、想定外の品質劣化が無視できない場合には更新を中止する仕組みとすること。また、オフラインでの更新・修正手段を必ず確保すること。
- オフラインでの追加学習においては、運用での収集データと、システム初期構築時のテスト用データセット、および同じ手法で定期的に更新するテスト用データセットを用いて品質を管理すること。

7. 品質管理のための具体的技術適用の考え方

7.1 A-1: 問題領域分析の十分性

本項で必要とされる問題領域の分析（ML 要求分析と称する）は、システムやサービスの利用される実世界に対応して、機械学習要素に入力されるデータの多様性を様々な観点から分析して言語化し、リスクやシステムの要求の差異を機械学習の訓練・テストで用いるデータの属性として具体化する工程である。これは基本的には、従来のシステムズエンジニアリングにおいて行ってきたリスク分析やハザード分析・要求分析の手法と同じ領域の考え方であり、これらの既存領域の取り組みや文献が参考になるところも多い。

従来型のソフトウェア工学の観点ではこれらの工程は、超上流工程の一部として位置づけられながら、定石としての確立した方法論は与えられていない。一方で機械学習の分野の観点では、PoC などを通じてデータサイエンティストが職人芸的なノウハウとして行ってきたことを、品質を説明する為の工程として形式化・知識化することに相当する。

7.1.1 全体的な取り組みの方向性について

システムズエンジニアリングの観点からの分析工程全体の概観としては、「ソフトウェア工学」[93] など一般的な教科書をまず参考にされたい。また、特にリスク回避性について、より詳細化された分析工程の例としては、例えば Causal Loop Diagram や、STAMP/STAP [125] などを事例として掲げられる。これらの分析を通じて得られた結果は、従来では Fault Tree Analysis, Fault Mode and Effects Analysis, Loop Diagram, Feature Tree などの形で具体化されることが多い。本ガイドラインでは特に機械学習要素の構築に用いるデータの性質として具体化することから、これらの分析のいくつかを元に、最終的には Feature Tree を構築することを想定して全体を構成している。

これらの分析を具体化すると、「入力事象（特にリスク要因）の推定」「分析の出力としてのデータの構造の推定」の 2 点が特に大きな問題となり得る。本節ではさらに、この 2 つの観点から機械学習利用システム及び機械学習利用要素に特有の考え方について付記する。

7.1.2　入力側のリスク要因の推定

入力側の実空間などに存在するリスク要因の洗い出しは、基本的には正解のないブレインストーミングであり、その成果の充実度をデータで説明することはなかなか難しく、プロセス的な担保が基本となる。

この概説的な説明として、米国航空宇宙局（NASA）の「NASA Hazard Analysis Process」[53] はコンパクトながら、特に初期段階のハザード分析においてブレインストーミングの起点として着目すべき事柄として、

① 標準ハザードリスト
② 過去の経験・従来システムからの文書類
③ エンジニアリングの訓練と経験

を掲げている。さらに、同文献では①標準ハザードリストの起点として、26 項目の「NASA 一般ハザードリスト」を提示している。これらの中には他の応用では問題にならないようなハザード原因も含まれてはいるが、このリストは特に屋外環境で動作するシステムの機械的動作に伴うハザードの原因としては十分に汎用性が高く、他のシステムの検討においても分析の起点となり得る。

また、②はこれまでの機械学習システムの構築においてデータサイエンティストが行ってきたことと同種のものと考えられる。過去の類似のシステムや、同一システムの前バージョンの分析データは貴重なものであり、積極的に活用すべきものである。加えて、PoC 段階ではこのような出力を意図して構築実験を行うことで、本開発システムの構築に必要な対象の性質理解を進めることが推奨される。さらに③は、いわゆる「ドメイン知識」の活用に相当すると考えられ、機械学習の専門家だけでなくシステム発注側の業務内容に関する知識を盛り込むことに相当すると考えられる。同スライドでは、これらの観点を元に分析を行い、「明確な基準でハザードの分類を判断すべき」とまとめられている。

本ガイドラインでも、基本的には上記①②③の 3 つの観点全てを俎上に上げた検討を行い、さらに PoC 段階において得られた知見を踏まえ、その検討過程と結果をきちんと記録することを推奨する。具体的には、

- ①として、既存の機能安全設計の基礎知識の活用や、応用ごとの既存のハザードリスト、あるいは上記の NASA ハザードリストを元にした「読み替え」のブレインストーミングなど
- ②として、先行システムや過去の機械学習を利用した類似システムの分析事例の活用、

- ③として、企画者（受託開発における開発依頼者）とのブレインストーミングによるドメイン知識の導入、さらに
- PoC 段階での予備的なデータおよび機械学習訓練の試行において得られた例外的ケースなどの知見

の全てを統合し、1 つのリスク要因リストとして整理する。

7.1.3　出力としてのデータの構造の推定

一方で、この工程の出力は機械学習に用いるデータの属性付けであり、次工程である実際の機械学習・訓練プロセスの特徴が、要求として必然的に影響してくる。次工程の訓練プロセスの観点から見た本工程への要求は、「機械学習工程を通じて『別のもの』として扱う属性」を特定することであり、これには

④　出力値やリスクの違いから、「別のもの」と認識しないと困るもの
⑤　入力値や機械学習モデルの特性上、出力値が同じであっても「別のもの」として扱われてしまう可能性のあるもの

の 2 つの側面がある。

このうち④は、システム仕様と前記のハザード分析から主に導出される。出力値の違いは訓練用データのラベルの違いでもあり、またリスクの違う事象はハザードの原因を特定すれば基本的には特定される。

一方で⑤には例えば、画像認識の背景の違いや文字の書体の違いなど、応用ドメインごとに特有の事情があり、また実装上の前処理や学習に用いるネットワークの種類などにも影響される可能性がある。このような分析には、どうしても最終的な実装形態をある程度想定した分析が必要となる。これはソフトウェア工学的には Implementation Forecast と呼ばれる事柄であり、不適切に行うとリスク分析を歪まされる危険性もあるものの、機械学習要素の品質管理においては避けがたいものと考えられる。具体的には、本実装段階を意識した PoC 段階での机上検討と、データを用いた分析が対応すると考えられる。

実際に Forecast を行うべき観点の抽出そのものは、「構築された AI が似た状況で判断を異にするリスク」の観点として考えると、前節で掲げた 3 つの分析上の観点が再び有効であると考えられる。このような観点から、前節の②③及び本節の⑤を合わせて、各応用分野ごとに、ある程度のシステムの分析結果を集積し、分野毎のより詳細化された標準ハザードリストとして整備することが、将来的には望ましいと考える。本ガイドラインの執筆者としても、この観点からの周辺文書の整備を将来の方向性として積極的に考えたい。

7.2　A-2: データ設計の十分性

7.2.1　基本的考え方

前節の分析が十分になされていることを前提にして、本内部品質の目的は 2 つあり、機械学習利用システムが「全く学んだ・試されたことのない未知の状況」に遭遇しないようにするために、必要なデータを訓練時・テスト時にきちんと用意しておくことと、前節で膨大に準備した属性の組み合わせについて、機械学習の構築を現実的にするための「間引き・統合」をシステマチックに行うことにある。

仮に、極めてシステムの解こうとする問題がシンプルで、属性の組み合わせが高々数十程度な場合（例：数字認識で 10 文字×2 書体程度を認識すれば良く、紙質の差異なども考慮しなくて良く 20 通りしか属性組み合わせがない場合）であれば、それぞれの属性毎に認識率を一定以上に与えるデータの量を考え、それぞれを満たすデータの総量があれば、網羅的に十分な訓練を説明することが出来る。しかし実際には従来型のシステムであっても、属性の組み合わせは単純なかけ算では数万通り以上になることも多く、全ての組み合わせのデータを検査用に用意することは現実的でないことが多い。まして機械学習要素においては、属性の組み合わせが細分化された結果として訓練用データや検査用データがそれぞれ 0 件から数件となるようでは、大数の法則が使えず訓練の収束性などの判断が不可能になってしまう。

このような問題への対策として、従来のソフトウェア工学では「組み合わせテスト」と呼ばれる技術があり、これは「網羅性基準」と呼ばれる考え方を用いて、属性の分類の組み合わせの詳細度と、テスト工程における検査項目の充実度の基準を、関連させつつも別々に設定する考え方である。この考え方は従来のソフトウェア構築ではテスト工程での検査用データの評価によく用いられるが、機械学習の訓練工程もまたデータ主体のプロセスであることから、特に有効と考えられる。

実際に組合せテスト技術（*t*-way）を元にした機械学習用のデータセット評価の事例が Barash [43] にも紹介されている。

7.3 B-1: データセットの被覆性

データセットの被覆性は、機械学習の品質管理でも極めて難しい問題で、ここに誤りがあると、判断が安定しなくなる・特定の想定内の状況で判断を誤るなどの結果を招き、公平性の失陥の原因にもなりうる。

既に「要件分析の十分性」において、「言語化できる」状況の差異についてはカバーをしているので、ここではその他の状況の微少な差異をデータ構築時に欠落させないことが求められる。

ライフサイクルにおいては、データの事前準備・データの評価が主に対応するが、テスト段階においても補助的な確認を行うことが考えられる。

7.3.1 データ取得段階における配慮

本項目の基本的な実現手段は、データ準備段階でのデータ取得計画に頼らざるを得ない。

実運用時に想定される状況を偏り無く取得するためには、データを収集する範囲や期間・規模などの計画が不可欠である。具体的な方法は応用ごとにブレインストーミング的に考えざるを得ないが、「要求分析の十分性」の分析の段階で過去の同種の応用の知見を踏まえつつ深くブレインストーミングを行うことで、ある程度の多様性の度合いを事前に検討することができる可能性が高い。

7.3.2 データ整理段階における追加的検査

「要求分析の十分性」の段階で、洗い出した上で属性として採用しなかった属性候補がある場合や、「データセットの十分性」の検討段階で統合され隠れた属性が有る場合には、各ケースの内部で、これらの追加的な属性の分布を確認することは重要である。特に後者の段階で統合された属性に関しては、明確な言語化定義がされている上、場合によってはデータラベルが付与され機械的に確認できることも有り得る。

もちろんブレインストーミングの段階から完全に見落とされていた属性はここでも発見できないため完全な手法ではないが、検討に値すると考えられる。

またデータの種類によっては、前処理段階でデータ固有の特徴量のようなものが得られ、その分布を確認することも有り得る。

7.3.3 テスト段階での追加的検査

そもそもデータの分布自体に問題がある事例で、テスト段階で再確認できることは限られているが、例えば属性として採用した特徴量と、見落としている隠れ特徴量の間に明らかな隠れ相関がある場合には、機械学習モデルの入力値と出力値の間の相関・影響度の分析を行うことで、本来有るべき特徴と異なる特徴を捉えて判断していることが分かる場合もある。安全性などが特に要求される分野においては、このような分析も検討に値する。

7.4 B-2: データセットの均一性

データセットの均一性は、前節のデータセットの被覆性を、入力データ全域を単一の「ケース」として検討することに相当する。7.3 節の各節に掲げた対策が、この場合にも適用できると考えられる。

7.5 B-3: データの妥当性

7.5.1 データの側から見た品質管理のサイクル

本ガイドラインでは主に機械学習モデルの構築の視点からライフサイクルプロセスを検討しているが、データ品質の観点から見ると、データの収集ポリシーと実際のデータの整合性の構築のライフサイクルプロセスも重要となる。データの立場から見ると、

① データ収集ポリシーの決定
② ポリシーに即したデータの収集・選定
③ ポリシーとデータの妥当性検証

の各工程を循環するようなデータ整合性構築プロセスが存在する。

①のデータ収集ポリシーの決定においては、

- 目的の機械学習システムの品質にとって適切なデータ収集ポリシーの策定
- データ妥当性に関する許容範囲の事前検討
- データ妥当性の評価方法の事前決定

が必要となる。②データの収集・選定の工程では、

- 選定・収集方法の適切性の判断
- 実際の選定・収集方法のプロセス管理
- 選定・収集方法に関するエビデンスの収集・整理
- データの出所や追跡可能性に関するメタ情報の管理

を、決められたポリシーを基準に行う。③妥当性検証においては、

- ポリシーに沿ったデータの検証
- ポリシーそのものの説明性の再検討
- 収集データを元にしたポリシーへのフィードバック

が行われ、十分納得のいく妥当性を持つポリシーとデータの組が得られるまで、①に戻ってプロセスが継続される。更にこのようなプロセス全体に対して、

- プロセスそのものの完遂の評価
- プロセスを遂行する担当者のスキルの評価と適切な割り当て
- プロセス全体を管理する仕組みの評価と証跡の確保

が必要とされる。

1.8 節で掲げた開発ライフサイクルとの関係では、このようなデータポリシーの改善サイクルは、抽象的には PoC 工程における試行錯誤の繰り返しとして整理されるが、実際には実開発の作業においても上記の改善サイクルが回りアジャイル的に開発が進んでいくことも有り得る。そのような場合に特に、データ収集ポリシーと合わせて要件定義が更新されていないか、その更新に合わせてここまでの段階（内部品質 A-1〜B-2 など）で確認した内容についての再確認が必要で無いかなどのアセスメントが、全体としての品質確保のために重要である。また、そのようなポリシーや要件の更新について、きちんと変更管理がなされることも、後から品質を再確認・検証する際に極めて重要と考えられる。

7.5.2　外れ値とコーナーケースの整理に関する技術的支援

外れ値、あるいは対応すべきレアケースとしてデータ妥当性の判断を要するデータを効率的に抽出し整理するために、いくつか適用可能な技術がある。

まず、DeepXplore [98] では Neuron Coverage の技術をコーナーケース抽出に用いている。また、Surprise Adequacy [71] は機械学習モデルの入力に対する「驚き（surprise)」を指標化することで、やはりコーナーケースを入力値の中から抽出することができる。Tinghui et. al. [96] はこれらの技術のうち Distance-based Surprise Adequacy を改良して

コーナーケース解析を行ったものである。このようにして抽出されたデータが、外れ値として除去されるべきであるか、あるいは重要なレアケースとして取り扱われるべきかは、ポリシーと照らし合わせて個別に判断することになる。

さらにこのような手法を紹介した文献として、Zhang [122] の Sec. 5.3 や、Dong [55] の Sec. 2.3 も参考になる。

7.6 C-1/C-2: 機械学習モデルの正確性・安定性

機械学習モデルの正確性・安定性の検査は、従来のソフトウェア開発における「単体テスト」などのソフトウェア・テスティングに対応する作業と考えられる。本節では、

- ソフトウェア・テスティングの技術 [41] を、機械学習モデルを含む「推論エンジン」全体に適用する際に、機械学習要素が持つ特徴によって考慮すべき側面と機械学習要素テスティングの技術動向、
- 安定性を直接的に検査する技術の可能性

を紹介する。

7.6.1 機械学習要素におけるソフトウェア・テスティング

7.6.1.1 検査対象の特徴

第 1 に、推論結果の品質を議論する際に検討すべき対象が、訓練用データセットを入力し訓練済み学習モデルを求める訓練学習プログラムと、得られた訓練済み学習モデルが機能振舞いを規定する予測・推論プログラムの 2 つに分かれていることが挙げられる。第 2 に、計算結果が正しいか否かを判断する基準が明らかでなく、オラクル問題 [44] への対応を考える必要がある。訓練学習プログラムは、その計算結果（訓練済み学習モデル）の正解値を予め知ることが困難である。また、予測・推論プログラムが導く答えは確定的ではなく確からしさを伴う相対的な値であり、正解の絶対的な基準を定義することが難しい [56]。機械学習プログラムはテスト不可能プログラム [117] に分類される。

7.6.1.2　テスト・オラクル問題

テスティング技術は、テスト入力生成方法とテスト実行結果の確認方法からなる。第１に、前者は、検査の目的を効率よく達成するテスト入力を、如何にして生成するかが課題である。検査の対象となるプログラム範囲を定量的に表す指標を基に、検査済みの範囲を示す検査網羅性指標を利用して、新しいテスト入力を得る方法が知られている。後に紹介するように、機械学習要素のテスティング法として興味深い研究が多数ある。

第２は、与えたテスト入力に対する既知の正解とプログラム実行結果を比較する方法に関連し、テスト・オラクルの技術と総称される。テスト不可能プログラムは、与えたテスト入力に対する正解が既知でないプログラムであり、従来から、導出オラクル・疑似オラクル・部分オラクル等と呼ばれる方法が用いられてきた。

部分オラクルの代表的な方法にメタモルフィック・テスティング（MT）[49] がある。数値計算あるいはコンパイラを含むトランスレータなど、入力に対する計算結果の正解値を予め知ることが難しいプログラムの検査方法として考案され、その後、暗黙のオラクルを用いるシステム・ソフトウェアやセキュリティ・システムの検査に適用された [50]。現在では、機械学習要素に対する標準的なテスティング方法論になっている [89]。また、検査目的によっては、部分オラクルと統計的なテスティングを組合せる方法が有用である [91]。

7.6.1.3　テスティングの２つの観点

一般にプログラムの品質を確認する際、適宜、正常系テスティング（Positive Testing）と例外系テスティング（Negative Testing）を実施する。

正常系テスティングは、検査対象が想定した機能振舞いを示すことの確認である。ここでは、説明の都合上、プログラムが事前・事後条件を伴うとしよう。「事前条件を満たす時、プログラム本体実行後の状態で事後条件を満たす」という関係が成り立つ。事前条件を満たすテスト入力データに対して、上記の関係が成り立つ時、検査対象プログラムはテストに合格したとする。

例外系テスティングは、想定外の状態でプログラムが破滅的な不具合を生じないことを確認する。一般に、プログラムは事前条件を満たさない入力に対しても、妥当な振舞いを示すことが期待される。暴走して異常終了するようなことがあってはならない。実行結果が異常終了を導くか否かを調べる時は、実行結果の正しさの基準を必要としない。このようなテスト・オラクルを暗黙のオラクル（Implicit Oracles）と呼ぶ。

従来から、システム・ソフトウェアやセキュリティ・システムなどのオープンなソフトウェアの統合テスティングでは、条件付きランダム生成したデータをテスト入力とするファズ・テスティング（Fuzz Testing）の方法［86］が知られている。この方法は、例外系テスティングでも有効である。テスト対象ごとに適切なファズ（Fuzz）を生成する。

機械学習要素を対象とするテスティングでは、正常系テスティングは正確性の検査に相当する。訓練データと同様の統計的な特徴を持つデータをテスト入力とすれば良い。一方、例外系テスティングは、汎化性能に着目する場合の正確性の検査、訓練時に想定しなかった入力に対する振舞いを調べる安定性の検査を含む。安定性の検査は、入力として、欠損データ（Corrupted Data）や敵対データ（Adversarial Examples）を対象とする場合を含む。後述するように、機械学習要素のテスティングでは、検査の目的・検査の観点に応じて、さまざまなファジングの方法が提案されている。

7.6.1.4　評価メトリックス

機械学習要素を対象とするテスティングでは、訓練済み学習モデルにテスト入力した時、この入力データを起点として活性状態になった学習モデルの範囲を知ることが重要になる。研究初期に、活性ニューロン個数の割合によって規定するニューロン・カバレッジ（Neuron Coverage, NC）［98］が提案された。その後、NC よりも詳細な情報を扱うことを目的として、異なる層の活性ニューロン間の相関などの活性ニューロンを対象とする構造的なパターンによって定義する方法が提案された［82］。これらの構造的なカバレッジ値が大きければ、学習モデルの広い範囲を活性化するとみなせる。そこで、検査範囲が十分と解釈し、そのようなテスト入力が有用であると仮定している。

一般に、機械学習要素では、訓練データの統計的な性質が明示されないことから、例外系テスティングに用いるテスト入力を定義することが難しい。不意の適切さ（Surprise Adequacy）［71］は、テスト入力が訓練データの統計的な特徴から逸脱していることを確認する方法として提案された。テスティングという目的から導入したものであるが、データの外れ度合いを表す指標の一種である。この指標を内部活性状態によって定義している点が重要である。

7.6.1.5　テスト入力の自動生成

一般に、検査対象ごとに適切なテスト入力データを生成する方法が重要となる。また、正

常系テスティングあるいは例外系テスティングの目的に応じて、どのような方法が適切かは異なる。対象が訓練学習プログラムの時、その入力はデータの集まり（データセット）であることから、データセット多様性（Dataset Diversity）[87] の考え方が、正常系ならびに例外系テスティングのテスト入力生成方法論に指針を与える。

予測・推論プログラムのテスティングでは、訓練学習時に想定していないような多様なデータを入力し予測結果を確認する。つまり、テスト時補完（Test-time Augmentation）と呼ばれている手法である。実際、機械学習の技術を応用してデータ生成を行う方法が試みられている。画像データを対象とする従来のデータ補完（Data Augmentation）の方法 [75]、敵対生成ネットワーク（Generative Adversarial Networks, GAN）による方法 [61] がある。また、敵対データの生成法を応用するアプローチがあり、最適化問題に帰着する方法 [92]、誤差関数の勾配を利用する方法 [124] などがある。なお、検査データを多数生成する場合には、前述したデータセット多様性が基本的な考え方として有用である。

7.6.1.6　テスティングと評価メトリックス

テスト入力生成に際して、検査の目的・検査の観点から見て、検査を効率良く行える有用なデータを求めたい。従来のソフトウェア・テスティング技術と同様に、評価メトリックスを応用して、例えば、構造的な網羅性基準が向上するようなデータを生成する。この方法は、カバレッジに基づくテスト生成（Coverage-Guided Test Generation）と総称される。機械学習要素を検査対象とする時、前述した NC を用いる研究事例が多い。古典的なデータ補完によるデータ生成法との組合せ [109]、GAN によるデータ生成法との組合せ [123] がある。

一方、検査の目的・検査の観点によっては、構造的なカバレッジが有用ではない、という考察がある [63]。例えば、NC と敵対ロバスト性の相関が小さいことが報告されている [114]。むしろ、NC の値は、不具合の状況よりも、モデル・キャパシティとの相関が大きい [129, 第 4 章]。例えば、NC が小さくても正解率が高くなったり、逆に、正解率が悪くても NC 値が大きくなったりすることがある。なお、従来のソフトウェア・テスティングの技術においても、構造的な検査網羅性指標が、テストスーツの有効性あるいは欠陥発見の検査効率と相関が小さいという実験結果が報告されている [65]。見方を変えて、NC をニューロンの活性状況を表す簡易指標とみなすことができる [88]。実際、NC あるいは NC の集まりから求めた統計指標を検査指標に用いる実験が報告されている [129][91]。

機械学習要素では、従来のプログラムに比べて、不具合の原因となるコーナーケースとテ

スト入力の関係が極めて弱い。そこで、構造的な網羅性基準に代わるメトリックスを考える必要がある。たとえば、誤差関数を用いた方法を評価メトリックスに利用する [114]。

7.6.2　安定性の評価と向上に関する諸技術

図 17 に示す安定性の評価と向上に関する技術を、6.7.3 の確認要求事項のレベルを参照しながら説明する。

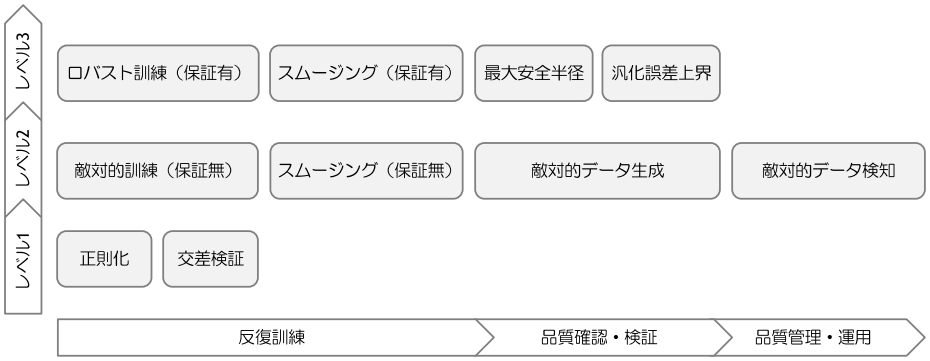

図 17　安定性の評価・向上に関連する技術（技術を適用する工程とレベル）

7.6.2.1　交差検証（cross validation）

交差検証は、データセットを訓練用、バリデーション用、テスト用に分割することによって、訓練時のデータセットへの過剰な適合を抑える訓練法であり、広く利用されている [72]。例えば、K-分割交差検証では、訓練用のデータセットを K 分割し、$1/K$ のデータセットをバリデーション用、残りの $(K-1)/K$ を訓練用として、訓練用とバリデーション用のデータセットを入れ替えながら訓練する。レベル 1 での適用が推奨される技術である。

7.6.2.2　正則化（regularization）

正則化手法とは、訓練時に学習するパラメータが過剰に大きくなることを抑える方法である [94]。例えば、訓練時に最小化する損失関数に正則化の項（学習パラメータの絶対値とともに大きくなる項）を追加することによって、訓練用データセットへの過剰な適合（過学習）を抑えられる。また、最近ではドロップアウトという正則化手法も使われている [107]。ドロップアウトでは、訓練中にランダムにニューロンを訓練対象から除外することによって、複数の機械学習モデルを同時に訓練する場合と同様の効果を得られる。安定性が必要な

場合、レベル 1 での適用が推奨される技術である。

7.6.2.3　敵対的データ生成（adversarial example generation）

敵対的データは、判定を誤るように摂動を加えたデータである。機械学習モデルでは、人間には認識できないようなわずかな摂動でも誤推論することが知られている。そのような敵対的データを生成する様々な手法が提案されている [98][47]。敵対的データを生成するために必要な摂動が大きいほど安定していることを意味するため、その摂動の大きさは安定性の評価にも利用されている。このような敵対的データを生成するための実用的なツールはまだ整備されていないが、将来的にはレベル 2 の評価に適用可能な技術である。

7.6.2.4　最大安全半径（maximum safe radius）

最大安全半径とは、元データとその敵対的データの距離の最小値である。7.6.2.3 の敵対的データ生成では近傍の敵対的データを探索するが、最大安全半径では近傍（最大安全半径の超球の内側）に敵対的データが存在しないことを保証できる。最大安全半径を正確に求めるために形式手法（ソルバー）を用いた手法が提案されている [69][110]。しかし、正確な最大安全半径の計算コストは非常に高いため、計算可能なネットワークのサイズ（ニューロン数）に制限がある。そこで、最大安全半径よりも小さい安全半径の近似計算法も提案されている [116][45][119]。また、100%ではないが、確率的に安全性を保証する半径を近似的に計算する手法も提案されている [115]。これらはまだ研究段階の技術であるが、最大安全半径の超球の内側には敵対的データは存在しないことを保証できるため、将来的にはレベル 3 での適用が期待される技術である。

7.6.2.5　汎化誤差上界（generalization error bound）

汎化誤差とは、（特定のデータセットに限らず）全入力データに対する不正解率の期待値である。一般に入力データは無数に存在するため、その不正解率を計測することは困難であるが、「汎化誤差が e %以下であることを確率 p %で保証する」ことができる汎化誤差の確率的な上界を見積る技術が提案されている [112]。現在のところ、まだその見積り精度は高くなく、研究段階の技術であるが、汎化性能を評価するひとつの指標として、将来的にはレベル 3 での適用が期待される技術である。

7.6.2.6　敵対的訓練/ロバスト訓練（adversarial training / robust training）

敵対的訓練は誤推論しそうな近傍データを探索しながら訓練を行う手法である [84]。通常の訓練と比較して、機械学習モデルの敵対的データに対する耐性を向上させることができる。このような訓練を行うための実用的なツールはまだ整備されていないが、将来的にはレベル 2 に適用可能な技術である。

ロバスト訓練は近傍に敵対的データが存在しないように訓練を行う手法である [118]。最大安全半径の近似計算法もセットで与えており、訓練用データセットの各データの近傍に敵対的データが存在しないことを保証できる。まだ研究段階であるが、将来的にはレベル 3 での適用が期待される技術である。

7.6.2.7　ランダムスムージング（randomized smoothing）

ランダムスムージングは、ランダムノイズを付加したデータをニューラルネットに入力したときの出力の期待値を推論結果とする手法であり、ランダムスムージングによって安定性が向上することが報告されている [81]。入力データの近傍に対するランダムスムージングの推論結果の正しさを保証するためのランダムノイズの付加方法も提案されている [77][52]。出力の期待値を平均値で近似するために複数回の推論が必要であることや、ランダムノイズを大きくすると安定性は向上するが正確性は低下するトレードオフがあることなどを考慮する必要があるが、推論結果の安定性を保証するために、レベル 3 で適用が期待される技術である。

7.6.2.8　敵対的データ検知（adversarial example detection）

運用時に敵対的データを検知する手法が提案されている [120][83]。まだ研究段階であるが、機械学習モデルに潜在する敵対的データを防御するために、将来的にはレベル 2 での適用が期待される技術である。

7.7 D-1: プログラムの信頼性

7.7.1 基本的な考え方

プログラムの信頼性は、通常のソフトウェアでも重要な項目であり、特にオープンソースを含む多数のライブラリを混用して用いる機械学習 AI の実装においては品質の管理が困難になることが想定される。オープンソース・ソフトウェアは無保証が基本であり、最終的な利用者との関係では、誤動作の差異の責任の所在だけでなく、潜在的な誤りの発見や監視、場合によっては修正までが、オープンソース利用者である開発者・運用者の責務になると考えられる。

さらに、機械学習モデルの構築においては、バグ（プログラムの誤り）の存在を訓練プロセスが織り込んでしまい、テストなどで表面的にバグの影響が現れないまま、品質の劣化が確認されることも判明している [90]。

一方で、通常のソフトウェアとの関係では、ライブラリや基盤ソフトなどの構成管理や品質のモニタリングは特にセキュリティ関係で重要視されており、そのための基盤やサービスが充実しつつある。そのサービスのもつデータの中には、限定的ではあるが、機械学習特有のライブラリなどの情報が含まれているものもある。機械学習応用においても、これらの存在する基盤は活用する価値があると考えられる。

7.7.2 オープンソース・ソフトウェアの品質管理

各事業者はオープンソースライブラリをどの程度信用し、その品質を自らメンテナンスするかについて、考えておくことが望ましい。

必要な品質のレベルと関係して、必要な場合には、品質の担保されたサポート付きのライブラリや、自社での品質検査プロセスを経たソフトウェアを使うようなことも想定される。

7.7.3 構成管理とバグ情報の追跡

ソフトウェア要素の構成管理については、例えばセキュリティ分野でシステムの構成要素を列挙するための共通 ID としての Common Platform Enumeration (CPE) [35][127] があり、これらをベースとして、利用しているソフトウェア部品のバージョンを管理し更新な

どの情報を抽出する商業製品なども存在している。これと関連する脆弱性情報リスト Common Vulnerability Enumeration (CVE) [36][128] には、セキュリティ脆弱性に直結しないバグが列挙に含まれていないが、少なくとも CPE に関係したツールは、ライブラリおよび基盤ソフトウェアの最新版の追跡に有益である可能性が高い。

7.7.4 テスティングによる具体的な確認の可能性

また、従来のソフトウェアと異なり機械学習要素においては、構成するソフトウェアにバグがある場合、そのバグが実際の機械学習結果に直接の影響を及ぼすとは限らない。特に、学習訓練のフィードバックループの中にバグを含むソフトウェアが置かれている場合、訓練済み学習モデルがそのバグの振る舞いを「覚えてしまう」ことにより、結果的に訓練が一見してうまく行ったように見える場合が存在する。このような場合において、7.6.1 節に掲げたメタモルフィックテスト技術を用いた統計的な振る舞いの分析により、ソフトウェア自身のバグによる潜在的な品質劣化を見付けられることがあることが報告されている。

7.7.5 ソフトウェア更新と性能・動作への悪影響の可能性

一方で、ソフトウェアライブラリの開発者が性能や動作の改善を意図した更新を行った場合でも、実際の複雑な構成のソフトウェアにおいては、却って動作が意図しないものとなる場合も多い。機械学習システムの構成によっては、バグの振る舞いが訓練過程において順応・学習されてしまうこともあり、ソフトウェア構成部品を更新した際には、動作及び性能の再確認が欠かせない。場合によっては訓練をやり直す、または脆弱性などの影響を考慮した上で古い版を使い続けるなどの判断も必要である。

具体的な判断はその状況に依存するため一概にガイドラインで推奨はできないが、このような場合に「きちんと品質を管理している」と主張するためには、その判断の経緯をきちんと記録しておき、説明可能にしておくことも重要となる。

7.8 E-1: 運用時品質の維持性

本節では、運用開始時点で充足されていた内部品質を、運用期間中を通じて維持するため

の技術について述べる。運用時に発生しうる外部環境の変化に対し、運用開始時点で実現されていた内部品質を維持するためには、機械学習要素を変化に対して追従させる必要があり、その運用形態は 6.8.1 節に述べたとおり、必要に応じたタイミングで開発環境に戻ってデプロイしなおすという一括処理的な更新を行うパターンと、運用環境にて随時または高頻度に自動的に更新を行うパターンの 2 つがある。前者のパターンでは、いつ更新を行うべきかの判断するためのモニタリングと、更新処理が主要素となり、後者のパターンでは、自動更新処理が主要素となり、実現にはオンライン学習［103］などが採用される。自動更新を行う場合にも、自動更新が正常稼働しているかのモニタリングと、正常稼働状態を逸脱した場合の対処としての更新処理は必要となる。

ここでは、いずれのパターンでも必要な、モニタリング技術について紹介し、特にコンセプトドリフト［60］と呼ばれる、データ分布が時間経過に伴い変化する現象を検知する技術について紹介する。そして、訓練済み機械学習モデルの更新処理の中核となる再学習のための技術と、そこで用いられる追加の学習データの作成技術について紹介する。

7.8.1　モニタリング

運用時においては、外部環境の変化などの様々な要因によって、新たに取得した入力データと出力（推論）結果との関係が、学習に用いた訓練データの同関係から変化している場合がある。そのような入出力関係が変化したデータに対し、設計・開発時に学習させた訓練済み機械学習モデルを運用時においても使用し続けた場合、パフォーマンス（精度）が低下し、重大な損害が生じる恐れがある。それゆえ、運用開始時点で充足されていた品質を、運用期間を通じて維持しているかを調べるために機械学習システムや機械学習要素の振る舞いを継続的にモニタリングする必要がある。

運用時のモニタリングタスクとしては、以下の 4 つが挙げられる。

- 精度モニタリング
- KPI モニタリング
- モデル出力モニタリング
- 入力データモニタリング

「精度モニタリング」は、訓練済み機械学習モデルの精度を直接測定する。精度モニタリングは、精度の計算に必要な訓練済み機械学習モデルの推論結果に対する正解収集方法に従って、いくつかのパターンに分かれている。即ち、推論のあと、(1) 一定期間後に自動で正解が手に入る場合、(2) 自動で正解が手に入らず、人力でラベル付けを行う必要がある場

合、そして、(3) 人力のラベル付けがコストに合わない、またはラベル付けが不可能な場合である。それゆえ、精度モニタリングを適切に行うためには、上記の正解収集方法の場合分けに従って、適切なモニタリング手法を選択する必要がある。またアプリケーションによっては、モデルの単なる精度だけではなく、コンバージョン率などユーザ利益に即した KPI の観点での監視、即ち「KPI モニタリング」が重要視される場合がある。そのような場合においては、モデルの精度と KPI の一貫性にも留意してモニタリングを行うことが必要である。

続いて、「モデル出力モニタリング」および「入力データモニタリング」は、それぞれ訓練済み機械学習モデルによる推論結果またはその入力データの監視を表しており、その監視方式は、人力監視（全数、サンプリング）、自動監視（アラート条件が既知）、フィルタリング（アラートの可能性が高い条件が既知）に分類される。特に、モデル出力モニタリングにおいては、人力監視において、医療診断など出力系に専門家の確認が推論毎に必ず入る場合と、ある一定期間後にまとめて事後確認する場合に細分化される。同様に、フィルタリングによる監視においても、偽陽性は許容できるが、偽陰性は許容できないなど、様々な条件が付随する場合がある。加えて、入力データモニタリングにおいて、フィルタリングによって監視する場合にアラートを出すか、入力を捨てるかは、アプリケーション依存となる。

7.8.2　コンセプトドリフト検知手法

コンセプトドリフトは運用時に訓練済み機械学習モデルが精度低下を引き起こす主な原因の１つであり、近年、様々な監視（検知）手法が提案されている。コンセプトドリフト検知手法は、運用時に取得したデータに関する正解ラベルの使用の有無に従って、表 4 のように分類される [101]。

表４　コンセプトドリフト検知手法の分類と特徴

正解ラベルの使用の有無	手法	特徴
ラベルあり検知（教師あり検知）	Sequential analysis	Accuracy などの絶対値の監視
	Statistical Process Control	エラー率の増減の監視（予兆発見による早期検出）
	Window based distribution monitoring	学習時と運用時の分布のずれの監視

ラベルなし検知 （教師なし検知）	Novelty detection / clustering method	クラスタリングによる簡易検知
	Multivariate distribution monitoring	データの分布に対して統計的仮説検定などを適用して検知
	Model dependent monitoring	モデルに依存する出力。確信度(confidence)スコアなどを利用

特に近年においては、ラベルなし検知手法に関する研究が盛んに行われており、例えば Faria [58]は、k-means 法などのクラスタリング手法に基づいたマルチクラス問題における未知クラス検出アルゴリズム（MINAS）を提案している。また、Reis [100] は、サンプルが同じ分布から発生しているか否かを判断するノンパラメトリック検定手法であるコルモゴロフ－スミルノフ（Kolmogorov-Smirnov, KS）検定を、逐次化することで計算量を改善したインクリメンタル KS 検定のアルゴリズムが提案されている。さらに、 Lindstrom [80]では、訓練済み機械学習モデルの出力に付随する確信度スコア（confidence）を用いることで、正解ラベルを用いずにデータの変化を検知する手法（CDBD）が提案されている。

7.8.3　再学習

前述のモニタリングによってデータ分布の変化や訓練済み機械学習モデルの精度低下が検知された場合、直近のデータを訓練用データに追加または既存のデータと差し替えたデータセットを用いて機械学習モデルを再訓練する必要がある。この再学習に関する研究も盛んに行われており、例えば、Tian [108] では、既存の機械学習フレームワークに対し、モデル更新の適切なタイミングを自動で判断するためのプラットフォーム設計方法が提供されている。また、破滅的忘却（catastrophic forgetting）と呼ばれるニューラルネットワークの再学習による従来タスクの忘却を軽減するために、「既存のモデル」と「新たなタスクを学習させたモデル」の両パラメータのバランスをとったパラメータを、再学習後のモデルに適用する手法も提案されている [78]。

7.8.4　追加の学習データ作成

正解ラベルが自動収集できないケースは実用上よくみられるが、この場合は、運用時に新

たに取得したデータに対して手動で正解ラベル付けを行う必要がありコストが高い。この問題に対し、GUI (Graphical User Interface) を備えたソフトウェアを利用してラベル付け労力を軽減する方法 [113] や、アクティブ・ラーニング(能動学習)により、ラベル付けを行うデータ数を減らし再学習のコストを削減する研究 [102] などが、提案されている。

8. 公平性に関する品質マネジメントについて

本章では、公平性に関する品質マネジメントについて、社会的な背景や問題の整理、公平性に特有の技術的課題などを分析するとともに、内部品質特性として留意すべき事柄を整理する。

8.1 背景

8.1.1 社会的要請と社会原則

8.1.1.1 AI 社会原則等

人工知能に対しては近年、とくに「倫理性」(ethicalness, ethics 倫理）に関する懸念や社会における規範のあり方の議論がなされている。このような状況に対応して、1.9 節で紹介した AI 社会原則の文書類はいずれも、公平性や倫理性に関する要求を項目の 1 つとして明確に掲げている。総合イノベーション戦略推進会議による『人間中心の AI 社会原則』[20] では、第 6 項に「公平性、説明責任及び透明性の原則」として、「AI の設計思想の下において、人々がその人種、性別、国籍、年齢、政治的信念、宗教等の多様なバックグラウンドを理由に不当な差別をされることなく、全ての人々が公平に扱われなければならない。」として、設計段階からの公平性への配慮を求めている。また、OECD の AI 原則［21］は、AI システムは人権や多様性などに配慮した設計と適切な「セーフガード」の仕組みを含むことを求めている。さらに、この OECD の AI 原則を基に、EU AI 白書［24]や欧州 AI ハイレベル専門家グループの AI 透明性ガイドライン［26］などでも触れられ、原則レベルではコンセンサスが得られつつあると考えられている［57][29]。

8.1.2 AI ガバナンス

経済産業省のレポート［29］によれば、AI ガバナンスとは「AI の利活用によって生じるリスクをステークホルダーにとって受容可能な水準で管理しつつ、そこからもたらされる正のインパクトを最大化することを目的とする、ステークホルダーによる技術的、組織的、

及び社会的システムの設計及び運用」のことを指すとされる。前節に例示した原則との関係では、原則類は概念的で技術中立的・分野横断的な目標の提示であり、AI ガバナンスはそれらを具体的に、法的拘束力がある規則、法的拘束力がないガイドライン、国際標準といった様々な拠り処に基き、開発当事者側や監視や規制執行側の活動によって実現することを目指す活動を指す。

8.1.2.1 法的拘束力のある規則

公平性については、アメリカの法制度整備に長い歴史があり、いくつかの概念のもととなっている。半世紀以上前に、Civil Rights Act of 1964（米国公民権法）人種や宗教、等による差別を禁じる法律が制定された。この中の Title VII では雇用における差別について述べている。さらにその後、住宅に関する Fair Housing Act of 1968 など、分野ごとに人種等の属性による差別を禁じる法律も制定され、これらの法律の適用を通じ、後述する Disparate impact と Disparate treatment という、二つの重要な法的概念が確立されてきた。しかしながら、これら昔からの法律は、現在の AI システム技術特有の問題は扱っていない。2019 年の Facebook の提訴 [10] も AI に限定しない技術中立的な法律に基づくものである。

一方で、2021 年 4 月には EU から AI 規制枠組み規則案［22］を含む AI に特化した政策パッケージが発表された。専門家の間では、特定分野の規則ではなく「横断的な」法的規制は、AI によるイノベーションへの重要な足かせになりかねない面があるとして活発な議論が続いてきたが、この EU の規則案は、AI システムを利用目的に応じ分類する現実的なリスクベースアプローチをとっている。例えば、公平性と深く関連する基本的権利保護と相反しかねない AI は「unacceptable risk AI」と分類され原則禁止となる。また交通機関や社会システムなど安全性要求が高い AI などは「high-risk AI」に分類され、開発や運用に所定の手続きが要求される。今後審議を経て正式決定された場合、この規則は GDPR 同様、EU 域内の AI システムの提供事業者とその利用者だけでなく、AI システムのサービスが EU 域内で利用される場合には、EU 域外の AI システム提供事業者や利用者にも適用されると想定される。

[10] 2019 年 3 月、米住宅都市開発省が Facebook のターゲティング広告が人種等の特徴に基づいて特定の層を広告配信対象から外していたことを公正住宅法違反で提訴。

8.1.2.2　法的拘束力の無いガイドライン

一方で、「法的拘束力のないガイドライン」は、

① 法的規制が存在する場合に、その要件を実際に満たすための手段をチェックリストなどの形で具体化するため

② 低リスク分類など、法的規制が適用されない場合において、提供する AI 製品・サービスの品質を担保するため

の 2 つの位置づけを持つ。

例えば、 欧州ハイレベル AI 専門家グループによる The Assessment List For Trustworthy Artificial Intelligence (ALTAI) [27] は、公平性について「多様性、非差別、公平性」要求とし取り上げ、「不公平なバイアスを避ける」趣旨として、例えば以下のチェック項目をあげている。

- データ使い方、アルゴリズム設計両面において、不公平な偏り避けるための戦略をたてたか？
- ユーザやデータの多様性や代表性について検討したか？
- 選択した “fairness”[11] は適切か？（他の定義も考えたか？コミュニティに相談したか？など）

本ガイドラインも同様に、上記①、②の位置づけを持つ。

8.1.2.3　国際標準

ISO・IEC、IEEE などにおいて、AI の倫理性・透明性・公平性などに関するレポートや標準の策定が進められている。これらについては 10.2 節に整理する。

8.1.2.4　その他

公平性に関する実現は、本質的な「多様性の実現」でもあり、開発プロセス当初から様々なコミュニティを巻き込んだ「包摂的な開発、設計および展開」が、さらなる社会的損害を防ぎ、既存の社会的不平等を軽減するのに役立つ可能性があるとされる。この「包摂性」も含め、人間中心の責任ある AI 開発と利用を実現するための国際的マルチステークホル

[11] ここでの fairness は、本ガイドラインでは “fairness metrics” と呼んでいる概念に対応すると考えられる。

ダーイニシアチブ（Global Partnership on Artificial Intelligence）が 2020 年 6 月に設立された。

8.2　本ガイドラインにおける「倫理性」と「公平性」

「倫理性」「公平性」のような語の使い分けについて、現時点では社会・技術領域ともに十分に合意された統一的な考え方は得られていない。本文書では倫理性を、社会学の領域の性質として、「実社会の規範に照らして良い振る舞い（well behavior）を行うこと」と定義する。この倫理性は公平性に限らず、特定の形態の判断の抑止やある種の安全性、ジレンマのある判断についての方針なども包含されると考えられる。一般にシステムに対する倫理性の要求は暗黙的には社会が提示するものであり、必ずしも明示されない社会規範から導出され、主にシステムの企画・設計段階において、要件定義への制約および検討材料となる。

一方で公平性 fairness は本文書では、「定義された要件以外の入力の状況の差異に起因して、定義された何らかの基準で異なる取扱いをされないこと」と狭く定義する。社会規範レベルの議論では、この定義の公平性は倫理性の一部であると考えられる。

更に、本文書が実現する機械学習要素の公平性は前に述べたとおり、エンジニアリングのレベルの性質として、具体的な公平性担保の対象となる識別要件とセットで、倫理性や機能性などの要請から要件定義の段階で発見され、要件定義から具体的な要求としてシステム提供者が導出するものと（狭く）定義する。

8.3　公平性の難しさ

8.3.1　要求の多様性

前節において公平性の本文書における一般的な定義を提示したが、「同等に扱われていること」を説明する・実現する・納得する観点は複数あり、単に「公平であること」というだけでは具体的な取扱いに困る場合も多い。それぞれのシステムに応じ、より踏み込み数式化可能な形で定義して初めて公平なシステムを開発が可能となるが、そうした導出を行うためには、公平性要求の重要視点について理解する必要がある。

本節では参考として、アメリカ合衆国の雇用均等法制において、防ぐ差別の対象として定

義されている次の 2 つの観点に着目する。

A）異なる取扱い型差別（Disparate treatment discrimination）

B）異なる効果型差別 （Disparate impact discrimination）

「異なる取扱い型差別」はプロセス自体に何等かの不公平な処理があり、いわば意図的に公平性が損なわれている場合を指しており、雇用プロセスにおける基本的な要求として考えられている。

一方で「異なる効果型差別」は、マイノリティへの不利な事象の防止など、雇用結果に直接の公平性を求められる場合である。結果としての公平性の定義は、目標とする指標を用いて詳細に定める必要があり、例えば採用における男女差についても「人数を同じにする」「労働人口比に対して採用人数比を均等にする」「応募人数比に対して採用率を均等にする」など、倫理性の要求に対してどのような公平性指標が正しいのかが大きな検討課題となる場合が有る。また、「処理の中に何等かの意図的差別を含まない」だけでは達成できないことも多く、差別を解消するために「区別した取扱い」を意図的に実装に導入する必要がある場合もある。

更に後者の派生としては、社会レベルでの目標値の設定や、affirmative action（積極的是正措置）と呼ばれるような意図的な不平等性の導入が行われる場合もある。これは社会全体に対するエンジニアリングの観点としてみれば、正しいと考えられる目標値に向かってオーバーシュートを伴うフィードバック制御を掛けている状況と考えられるが、人工知能などのシステム実装の観点からは、社会レベルでの倫理性・公平性の検討を元に、機能性の要件として特定の出力分布を求められていると考えられる。

また、日本でよく言及される「機会均等」「取扱い均等」の捉え方にもいくつかの段階があり、システムとして取扱い（＝アルゴリズムなど）が均等になるように構築段階で対処を行うという「強い取扱い均等」のほかに、システムや手順の構築の段階において不公平な取扱い（＝構築プロセス）を行わないという「弱い取扱い均等」を指している場合もある。しかし、特に統計的機械学習の分野においては、後者の構築プロセスの取扱い均等だけでは、様々な要因から十分な公平性を担保できないことが多い。このことは、例えば人による雇用の取扱いのように、法令やルールなどで想定されている異なる取扱い型差別の一般的な回避方法とは異なるものとして捉える必要がある。たとえば、人事評価などでは「当事者個人」にとって感じる「正当性」を明確に必要とする場合もある。

（参考）集団公平性と個人公平性

一般に公平性要求においては、人種や性別など「不公平」を生じかねない属性（要配

慮属性）についての扱いが問われる。　この際、ある要配慮属性値について、異なる集団の間で差別（例：女性に不利な扱い）を起こさないことが集団公平性であり、いっぽう必ずしもそうした特定の属性による分類に限定せず「似た人」の間で差別を起こさないことが個人公平性である。以下の節で記述するとおり現時点で汎用的に使える機械学習要素の公平性評価（メトリクス）や施策は、「要配慮属性」を要とする集団公平性を前提とするものが主流であって、本ガイドラインでも断りが無い限り集団公平性視点を前提としている。

前述したような「当事者個人」にとっての「正当性」が要求される場合は、個人公平性視点が求められる為、一般的なメトリクスが定義しづらくシステムごとに要求を満たすための施策を検討せねばならない。個人公平性については、距離学習を用いた「似た度合い」の研究［64］等も提案されており今後に期待したい。

8.3.2　曖昧な社会的要請

さらに、システムへの公平性の要求が（部分的にであっても）法令等の要求に起因する際には、法令上の要請と技術的な実現とのギャップも問題となる。例えば法令に「不利益な取扱いをしてはならない」と規定されている場合に、人が直接作業を行う場合のような意志や注意義務のようなものと対比して、機械による自動処理とその設計においてどのような作為または不作為が不利益な取扱いにあたるのかは、必ずしも自明ではない。あるいは、「取扱いを均等にする」ということが、人が機械学習を構築する際に均等に取り扱えば良いのか、機械学習利用システムが可能な限り均等に取り扱うように人が構築を行うのかは、実際の運用上でも大きな違いとなり得る。いずれは本ガイドラインのような規範類やベストプラクティスが普及していくことで、公平性対策の相場観のようなものが形成されていくと考えられるが、現時点においては「どのように考え実装したか」をきちんと説明できることが、強く求められると考えられる。

また、公平性を担保する対象についても、法令などの要請は必ずしも自明では無い。公平性の要求は男女機会均等のように明示的に属性を指定されることもあるが、多くは「性別や人種など」のように非限定の列挙で行われることも多い。また明示されている場合であっても、その属性が入力に明確に含まれない場合や、その属性そのものが推定の対象となり誤差を含みうる場合などについても、やはり実装方針の明確な説明が必要となるであろう。

8.3.3　社会に埋め込まれた不公平性

機械学習要素をデータから構築する際には、訓練用データセットなどのデータそのものに不適切なバイアスが含まれる場合を考慮することが重要である。実社会から得られたデータは時に、社会そのものに含まれる不公平性をデータに内在する場合がある。実際 8.1.2.1 節で紹介した採用判断の事例は、過去の採用プロセスそのものに気付かない差別性があり、機械学習が忠実にその傾向を再現してしまったことが問題の原因と言われている。

この社会そのものの不公平性は、構築プロセスから要配慮属性などを排除しただけでは必ずしも公平なシステムができない 1 つの原因となっている。

このようなことから、公平性が重要な機械学習利用システムの構築にあたっては、データそのものからの演繹だけでなく、分析的な視点から公平性要求を明確にして、データ整理段階での精査などに繋げるべきである。このような活動はまた、必ずしも具体的に明言されていないシステムへの社会的要請を明確にし、説明性を向上させることにも繋がる。

（参考）COMPAS 事例：

米国で禁錮刑受刑者の仮釈放の判断材料として再犯率の推定に用いられる COMPAS システムが、黒人受刑者に対して有意に不利な判断をしているのではないかと指摘され、さらに公平性の担保として、犯罪摘発の偏りなどの社会的なバイアスが訓練データそのものに含まれているのではないかと指摘された。公平性メトリクス定義により、不利か否かの評価が異なってくることからも、メトリクス定義の重要性をも示唆することとなった。

8.3.4　隠れ相関と Proxy 変数

前項に加えて、膨大な属性や情報量を持つ入力データから機械学習要素を構築する場合には、一見して差別的ではない他の属性や、明確に属性として特定されていないような入力の特徴などが要配慮属性と隠れた相関を持ち、結果的に差別を統計的に再現してしまうことがある。例えば、氏名と性別とか、学歴の学校名と性別、住所と収入額のような属性間の相関関係や、画像の背景と人の性格・性別などの、要配慮属性と相関する情報が訓練元データに含まれていると、要配慮属性の出力への直接的な寄与を削除しても、結果的に公平でない出力が生み出されることが有り得る。

またこのような場合には、前項のように社会的な不公平をデータ合成で削除しようとしても、実社会において妥当な訓練データを合成できない場合も有り得る。例えば学校名に男

子校・女子校のような特徴がある場合、入力データの性別だけを反転させたデータを人為合成して加えても、妥当なデータにはならない。

8.3.5　不公平性と「区別すべき」変数

構築過程で不公平性の除去を行おうとした場合、要配慮属性と間接的な相関を持つ属性値については、たとえ正しく同定できたとしても、どのような扱いをすべきかは自明ではない。例えば貸し出しの与信業務のような事例において、「返済可能額」は測定不能だが推定したい属性であると考えられる。このとき、性別のような明確に「差別すべきでない」属性や、あるいは「貸出額」のような目的関数に明確に入力として特定できる変数の場合には、それぞれ扱い方の方向性は自明である。一方で例えば「収入額」のような属性は、「返済可能額」と密接に相関する一方で、社会的な格差・差別を反映している場合も有り得る。従って推定性能向上視点からは使いたいが、公平性視点からは使えないといった可能性が出てくる。このような場合には、公平性担保の方針の決定は難しく、また説明も重要になる。

8.3.6　公平な AI に対する回避攻撃

最後に、公平な機械学習利用システムに対して利用者などが様々な攻撃をしてくることも、想定に入れる必要がある。継続的学習を行うようなシステムで、攻撃者が偏ったデータなどを与え続けることで不公平性をシステムに埋め込む場合や、逆に公平にするプロセスによって歪んだ機械学習モデルに対して敵対的データを与えて利得を得るようなケースも考えられる。

8.4　公平性担保に対する基本的な考え方

本章では、前節で列挙した課題を踏まえ、本文書が目指す機械学習応用における公平性品質マネジメントの考え方を整理する。

8.4.1　公平性担保の構造モデル

一般的に製品・サービスに関する平等性・公平性などの要件は、利用者とサービス提供者

の間の関係（利用時品質）においては「不公平に取り扱われない」「平等に取り扱われる」などの抽象的かつ定性的な要求として表現される。一方で、機械学習 AI 技術などの分野の文献等で扱われる「公平性メトリクス」等（8.5.2.4.2.1 節）は、機械学習品質マネジメントガイドラインで言う「内部品質の指標」の 1 つに当たり、システム構築から運用までの段階で、入力データセットや出力結果についてある特徴に着目した各種の偏りの度合い（バイアス）を数値化したものであり、データの 1 つの側面を切り取った数値指標となっている。この 2 つの表現の間には大きなギャップがあり、利用時品質である「平等な取扱い」の実現を確認する為に、どのような特徴に着目した公平性メトリクスで評価を行うかを検討することが、製品の公平性担保のために重要なポイントとなる。

このような観点から本ガイドラインでは、

- 社会的要請や利用時品質など高い抽象度の段階における「平等な取扱い」を定性的に扱い、
- 不公平性の発生をリスクと捉えたリスク分析アプローチにより具体化し、
- 必要に応じて開発のいずれかの段階から「結果の平等性」など定量的な公平性メトリクスを通じて実現する

一連のプロセスを、機械学習システムにおける公平性の実現方法として想定する。この考え方は、公平性メトリクスの設定や選択の妥当性を、分析や設計のアプローチによって担保しようとするもので、「リスク回避性」や機能安全性などの実現におけるリスク分析ベースのアプローチとも類似しているとも考えられる。

下の図 18 は、このような考え方の一例を図示したものである。この図は個々の開発の考え方や段階の設定を拘束するものではなく、定性的なアプローチから定量的な手法に至る流れを整理するモデルである。

① 公平性の要求は、抽象度の高い「正義」あるいは「人権」のようなレベルでは、「平等」「平等な取扱い」のようなキーワードで表現される事柄から起因すると考えられる。

② 法制度・あるいは暗黙の倫理的行動などの社会ルールのレベルでは、8.3.1 節で示すように、「取扱いが平等である」ことを要求する場合と、数値目標などの形で「結果が平等である」ことを求められる。

　後者の場合、数値目標の見直しなどの後プロセスはあるにしても、制度レベルでは取扱いが平等でないことを許容していると考えられる。例えば、男女雇用機会均等の文脈で、女性の社会進出が進まないことが社会経験豊富な女性の育成を妨げ、そのことが女性の社会進出を遅らせているような負の事象の正帰還の構造がある場合に、ある段階では社会進出を先に数値目標などで強制的に促進することにより、

正帰還を正の事象に誘導するようなことが考えられる。この場合、結果の数値的な均等性の達成が第一義の目的になるので、これ以降の段階も全て数値的な均等性の達成に帰着される。

また、「取扱いが平等である」ことは、「平等な取扱いになるように積極的に設計実装を行う」という強い要件と、「不平等な取扱いを意図的に行うことは避ける」という努力目標的な弱い要件の 2 つを含んでいる。本文書では主に前者の積極的な達成を取扱い、努力目標で許されるレベルの弱い要件については対象外とする。

③ システム全体の設計に対応する利用時品質および、

④ 機械学習要素の設計に対応する外部品質のレベルでは、目標設定に再び結果の数値的な均等性と取扱いの平等性の 2 つの選択肢が現れる。この机上検討段階で結果の分布などが十分に想定可能な場合には、このレベルで数値目標に転換してシステム構築することが考えられる。

⑤ 次に、内部品質の一部（機械学習品質マネジメントガイドラインにおける内部品質 A-1〜A-2 に相当）システムの構築プロセスの中でも、同じように 2 つの目標設定の選択肢が有り得る。この段階では、収集した具体的なデータなどを元に分析を行い、目標を設定することが考えられる。

⑥ 最後に、品質の確認手段と対応する内部品質のレベルでは、結果の統計的分布を分析する方法、結果分布以外の統計的・分析的指標をモニタリングする方法、そして実装の論理的構造から取扱いの平等性を説明する方法が考えられる。

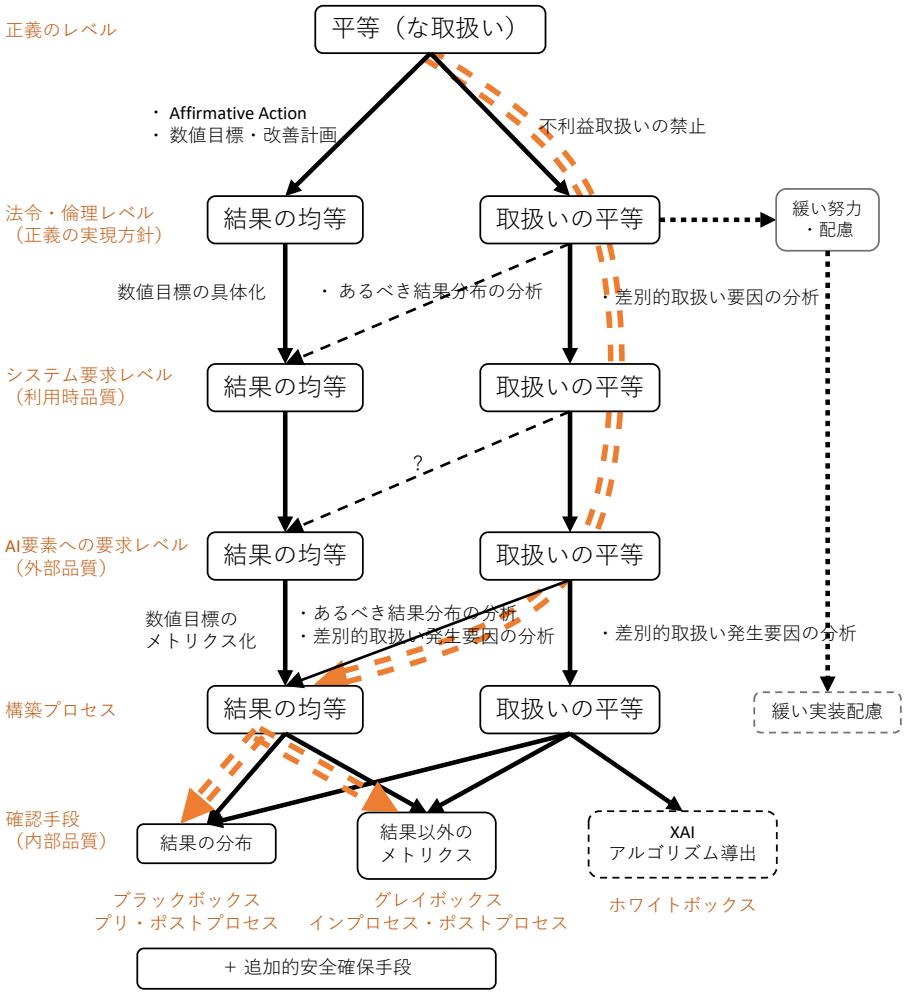

図 18: 公平性品質の確保に関するプロセス構造の例示

8.4.2 公平性担保の方向性

さらに具体的に、本章の残りでは図 18 中の点線で示すようなプロセスを、機械学習実装における基本的な公平性マネジメントにおいて一般的に検討できる基本的な方向性として想定する。すなわち、外部品質のレベルでは 1.5.3 節の公平性の定義に沿って特徴・属性間の担保すべき「公平性」を論じ、内部品質の検討の段階でデータ分布などの分析を行い、数値的な結果の「バイアス」への要件に転換する。そして、その分析結果に基づきシステム構築段階およびテスト段階で結果の分布のバイアスを確認するとともに、必要と可能性に応じて付加的な指標で直接的な「公平性」の確認を目指す。

これは、機械学習の実装がデータから導出されることから、構築段階⑥では訓練データセット・テストデータセットの分布に公平性要件を反映するべきという基本的な考え方と、8.3 節で議論した複雑な問題構造を整理するためには、実装の段階で具体的なデータに基づ

く検討が必要であり、対応する実装開始段階⑤までは抽象的な公平性の要件を維持する必要があるという考察に基づく。

もちろん、個々のシステムの置かれた機能要件の状況などに応じて、図中のとりうる開発プロセスは選択して良い。例えば、数値目標があらかじめ②段階で設定されているような応用では、その数値目標を達成することを機能要件として実装を行うことになる。

ここで、前節②に掲げた「努力目標としての意図的な不公平な取扱いの回避」は、AIFL 0 レベルの製品サービスに対する普遍的な目標と対応づける。(AIFL 1 の “best” effort を満たすとは考えない。)

8.4.3 要配慮属性に関するデータの取扱いに関する留意点

個人情報の中には、人種・性別・出身地など様々なレベルで、取扱いに注意を要し、それに起因する不公平な扱いを避けなければならないような「要配慮データ」が存在する。公平性を要する機械学習要素を構築するにあたっては、システムとしての要件を検討する上でも重要となってくるため、図 18 のプロセスの早い段階からこれらの情報の扱いには慎重を要することは言うまでもない。

本文書は、これらの要配慮データを取り扱わなければ自動的に公平になるという立場は取らない。特に実世界から取得したデータは複雑な構造や相関を持ち、直接的な要配慮属性を入力に含めない場合でも、他の入力データから構築した AI が結果的に要配慮属性に相当する値と相関を見いだすことが十分に考えられる。さらには、要配慮属性を含むを構築 (特にテスト工程) に用いないことにより、公平性に関する品質を確認・検査不可能になることが大きなデメリットになることがありえる。むしろ公平な機械学習システムを構築するためには、公平性担保の対象とする属性データについては少なくとも構築時に取得すべきことが多いことを明確にし、システム設計および構築プロセスの検討の段階で十分に配慮するべきと考える。

一方、品質のモニタリングや追加学習の目的で運用時にセンシティブなデータを取得できるか否かについては、個人情報保護などとの関係から検討を要する。また、例えば人種や病歴などとくに要配慮の機微情報に関しては、システム構築の段階でも取扱いそのものが不可能である場合も考えられる。これらの場合には、モデル構築段階や追加学習段階で、どのように偏りを除去しモニタリングを行うか、について十分な配慮を行う必要がある。

8.5 公平性の品質マネジメント

8.5.1 公平性要求の詳細化・言語化（事前準備）

機械学習において公平性の要求レベルが AIFL 1 以上と想定される場合には、いわゆる要件定義の段階において、以下の点について検討を行う。複数の要素が公平性に関係する場合には、原則として最も要求の厳しい要素に対応してレベルを決定するものとする。

8.5.1.1 公平性の目標の明確化

はじめに、公平性の対処目標を以下の観点から明確化する。

① 公平性の取扱いの判断の元となる属性を検討する。

(ア) 当該機能が公平性の配慮対象とすべき直接的な要配慮情報を明確化する。この情報は、入力データの属性として明示されるものに限定しない。例えば、性別・年齢・人種・家系などの情報が特定されうる。
(イ) 可能であれば、上記の要配慮属性からの影響に留意しつつも判断に用いてよい可視の情報（例えば筆記試験の成績、選択学科など）を明確化する。
(ウ) 更に可能であれば、上記の判断対象属性の元となり、要配慮属性に影響されない理想的な特性（潜在的な学力、卒業可能性など）を言語化しておく。

この検討においては、対象システム（あるいはそれを含む社会的な活動）が考慮するべき公平性の範囲、あるいは事前条件的に与えられた「公平である／公平でないとみなす指標」の仮定などにより、検討結果やその後の取り組みが異なってくることに留意する。

（例示）
例えば「筆記試験による入学試験」というシステムを考え、その一部としての採点・合否判定処理を検討の対象とする。
用意された試験問題を一斉に解くことが受験者に公平な機会を与えていると考える立場では、「試験問題に対する解答の妥当性」が合否の最終判断に用いるべき抽

象的な指標となり、「採点結果の点数」がそれに対して適切な数値指標になるかどうか等が問われる。

一方で、例えばアメリカで近年議論の対象となる大学受験の事例 [51] のように、試験というシステムそのものに公平性の問題が有り得るという立場では、例えば「潜在的な学力」「卒業可能性」などが理想的な判断基準の特性となりえる。そして、「採点結果の点数」にはこれらの基準となる特性に加えて、過去の教育機会の不利の影響などが加わると考える。そして、これらの不利に強い相関を持つ特徴として、人種差や収入差などがあり、これらの要素を判断して「採点結果の点数」に補正を加えなければ、理想的な判断基準に近い公平な合否判断をできない、と論ずることになる。

② それぞれの要配慮情報について、基本的な公平性取扱い方針を明確にする。

(ア) その要配慮情報は、対象者が当該情報を理由として差別的に扱われない (取扱いの均等) ことを要求されているのか、あるいは具体的な数値目標などに照らして結果が一定の事前定義された分布に依ること (結果の均等) を要求されているのか。

また、当該の情報の取扱いに関して、男女雇用機会均等や人種均等、affirmative action など法令等の要請があるか。

(イ) その要配慮情報に関連する差別が仮に現実社会に既に存在している場合、その現実社会から採取されたデータに基づいて構築される AI は、その差別をどのように扱うのか。可能な限り除去すべきなのか、一定程度の残存は許容できる可能性があるのか、あるいはシステムの要求上再現しなければならないのか。

(ウ) 機械学習の成果に残存する差別的な判断を除去するために取りうる手段への制約。訓練データへの意図的な補正の導入や、意図的な訓練済みモデルの調整を行うことが許されるのか。また、取りうる手段で問題を修正できない場合に開発中止の判断をどうするのかについても明確にしておく。

以上の判断を行った後、公平性要求レベル (AIFL) の確認を再度行う。

8.5.1.2　データ属性間の依存性と因果関係のモデル化

実世界から取得したデータは複雑な構造や相関を持つため、属性間の依存性・因果関係を

正しく把握しないで学習をした場合、意図と異なったり、不正確な結果を生んだりすることに繋がりかねない。

元よりプログラム記述による論理記述でなく、データを用いた機械学習を用いた実装を予定する以上、属性間の依存性・因果関係は完全に既知のものとなっているわけではない（分かっているのであればプログラムを記述できるであろうから）と考えられ、分析を尽くしても完全に把握することは困難であると考えられるが、公平性を要求されるシステムの実装は、以下の点について、出来る限り事前に把握することが重要と考えられる。

① 不公平さに「繋がりかねない」データパス

要配慮属性がシステムの出力に不公平な影響を与える過程には、その属性がシステムの出力に直接影響を与える場合に加えて、要配慮属性の影響を分布に受ける別の属性、更にその属性に間接的に影響を受ける属性などの寄与が有り得る。これらの一連の「パス」を把握しておかないと、直接影響を除去しただけでは最終的な公平性の要求を実現できない・説明できないケースがしばしば有り得る。

また、システムの判断に寄与するべき属性と、システムの判断に影響するべきでない要配慮属性の双方から影響を受ける属性が存在する場合、要配慮属性に関しデータセットの偏りを排除するといった典型的な事前処理は、結果としてシステム判断に対してマイナス要因となりかねないため、注意が必要となる。

② 「シンプソンのパラドックス」

「シンプソンのパラドックス」とは、二つのデータ属性間の関連を見出す際に、実はそれら双方に関与する第 3 の要因に気づかず学習した場合、現実には存在しない相関関係、あるいは現実とは逆の相関関係を学習してしまう現象を指す [97]。このような現象防ぐためには例えば該当する第３の属性を事前に把握し、それに関して「場合分け」を意識した学習用データを準備する必要がある。具体例はこの後で述べる。

③ 要配慮属性の必要性

あえて、要配慮属性を使わないと、公平性が実現できないケース、すなわち「fairness by unawareness」では目的が達成できないことが論理的に明らかなケースがあり得る。

上に示したような分析には、例えば関係表や文章による解析などを用いることができる

が、比較的簡潔にこれらの相互的な関係を表現できる図表現の 1 つとして、Causal Bayesian Networks（CBN）[51] を挙げる。CBN は、属性間の影響をシンプルな依存関係で表現し、間接的な影響を含む不公正さの発現するシナリオの可能性を見つけやすくする図表現である。また、使うべき公平性メトリクスの種類の検討にも役立つ。

（例示）

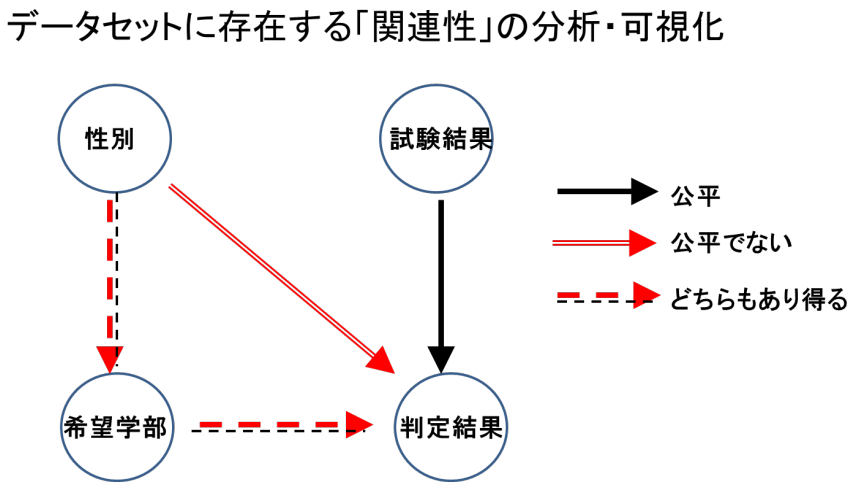

図 19:　CBN を用いた関連性可視化例（[51] より）

図 19 は、[51] 等で取り上げられている「大学の合否判定」事象である。CBN では、データ間の因果を矢印（causal path）で表し、それが「公平」なものか、「不公平」なものか、あるいは、「どちらもあり得る」のかを検討することで、関連性を可視化する。

この例では、

・　「試験結果」から合否判定への影響は、当然ながら「公平」である。

・「性別」が、直接合否判定へ使われならば、すなわち、同一の試験結果に対して性別によって判定結果が異なり得る、のであれば、当然ながらそれは「公平でない」関連性となる。

「性別」から「希望学部」への関連は、もしも女性は特定の学部を希望しやすい、という現実が、女性への暗黙の環境的圧力によるのであれ「公平でない」ことになり、同様に「希望学部」から判定結果への関連は、「女性が希望しがちな学部合否基準点を恣意的に操作する」といったことをするならば「公平でない」ことになる。

このように、合否判定に関連を及ぼすデータ間の関連性を丁寧に分析することで、「性別」−「希望学部」−「判定結果」というパスはその実態やアルゴリズムによって、公平性を損なう要因になることがわかる。　これは「性別」という要配慮情報のみを除いて学習させただけでは「判定結果」の公平性が担保される保証がないことを意味する。

図 20 は、先に触れた「シンプソンのパラドックス」の例を CBN で示した例である。「週当たり運動日数」と、「心臓健康度」という属性の間にある相関関係を抽出したい場合、その裏側で双方に影響がある「年齢」という第 3 の要因を考慮せず集めたデータセットで学習した場合、目的を果たせない可能性がある。
簡単化の為に、「年齢」は、「運動日数」に正の相関を持ち、「心臓健康度」には負の相関を持つ、と仮定しよう。この場合、本来「運動日数」と「心臓健康度」が正の相関を持っていたとしても、逆の見せかけの結果が発生しうる。このケースの「年齢」のような第 3 の要因は「交絡因子」と称される。　公平性視点に限らず、学習データセット準備において留意すべき事項となる。　CBN を描くことで「交絡因子」の発見が促され、また、もしも要配慮属性が交絡因子になっていた場合、それを除去するだけでは不適切な学習となりかねない事が、事前に認識できる。

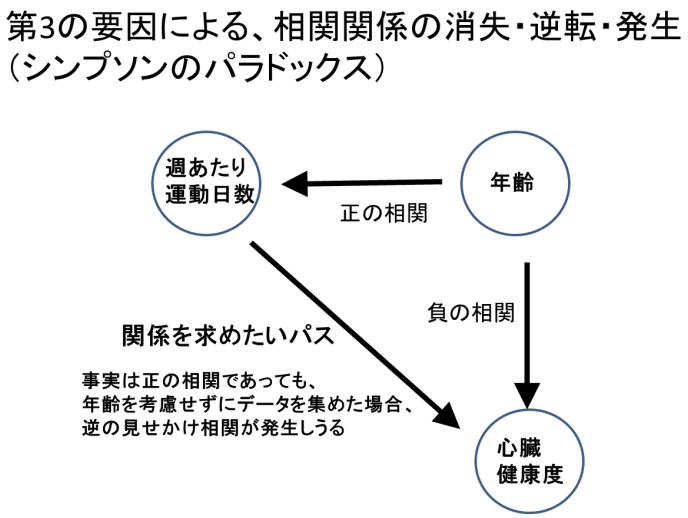

図 20: シンプソンのパラドックスの CBN 例（[51] より）

8.5.2 公平性要求の実現

8.5.2.1 施策概要

公平性実現のために必要な施策概要を説明する。施策のスコープにより、大きく次のように大別される。

A) 学習用データセットに対するもの
B) 学習そのものに対するもの
C) 学習済モデルへの調整
D) 運用時の調整・使い方の工夫

また、これらは「公平性対策の開発工程別の 3 分類」[85][130][95]に次のように対応する。

A): Pre-processing アプローチ

B): in-processing アプローチ

C)および D): Post-processing アプローチ

機械学習モデルを構築するプロセスの「どの部分」を実施可能か、すなわち開発可能なスコープによって、以下のように可能なアプローチが定まってくる。

- 学習用データセットを直接分析・調整可能であれば pre-processing は「特定のメトリクスで、公平であることを担保したデータセットで学習した」といった比較的判りやすい説明が可能になり、また一般に効果も高いため試みる価値がある。
- 学習フェーズを初期から実施できるのであれば、in-processing のアプローチが可能である。
- 学習済モデルを受け取り、それに対しての後処理（あるいは、さらなる追加学習）を実施するのみであれば、post-processing に限定される。

公平性の目標レベルも踏まえると、おおよそ基本方針は次表のようになる。

表 5　作業スコープと AIFL による施策の概要

	学習用データセットに手を加えられる場合	モデル開発・学習が出来る場合	学習済みモデルの後処理のみができる場合
AIFL 2	学習用データセットに関する公平性メトリクス「目標」を定義し、可能な限り満たされるよう必要であれば複数の pre-processing を行い改善させ記録する。	モデル出力に関する公平性メトリクスを「学習目標」に追加。AI パフォーマンス等他のメトリクスとバランスを取りながら、in-processing を駆使して目標値を達成させる。	N/A (学習済モデルの調整、運用時の調整で可能な施策を予め充分検討、必須条件が満たされる可能性が低い場合は、作業スコープの見直しが必要となる。)
AIFL 1	学習用データセットに関する公平性メトリクス「目標」を定義し測定。乖離がある場合は pre-processing で改善させ記録する。	モデル出力に関する公平性メトリクスを「学習目標」に追加し、AI パフォーマンス優先で学習させた結果が、目標と乖離がある場合は in-processing で改善させ記録する。	モデル出力に関する公平性メトリクスを定義しテスト時に測定. 目標と乖離がある場合は学習済モデルの調整、運用時の調整で改善を図る。
AIFL 0	学習用データセットに関する公平性メトリクスを定義し、測定し記録する。	モデル出力に関する公平性メトリクスを定義し、学習時に測定記録する。	モデル出力に関する公平性メトリクスを定義しテスト時に測定記録する。

以下の節では、主として 1.7 節および 6 章で掲げた内部品質に沿って施策の詳細を述べる。

8.5.2.2　A-1: 要件分析の十分性

事前準備で提示したガイドに従うことで、以下の観点での確認が可能となる。

- 属性間の依存分析
- 訓練データ分布への要求明確化
- 確認すべき測定指標（公平性メトリクス）

加えて、開発システム全体の性質をふまえ、次の観点の要件を十分に確認する必要がある。

・　法的・社会的要請との整合性

法的・社会的要請は、開発チーム内で充分な知見を持たない場合もあり、当該分野エキスパートに諮ることが望ましい。

目標とする AIFL レベルに応じ、実施すべき施策は以下のとおりである。

・ AIFL 1
  - 公平性要要求の定義をし、経緯を含めて記録しておくこと。
  - 当該要求を踏まえ、データ準備に対しての要求（学習用データセットに関する公平性メトリクス）を定めること。
  - 当該要求を踏まえ、モデル出力に関する公平性メトリクスを定めること。
・ AIFL 2
  - AIFL 1 の要請に加えて以下の対応を取る。
  - データ属性間の依存性と因果関係のモデル化をすること。
  - モデル化の結果を学習用データセットの公平性メトリクスに反映させること。

学習用データセットに関する「公平性メトリクス」は、モデル出力に関するものと同一視点が要求されるのが基本である。例えば、出力に対して「性別のみの差による合否判定の差を起こさない」が公平性要求の場合、当然ながら学習用データセットにも、その視点での偏りが無いことが望まれる。ただし、学習用データセット（これまでの現状）が偏っていても、前述したように作業スコープによってはデータセットに関するメトリクス達成は目指せず、モデル出力についてのみとする場合もある。なお、メトリクスの例については、8.5.2.4.2.1 で述べる。

8.5.2.3　B-1～3: データ準備（データの被覆性、データの均一性、およびデータの妥当性）

データ準備段階では、主に「データの妥当性」「データの網羅性」および「データの均一性」の内部品質視点で、公平性確保を目指して「Pre-processing」の対策を取る。一般に Data Fairness（「データの公平性」）実現と呼ばれるのはこの段階の処理であり、データ収集に関するものと、収集したデータに対するものがある。

8.5.2.3.1　偏りを入れ込まない学習データ収集プロセス

異なる取扱い型差別（Disparate treatment）防止視点で、sample size disparity や tainted examples[12] など、まずは「データ収集に関するバイアス」、すなわち、現実世界に対して遍在をもったデータになってしまう状況、を起こさないプロセスを極力確保する。

さらに、収集されたデータに対して、グループによる重み付けによって対等性を確保する方法（Reweighing [74]）なども活用できる。

これらの施策は、「注目すべき属性組み合わせに対して十分なデータがあるか」の確認、あるいは「現実世界の分布に近いデータになっているか」の確認、という意味で、「データの網羅性」および「データの均一性」視点の作業といえる。

8.5.2.3.2　データ調整・強化・合成

異なる効果型差別（Disparate impact）を防ぎたい場合は、たとえデータは現実に対して偏りがなくともデータ自体に内在する（現実の）歪み、差別要因への対策が「データの妥当性」視点から必要となる。

主な対策例には以下がある。

- Disparate Impact Remover

  いわゆる「Proxy」によるバイアスを除去するために、要配慮属性と関連がありそうな属性「値」に調整をかけ、当該属性から間接的に要配慮属性が推察できる可能性を下げる。

- Optimized Pre-Processing

  学習用データ（そして使用の際のインプットデータにも使われる）を事前に処理するパイプライン的な変換のためのフレームワークを。　バイアスの除去と、データの有用性（実データから離れすぎない）の両面への配慮がされている [46]。

8.5.2.3.3　必要な施策

目標とする AIFL レベルに応じ、データ準備段階で実施すべき施策は以下のとおりである。

---

[12] Sample size disparity: 要配慮属性の値により大きくデータ数が異なること。

Tainted Sample: 人によるラベル付けにより生じるバイアス

- AIFL 1
  - データセット用公平性メトリクスを測定し記録すること。
  - データセット準備が作業スコープ内であれば、メトリクスが目標と乖離している場合、少なくとも 8.5.2.3.1 で述べた「異なる取扱い型差別防止」手法を実施し、改善を図ること。

- AIFL 2
  - AIFL 1 の要請に加えて以下の対応を取る。
  - データセット準備が作業スコープ内であれば、メトリクスが目標と乖離している場合、8.5.2.3.2 で述べた「異なる効果型差別」防止施策手法の実施を検討すること
  - データセット準備が作業スコープ外の場合、目標との乖離がこの後のモデル出力に関するメトリクスに与える影響を最小化するための施策を検討すること。場合によっては、機械学習要素の外側での担保なども事前に検討しておくこと。

（参考）
ここで説明した Pre-processing 手法は、8.6 節で紹介する開発基盤・ツールとしても提供されており現時点でも比較的容易に用いることが可能である。ただし入力データは、要配慮属性が明らかになっている一連の「属性と値」（構造化データ）を想定しているため、例えば大量の文章（コーパス）を入力とする自然言語処理系の機械学習システムでは、直接用いることが難しい。自然言語処理における公平性については、言語モデリングおよび特徴学習手法におけるバイアス除去手法などが Mehrabi et al. [85] でも紹介されているが、現場で広く活用するためには更なる研究が待たれている。現時点では、学習結果として出来上がった機械学習モデルの post-processing でのテスト・調整にて公平性を担保する事も現実的なアプローチとなる。

8.5.2.4　訓練、テストでの確認（C-1～2: モデルの正確性およびモデルの安定性）

準備されたデータセットを用いた訓練では、内部品質「モデルの正確性」「モデルの安定性」視点での公平性確保を目指して「in-processing」および「post-processing」の対策を取る。

8.5.2.4.1　モデル開発・学習段階での対策（in-processing）

In-processing な対策とは、モデル自体への公平性の埋め込みを狙うもので、以下に述べるとおり objective function への修正（制約条件の追加）、あるいはモデル自体追加にて実施される手法がある。

8.5.2.4.1.1　制約条件の追加型対策の例

- **Prejudice Remover**
  要配慮属性を直接使うことは避けた場合でも残る、「間接的な prejudice」[13]を除くための制約を追加した形で、結果確率分布を調整する。その為に制約自体は要配慮属性を用いた Regularizer として実装され objective function に追加される。
  適用可能な課題対象は識別問題に限定されるが説明性が高い [70]。

8.5.2.4.1.2　モデル追加型の例

- Adversarial Debiasing（敵対的バイアス除去）
  Pre-processing の段階で Proxy の解析を完全に行えてない場合、要配慮属性には依存しないように学習させたいターゲットモデルに加えて、そのモデルを使った推論結果を入力とし、要配慮属性を推論する Adversarial モデルを追加する。そちらの推論が出来る限り失敗するようターゲットモデルを学習させる。
  適用可能な場合には、汎用性が高くかつ不十分な Proxy 解析の問題をカバーできる可能性があり推奨できる [121]。

8.5.2.4.2　学習済モデルへの調整

Post-processing に分類される対策のうち、ここでは「訓練、テストでの確認（モデルの正確性およびモデルの安定性）」に相当する「学習済モデルへの調整」について述べ、「差別が残っていることを前提とした運用時の調整」については、次節　内部品質「運用時の品質維持性」で述べる。

学習済みモデルに対して要求される公平性目標メトリクスに応じて、Equalized Odds Framework やキャリブレーションといった調整がある [99]。

[13] バイアスの原因のひとつ。目標変数あるいは非要配慮属性がもつ、要配慮属性への統計的依存性

目標メトリクスは、学習時（前述の in-processing の対策時）においてもむろん「モデルの正確性」内部品質対象として定量評価が必要となり、適切に定め計測することが欠かせない。

8.5.2.4.2.1　目標メトリクス例

メトリクスは、いずれも「異なる効果型差別」が無いことを計測するための指標として位置づけられるが、差別の考え方によって選択される。主なものを以下にあげる。このほかの例は文献［85］等を参照されたい。

- equalized odds　要配慮属性に拠らず、「正しい判断」（$TP/(TP+FN)$ および $TN/(FP+TN)$）をされる割合が等しい
- predictive parity　要配慮属性に拠らず、「精度」（$TP/(TP+FP)$）が等しい
- demographic parity　要配慮属性に拠らず、「望ましい結果」（$(TP+FP)/(TP+FP+TN+FN)$）割合が等しい

8.5.2.4.3　必要な施策

目標とする AIFL レベルに応じ、訓練・テスト段階で実施すべき施策は以下のとおりである。

- AIFL 1
  - モデル出力メトリクスを測定し測定すること。
  - 目標との乖離がある場合、in/post-processing 手法を用いて改善を図り、乖離がある場合は差異を記録する。
- AIFL 2
  - AIFL 1 の要請に加えて以下の対応を取る。
  - 乖離がある場合、複数の in/post 手法を試し、乖離の解消を可能な限り図る。
  - それにも拘わらず解消されない乖離がある場合、運用時の追加施策を組み合わせる検討をする。

8.5.2.5　運用時の調整・使い方の工夫

これまでに述べた対策をとった場合であっても、学習済みモデルには、バイアスが除去しきれてないことを前提とし、実際に推論をする場合に何等かの調整をする対策がある。推論実

施環境にて柔軟な運用が可能な方法として提案されている手法の一例をあげるが、これ以外でも、機械学習要素の外側も含めたシステム全体で現場に即した調整は工夫可能である。

- Reject Option-based Classification (ROC)
  Discrimination はモデル推論時「確信度」が低いところで起こる、という仮説に基づき、確信度の低い「差別になりかねない推論結果」は、判断に用いない、または結果を修正する。具体的には、「優遇グループ」の「有利な結果」や、「非優遇グループ」の「不利な結果」が、指定された確信度より低く出た場合に、その結果を書き換える [73]。

8.5.2.6　運用時品質の維持性（E-1）

公平性が重要な製品やサービスは、公共空間など外部に開かれた使い方をされる場合も多く、コンセプトドリフトなど運用時の品質変化には充分な配慮が必要である。前述したモデル出力に関する公平性メトリクスの測定、またデータセットに関する公平性メトリクスの測定を適切なタイミングで実施することは、AIFL レベルに拠らず求められる。

8.6　公平性に関する開発基盤・ツール

8.6.1　開発基盤・ツール活用の趣旨

8.5 節で述べた作業は、公平性以外の品質目標実現視点と共に機械学習ライフサイクルにて実施することになる。特に公平性は、データセットに内在するバイアスの同定や、学習済モデルに対する公平性メトリクス測定など含め、様々に絡み合った試行錯誤的な検討が必要となることが多い。また定性的なプロセス面での作業内容も重要なエビデンスとなろう。そのため、

- データセットに対し 8.5.2 節で述べた観点の検討を効率的に行う。
- 学習済みモデルについて、なんらかの可視化や、メトリクス評価、その他の公平性評価を行う。
- 作業記録、その際に用いたデータセットや一連の作業を適切に管理する。

といった目的のために開発基盤・ツールの活用は有益である。

8.6.2 開発基盤・ツールの事例

8.6.2.1 典型的な機能・モジュール

前節に述べた趣旨から望まれる主な機能群を以下にあげる。

① データセットの可視化（要配慮属性や関連の発見のための分析）
② メトリクス以外の公平性確認（CBN で述べたように特定のデータ（データポイント）に対する Counterfactual 等、一部の属性を現実と変えた場合の実験など）
③ 様々な公平性メトリクス計測
④ XAI ライブラリー（決定に、どの属性が寄与したかの可視化）
⑤ 運用後の継続的モニタリング＆再学習といったライフサイクルを支援するパイプライン構築基盤（いわゆる DevOps 基盤＋データセット、モデルに関する基盤）

以下に Google と IBM Fairness360 を参考に紹介するが、他の各種ベンダーからの新機能も継続的に提供されるため、最新の情報を確認のうえ、最適なものを選択されたい。

8.6.2.2 Google のツール例

前節で述べた機能①～⑤が、以下のようにカバーされる。

What-If-tool ①、②
https://pair-code.github.io/what-if-tool/learn/tutorials/walkthrough/
Fairness Indicators ③
Explainable AI ④
https://cloud.google.com/explainable-ai/
AI-Platform ⑤
https://cloud.google.com/ai-platform

公平性に直結する多彩な機能を提供する What-If-tool は単に訓練用データセットの可視化ではなく、仮想シナリオや、様々な属性でスライスした場合の分析機能などを通じて、データセットに潜在的に存在するバイアスパターンの発見に役立つ。

また、Explainable AI は、推論時にどの属性が決定にどれぐらい寄与したかを「定量的」に示すことが出来るためバイアスの有無確認に寄与する。（8.2 で述べたとおり、影響がある＝不公平とは限らない点には注意。）

8.6.2.3 IBM のツール例

前節で述べた①～⑤の機能が以下のようにカバーされる。

- AI Fairness 360 / IBM Watson OpenScale ・・・ ①、②、③
  - https://aif360.mybluemix.net/
  - https://www.ibm.com/jp-ja/cloud/watson-openscale
- AI Explainability 360 / IBM Watson OpenScale ・・・ ④
  - https://aix360.mybluemix.net/
  - https://www.ibm.com/jp-ja/products/cloud-pak-for-data
- IBM Cloud Pak for Data ・・・⑤

AI Fairness 360 と AI Explainability 360 は、IBM 基礎研究所が開発し公開したオープンソース・ソフトウェアであり、2020 年に Linux Foundation に寄付されている。AI Fairness 360 では機械学習モデルやデータセットのバイアスに対処するための豊富な機能が提供されており、本文書で解説した方法論が概ね網羅されている。

- Pre-processing：学習用データの再重み付け（reweighing）、差別的効果の除去（disparate impact remover）、学習用データの最適化（optimized preprocessing）他に対応。
- In-processing：偏見除去（prejudice remover）、敵対的バイアス除去（adversarial debiasing）他に対応
- Post-Processing：等化オッズ（equalized odds）、キャリブレーションされた等化オッズ（calibrated equalized odds）に対応。
- メトリックス：Predictive parity 他に対応。Equalized odds と demographic parity は提供されるモジュールを使って計算が可能。

IBM Cloud Pak for Data は機械学習モデルのライフサイクルを統合的にカバーするマルチクラウド対応ソフトウェアであり、IBM Watson OpenScale はそれを構成する 1 つのソフトウエアモジュールである。モデル運用時に入出力データを監視し、公平性指標の時系列変化を検知する機能は、運用開始後に発生するバイアスの改善に役立つ。

9. セキュリティに関する留意事項

9.1 全体の考え方

本章では機械学習利用システムや機械学習要素を構築する際に留意すべきセキュリティ面の要点を整理する。

攻撃の可能性をアタックツリー解析によって洗い出し、対策していくという考え方は、機械学習利用システムの構築においても変わらない。その際、機械学習におけるセキュリティには、一般的な情報セキュリティでこれまで対応されてきた脅威のほかに、機械学習特有の攻撃界面や脅威があり、そのための追加的な対策が必要となる。9.2 節で機械学習システムへの攻撃界面の分析を行い、9.3 節では対策の参考になる考え方や技術の情報を提示する。なお、セキュリティ攻撃の分析と対策においては、通常、すべての攻撃を網羅的に把握し対策することができない点に留意する必要がある。

機械学習のセキュリティについては本節のほかに、米国 NIST が作成中のドラフトである NIST IR 8269 [31] が、機械学習システムの安全確保を目的として、機械学習にまつわるセキュリティを「攻撃」「防御」「影響」の 3 つの視点で分類しており参考になる。また関連する文書として、リスクアセスメント実施ガイド NIST SP 800-30 [32] もある。

なお、本章で挙げる各攻撃例については、これらだけを実際に想定される事象として参考にするのではなく、設計・開発・保守・運用における脅威と脆弱性を洗い出す目的で参考とし、列挙した攻撃例以外の事象についても最新の情報を収集して脅威と脆弱性の洗い出しに取り込む必要がある。また、当該の作業は、設計開発時のみならず定期的に実施するセキュリティリスクアセスメントの際にも実施する必要がある。

9.2 機械学習利用システムに対するセキュリティ攻撃の分析

9.2.1 攻撃の対象による分類

機械学習利用システム（以下「システム」）に対する攻撃を、攻撃の対象や内容から分類すると、大きく以下の 4 つに分類できる。

A） 訓練済み学習モデル（以下「モデル」）の誤動作の誘発（9.2.1.1 節）
B） モデルについての情報の窃取（9.2.1.2 節）
C） 訓練データに含まれるセンシティブ情報の窃取（9.2.1.3 節）
D） モデルやシステムの解釈/説明機能の誤動作の誘発（9.2.1.4 節）

9.2.1.1　モデルやシステムの誤動作

モデルの誤動作を誘発する攻撃は、そのモデルを利用する機械学習利用システムの機能要件の達成を妨げリスクを生じる。例えば下記のような影響が生じうる。

- リスク回避性の低下:
  - 自動運転における物体検知の回避による事故誘発、運転者の異常の見逃し
  - 情報セキュリティ対策におけるマルウェア検知の回避
  - 防犯システムにおける侵入検知の回避、異常行動検知や危険人物検知の失敗
  - 病理診断の精度低下による偽陽性・偽陰性の増加
- AI パフォーマンスの低下:
  - 顔認証などの認証の失敗
  - 運輸/物流における配車の割当の効率低下、交通渋滞や物流コストの増加
  - 小売分野における商品推薦や需要予測や店舗状況把握の精度低下、売上の低下
  - 株取引の利益低下、不適当な融資による損失
  - 入学・雇用・人材配置の適切性の低下
- AI 公平性の低下
  - 与信審査システムによる不公平・差別的な融資
  - 人材評価システムによる不公平・差別的な入学・雇用・人材配置
  - 防犯システムによる不公平・差別的な犯罪リスク判定
  - 病理診断システムの精度の差による不公平・差別的な治療効果

モデルの誤動作を誘発する攻撃手法には、特定のモデルに対して誤動作を誘発する手法と、不特定のモデルに対して高い確率で誤動作を誘発する手法がある。

9.2.1.2　モデルについての情報の窃取

モデルについての情報のうち、通常公開されないパラメータ等を窃取する攻撃は、モデル自体の知的財産権の侵害のほかにも、セキュリティやプライバシーへの影響が生じうる。

- モデルについての情報を利用すると、システム誤動作を誘発する入力を構築する手がかりとなることがある。
- モデルの学習に用いた訓練データに関する部分情報など、プライバシー情報が漏洩することがある。

9.2.1.3　モデルからの訓練データに関する情報の窃取

機械学習モデルの学習に用いた訓練用データ等に関する情報を、モデルに対する攻撃により得られる場合がある。特に、訓練用データセットにプライバシー保護や個人情報保護を要するデータが含まれている場合には、これらの影響に留意する必要がある。

9.2.1.4　モデルやシステムの解釈/説明機能の誤動作

モデルやシステムに動作の解釈・説明を与える機能がある場合には、これらの出力を誤動作させる攻撃により、システムの利用時品質の低下や透明性の悪化などが起こりえる。

9.2.2　具体的な攻撃手段例

攻撃手段は、攻撃者の持つ知識に応じて、ホワイトボックス攻撃（訓練済み機械学習モデルのパラメータ等を利用する攻撃)、ブラックボックス攻撃（訓練済み機械学習モデルのパラメータ等を全く利用しない攻撃)、グレーボックス攻撃（両者の中間）に分類される。以下では、攻撃手段の例として、一般的な情報セキュリティの攻撃と、機械学習アルゴリズムに対する攻撃について述べる。

9.2.2.1　一般的な情報セキュリティの攻撃

9.2.2.1.1　データの収集プロセスに対する攻撃

データの収集プロセスに対する攻撃により、プライバシー、知的財産、保安、軍事などの法規制や契約に違反するデータが収集され、センシティブ情報を含むデータが機械学習モデルの訓練に利用される可能性がある。例えば、同意なく収集された個人情報や、撮影禁止

の軍事施設の画像データなどが訓練データとして利用される可能性がある。この攻撃と、訓練データについての情報を窃取する攻撃（9.2.1.3 節）の併用により、法規制や契約に違反してセンシティブ情報を漏洩させる可能性もある。

なお、データの収集プロセスに対する攻撃により、機械学習モデルの誤動作を誘発する場合（データポイズニング）については 9.2.1.3 節で述べる。

9.2.2.1.2　ハイパーパラメータ窃取・データ窃取

訓練に用いたハイパーパラメータや訓練用データセットは、システムに対するほかの攻撃の情報として、例えば敵対的攻撃の探索などに利用できる可能性がある。通常の情報セキュリティの観点からは、脆弱性の秘匿によるセキュリティ確保は奨められるものではないが、機械学習においてはその特性上、データを公開しないことによる情報の非対称性を防御に利用し、ホワイトボックス攻撃を防がなければならない場合もあり得る。

9.2.2.1.3　機械学習フレームワークに対する攻撃

機械学習に用いるソフトウェア・フレームワークに脆弱性がある場合、フレームワーク自体や事前訓練済みモデルなどに、マルウェアやバックドアなどを埋め込むことができる可能性がある。

9.2.2.1.4　テスト用データの改竄

テスト用データセットの信頼性を適切に確認しない場合、テスト用データに意図的な偏りを導入する攻撃により、特定のデータに対する誤動作が発見されにくくなる可能性がある。

9.2.2.1.5　モデル窃取

訓練済み機械学習モデルのパラメータは、9.2.2.1.2 節のハイパーパラメータや訓練用データセット以上に、システムに対する敵対的入力による攻撃などを構築する手がかりとなる可能性がある。そのため、訓練済み機械学習モデルの情報を攻撃者に窃取されないことが重要となり得る。

9.2.2.1.6　モデル改竄・解釈/説明機能の改竄

訓練済み機械学習モデルやその解釈/説明機能が改竄された場合、システムの誤動作やバックドアの埋込みなど様々な攻撃が可能になる。

9.2.2.2　機械学習アルゴリズムに対する攻撃

9.2.2.2.1　データポイズニング攻撃（data poisoning attack）

データポイズニング攻撃は、学習に用いる訓練用データに意図的な改変を加えることにより、完全性（integrity）を低下させる攻撃である。大きく分けて、運用時の不特定の入力に対してモデルやシステムの性能を低下させる攻撃［66］と、運用時の特定の入力に対してのみモデルやシステムの性能を低下させる攻撃(バックドア攻撃)がある[48]。後者では、運用時の入力に特定のトリガーとなる情報（例:数字９が書かれているなど）が含まれるときのみ、モデルやシステムを誤動作させる。トリガー情報が含まれない入力に対しては誤動作しないため、バックドア攻撃の検出は一般に難しい。
データポイズニング攻撃における訓練用データの改変は、直接的なデータ改竄だけでなく、データの収集プロセスへの干渉などの手段でも実現できる可能性がある。

9.2.2.2.2　モデルポイズニング攻撃（model poisoning attack）

差分開発・転移学習などで既存の事前訓練済みモデルを学習に利用する場合には、事前訓練済みモデルの中にあらかじめバックドアを仕込んでおくことで、構築した訓練済みモデルに対して不正な動作などを埋め込めることがある。

9.2.2.2.3　回避攻撃（evasion attack）

運用時に機械学習利用システムに特定の改変した入力（敵対的データ、adversarial example）を与えることで、機械学習要素に想定外の誤動作を生じさせる攻撃として「回避攻撃」がある。敵対的データを得る方法としては、特定のモデルについてその内部の情報を元に探索する手法のほかにも、システムの振る舞いから探索する可能性や、様々な種類のモデルに対して誤動作を誘発するような汎用の敵対的データを構築する手法などが研究でされている。

9.2.2.2.4　モデル抽出攻撃（model extraction attack）

運用時に、入力データに対する出力の振る舞いを観察することで、訓練済みモデルと同様の動作をするモデルを抽出し窃取できる場合もある［111][68]。これにより、攻撃者が更なる攻撃の手がかりを得られる場合も有り得る。

9.2.2.2.5　メンバシップ推測攻撃・モデルインバージョン攻撃・性質推測攻撃

同様に運用時に、入力データに対する出力の振る舞いを観察することにより、特定の入力が訓練用データセットに含まれていたかなど、訓練用データに関する情報を窃取する攻撃があり得る。この攻撃により、訓練データに関するプライバシー情報の漏洩などが生じることがあり得る。

具体的には、訓練済みモデルに正常な入力データを与え、モデルの出力を観測することにより、入力データがモデルの学習に用いられた訓練データセットに含まれているか否か（メンバシップ情報）を特定する攻撃として、「メンバシップ推測攻撃（membership inference attack）」[104] があり、様々なブラックボックス攻撃が知られている。さらに、訓練済みモデルからの出力情報から学習データを復元する攻撃として、「モデルインバージョン（model inversion）攻撃」[59] がある。また、訓練データセットが満たす統計的な性質を推測する攻撃として、「性質推測攻撃（property inference attack [42]）」がある。

9.2.2.2.6　解釈/説明機能を誤動作させる攻撃

解釈/説明可能性を実現する技術は、多くの場合ブラックボックスであり、敵対的データによって誤動作を生じることがある。例えば、解釈/説明機能によって出力される説明内容の価値を下げたり、間違った説明を生成したりする攻撃が知られている [54][106]。

9.3　セキュリティに対する対策の考え方

これらの攻撃に対する対策としては主に、周辺の機械学習利用システム全体で取るべき対策、構築プロセス全体で行うべき対策と、特定の攻撃に対する防御策があり、時にはこれらを組み合わせて行うことが必要である。

9.3.1　システム全体での緩和策

- 運用時に攻撃者がモデルへの入力を行える機会を減らす。
  - 運用時の入力データを第三者やインターネット等の公共の場から取得する場合、入力データや提供者の信憑性を確認する。
  - 運用時の入力データを外部環境から取得する場合（例：カメラ画像）、外部環境における攻撃の機会を減らす。運用時に攻撃の機会を減らす手段としては、例

えば、ユーザが一定時間にシステムに入力できるデータの量や回数を制限することが挙げられる。

- 攻撃に利用できる公開情報を減らす。
  - ホワイトボックス攻撃を防ぐために、モデルやシステムの内部の情報（ハイパーパラメータ等）をできるだけ公開しない。
  - 攻撃を防止・緩和するために、モデルの出力に含まれる情報（信頼スコアなど）のうち、製品の動作に必要ない情報をできるだけ秘匿したり、摂動を加えてから公開したりする。
  - 攻撃を検知しやすくしたり、攻撃の成功確率を低めたりするために、事前知識や運用時動作の観察を防止・緩和する。

  ただしこれらの対策では、攻撃の可能性を緩和できても、厳密な耐攻撃性は保証されない。また、透明性や説明可能性との関係の整理が必要である。
- 訓練済み機械学習モデルの外部において、攻撃の検知手法を利用する。

9.3.2 構築プロセスでの緩和策

- 訓練に使うデータや事前訓練済みモデルは、その信憑性を確認し、信頼できる提供者のものを用いる。電子署名等の利用や、信頼できるデータセットを提供するフレームワークの利用により、情報が改変されていないことを確認する。
- 機械学習フレームワークなど使用するソフトウェアの脆弱性情報を把握する。
- 機械学習に限定されない情報セキュリティ対策を行う。

9.3.3 一般的な情報セキュリティ対策

一般的な情報セキュリティの分析とセキュリティリスクへの対応については、ISO27000 シリーズ [10]、ISO/IEC 15408（Common Criteria）[3]、NIST SP800 シリーズ [32]、NIST Cyber Security Framework [33]、情報処理推進機構のガイドライン類などのフレームワークが提供されている。

各種ビジネス分野に特化した観点については、分野ごとの ISAC（Information Sharing and Analysis Centre）が資料やガイドラインを提供している。例えば以下のような ISAC が日本では構築されている。

- Japan automotive ISAC

- ICT ISAC Japan（https://www.ict-isac.jp/）
- Japan Electricity ISAC（https://www.je-isac.jp/）

ISAC が設立されていない業種、例えば生命・財産に関係するビジネスについては、PCIDSS（Payment Card Industry Data Security Standard）[37] などを参考にするとよい。

9.3.4　機械学習要素に特有な攻撃への対策

現時点では、機械学習そのものを対象とするセキュリティリスク分析手法・対策は存在しておらず、個別の技術詳細に関する論文が学会で発表されている段階である。設計開発の参考となるセキュリティのフレームワークに関しては、Microsoft や MITRE らによる Adversarial ML Threat Matrix [34] のような取り組みもあるが、実用的なものがまだない。そのため、開発するシステムで用いられる技術に関連する技術を調査し、技術検討する必要がある。
その作業のため、以下に技術調査に向けた参考情報として、機械学習と関係が深いセキュリティ攻撃の緩和策について記述した文献を紹介する。

9.3.4.1　ポイズニング攻撃

汚染されたデータやモデルを学習させることにより、誤判断に導くポイゾニング攻撃については、Jagielski et al. [66] が参考になる。特に、バックドア攻撃の緩和策については Li et al. [79] が参考になる。

9.3.4.2　回避攻撃

攻撃を目的とする意図的な摂動を入力データに混入（敵対的データ、adversarial example）し、機械学習モデルの誤判断を誘発する回避攻撃については、Goodfellow et al. [62], Krakin et al. [76], Akhtar and Mian [40] が参考になる。

9.3.4.3　モデル抽出攻撃

学習モデルに大量のクエリを発行することで学習モデルと同等の性能を持つ学習モデルを作成するなど、学習モデルを復元する攻撃については、Tramèr et al. [111], Juuti et al. [68]

などが参考になる。

9.3.4.4 訓練データについての情報の摂取

メンバシップ推測攻撃やモデルインバージョン攻撃など、訓練データについての情報を窃取する攻撃の成功確率を下げる手段としては、

- 過学習を防ぐ手法（正則化やドロップアウトなど。訓練データについての情報漏洩をある程度緩和できる。）
- 差分プライバシー保護技術［39］（確率的に摂動を付加するため、性能が低下する。そのため、摂動の規模を決める際は、性能と情報窃取緩和の度合いのバランスに留意する。）
- 敵対的データを用いて信頼スコアを改変する技術［67］

などが挙げられる。

9.3.4.5 解釈/説明機能を誤動作させる攻撃

解釈/説明機能に対する既存の攻撃は、解釈/説明可能性の特定の手法に対して有効なものであり、解釈/説明可能性の複数の手法を併用することにより、防御できる可能性がある。

9.4 （参考）セキュリティチェックリスト

機械学習品質マネジメントガイドラインで規定している内容に即し、各開発段階で実施するべき事項を表 6 に整理した。

表 6: セキュリティチェックリスト

項番	章・節	ガイドラインの記載内容	セキュリティ対応案	備考
	1.3.2	継続的リスクアセスメント	セキュリティリスクアセスメント実施時には、機能安全のリスクアセスメントの結果、優先的に取り扱う項目について攻撃の対象に入れる。	プライバシーや公平性など、システムが利用される対象ビジネスに対応する項目を入れる。
	1.3.3	機械学習利用システムを分業で開発	AI や機械学習に関する規約やガイドラインを参照する場合は、サプライチェーン全体に対し参照する規約やガイドラインの順守を契約に盛り込む。	機械学習品質マネジメントガイドライン、ISO/IEC 規約類など。 GDPR、中国サイバーセキュリティ法などを設計開発するビジネスに応じて参照。 派遣法、下請法など業務委託関連法規も参照。
	1.3.3	不正データの意図的混入による学習結果の汚染のセキュリティリスク	●機械学習品質マネジメントガイドライン第 9 章に例示している Adversarial Example や Poisoning など AI・機械学習に特化する攻撃に関して情報収集を図り、攻撃シナリオ分析を行って脅威と脆弱性の対策を実施する。 ●攻撃手法と防衛手法に関する情報収集を行って定期的に防衛対策の更新を実施する。	●防衛対策全体の見直しの周期は 1 年以内が望ましい。 ●防衛対策は優先度付けし可能な項目から実施する。

	1.5.1	リスク回避性	リスクの対象について脅威・脆弱性を確認・検討すること。 ●安全・生命・経済性の観点から、被害の波及が大きな項目を洗い出し、被害の大きさに応じてリスクを分類し、攻撃の対象として想定し、脅威と脆弱性の分析に利用する。(例：公平性、プライバシー) ●特に公平性、プライバシー、安全を脅かすリスクについては、優先的に取り扱う。	
	1.5.2	AI パフォーマンス	パフォーマンスを攻撃対象とする攻撃シナリオを想定して、脅威と脆弱性を洗い出し対策を講じる。	
	1.5.3	公平性	●第 8 章を利用して攻撃対象となり得る情報（例：性別、肌の色など）について攻撃シナリオを想定し、脅威と脆弱性を洗い出し対策を講じる。●対象とするビジネスに関する情報収集を行って、公平性の攻撃対象となる項目を定期的に見直し、監視と対策の内容を更新する。	第 9 章を参考に分析する。
	1.6.2	セキュリティ・プライバシー	●特に情報資産に含まれる情報の性質が変化することが想定される場合（例えば複数の情報の間で強い相関が生成・消滅する場合）には監視と定期的な対策を実施する。 ●上記で獲得した新しい性質が機微である場合には攻撃を受けた場合の被害が大きくなるため注意が必要。 ●監視によって観測される新しい性質が、国内の個人情報保護法・GDPR・中国サイバーセキュリティ法・US の CCPA などの法規制に抵	●特に GDPR と中国サイバーセキュリティ法は要注意（規制と制裁が非常に厳しいため） ●NIST、ISO/IEC の情報を定期的に収集すること。(2021 年 2 月現在、NIST は ISO/IEC SC42 を参照)

			触しないか確認する。	
	1.6.4	倫理性などの社会的側面	利用するデータについて、法的な権利・公益的政策的な理由によるデータの取得利用に関する規制・契約について法規制をクリアしているか、権利・契約上の調整が必要かを検討する。	日本データベース学会 喜連川･柿沼らの情報を参照
	1.7	内部品質特性	開発するシステムにおける内部品質特性の状況を整理し、攻撃されるデータと被害が大きくなる攻撃のシナリオを想定して、脅威と脆弱性を洗い出して整理し対策する。	開発対象システムの内部品質の状況確認には、当ガイドラインのリファレンスを参考にするとよい。
	1.7	性質 B-1:データセットの被覆性	セキュリティ要件のうち完全性が損なわれる攻撃シナリオを想定して、攻撃を誘発した原因を列挙して脅威と脆弱性を分析し、対策を検討すること。	
	1.7.1	要求分析の十分性	開発するシステムが出力する推論結果、学習モデルが機微情報を包含するかどうかに配慮する必要がある。 BIA(Business Impact Analysis)/PIA(Privacy Impact Analysis)を実施する際にシステム動作フローのリストアップを行って状況の組み合わせを検証しておく（ユースケースレビュー)。	当該情報が機微であるかどうかが時系列的に変化する場合があれば対策が必要
	1.7.1	この段階でリスク分析・故障モード分析などの手法	開発段階のリスク分析は、機能安全など他のリスク分析と並行して実施する。	従来の情報セキュリティと AI に注目した場合のセキュリティの両方について実施する。

	1.7.2	状況の組み合わせ	組み合わせ毎に入出力データへの脅威と脆弱性を確認し対策を検討する。	
	1.7.4	「どのような公平性」が求められているか	第 9 章を参考に対象システムが受ける被害を想定し対策を検討する。	データに対する人為的加工によって公平性が確保できない状態に陥れる攻撃を視野に入れる。
	1.7.5	データに不適切な改変などがされていないこと(信憑性)、データが十分適切に新しいものであること	セキュリティリスクアセスメントの観点のうちの完全性の検証を攻撃シナリオの影響度計算に適用する。	学習に用いるデータが保管される環境のセキュリティリスクアセスメントを行う。
	1.7.5	データ選択妥当性 ラベリングの適切性	攻撃によって発生する被害の洗い出しの際に、データ選択の妥当性やラベリングの適切性を侵害する攻撃を入れる。	被害が発生する原因を洗い出して脅威と脆弱性のリストアップに適用する。
	1.7.5	新しい要件・ポリシーに対応したデータの再整理	データの再整理によって、攻撃による被害の想定が変化する場合は、被害の増減に応じて脅威と脆弱性のリストも改訂する。	データの再整理による波及範囲を確認しておく。
	1.7.6	過学習	被害を想定する際に攻撃対象を過学習に追い込む攻撃を入れる。	例えばワームによる訓練データの追加・加工など
	1.8.1	品質検査	品質検査の検査項目にセキュリティリスクに関する項目を盛り込み、検査中(あらゆる検査項目)にリスク上の問題点を発見した場合は上流工程での再検討に戻る工程上の作業パスを設ける。	リスクの項目としてセキュリティ以外についても必要に応じて入れ込む。
	1.9.1	人間中心の AI 社会原則	新たな攻撃目標となる観点について定期的な情報収集と被害の想定に基づくリスク対策を実施する。	セキュリティの PDCA サイクルに盛り込む。

	2.2.1	IEC 15408	●セキュリティリスク分析・対策立案のフレームワークとしてIEC15408 は代表的であるが、IEC27000 シリーズや IPA のガイドラインなどから要求分析に合うものを選択する。 ●AI 固有の攻撃シナリオの想定と脅威・脆弱性の洗い出し、対策検討を実施する。	
	2.3.1, 2.3.2	機械学習システムの構成、開発の当事者・ロール	AI 固有の攻撃について、アタックツリー・ユースケース分析・情報資産計上によって機械学習固有の脅威・脆弱性を洗い出し、対策検討する。	●例えば 9 章で例示している AI を狙った攻撃についてアタックツリーを想定し、その結果から対策と優先順位を決める。 ●ユースケース分析では、開発関係者以外のプレーヤ（例えばシステム利用者や保守員など）も追加する。
	2.3.3 の第３項	公平性	公平性が侵される被害をセキュリティの被害対象として挙げる。 ●被害の対象となる公平性の項目をリストアップ ●リストアップした公平性項目を狙うアタックツリーを作成する。	
	2.3.3 の第４項	耐攻撃性	セキュリティの耐攻撃性の観点として以下を挙げる。●そもそも攻撃を受けないように防衛する。●攻撃を受けて改ざんなどをされたとしても改ざんの影響が非常に小さく抑え込まれるため実ビジネスに影響がでないようにする。●攻撃を受けても攻撃前の計算状況が残っており、攻撃を受けたらそのときのリソースは全て廃棄して瞬時に残っていたリソースと入れ替えるので実ビジネスに影響がでない。	各テーマの実施規模を検討する。

	2.3.3の第５項	倫理性	倫理性の項目確認と被害が出るシナリオの検討（アタックツリーの検討）を実施する。	
	2.3.3の第６項	堅牢性	堅牢性に被害が出るアタックツリーを想定する。	堅牢性は robustness として英訳されている。
	2.3.4の第１項	システムライフサイクル	システムライフサイクルの各段階においてリスクアセスメント（ゲート）を設ける。	実施する内容・規模については、定期的な実施（1 年以内）を念頭に実施可能な項目を優先順位に従って実施する。（実施可能な規模に収まらない場合、形骸化する可能性が高い。）
	2.3.5	利用環境に関する用語	学習、推論のような AI・機械学習に特化したシステム構成要素、学習モデルが保管される環境については、従来の情報セキュリティのみならず、AI に固有の要件が必要となるので対応する。	
	2.3.6	機械学習構築に用いるデータなどに関係する用語	同上	
	2.3.7の第４項	運用期間中にデータの収集と追加的な機械学習の訓練を行い、随時・適時に訓練済み機械学習モデルの更新を行う	一般情報セキュリティでは、システムが利用するデータが運用中に新たな特性（公平性やプライバシーなど）を獲得することを想定していない。追加的な学習によって保管しているデータが新たな特性を生じていないかを定期的に監視し、必要に応じて対策検討することを盛り込む。	

	3.1	表 1:人的リスクに対する AI 安全性レベルの推定基準 表 2:経済リスクに対する AI 安全性レベルの推定基準	表の各セルについて、想定される影響の被害が大きな案件のアタックツリーを想定し、優先的に対応する。	IT システムのみでビジネスソリューションが構築される場合は安全性を考慮しない場合があり得るが、衝突軽減ブレーキなど安全性に関わる組み込みシステムなどでは安全性の検討は必須である。
	3.3	公平性	個人の権利や資産が被害を受けるケースを想定してアタックツリーを作成して脅威と脆弱性の検討項目に追加する。	特に追加学習と関連するシナリオには注意が必要
	4.1.1 図 11	複数の運用段階を伴う開発プロセスの取扱い	●PoC 毎に 要求仕様に沿うかの検証を実施する度にリスク要因(隠れ相関など)を検証する。 ●品質検査や運用の性能監視においてもリスク要因を監視し、対策検討する仕組みを入れる。	
	4.2.1 図 12	機械学習構築の段階におけるプロセスモデル	●データセット・モデル・テストの設計段階で機微情報の取り扱い、新しい機微情報の生成を確認し、脅威と脆弱性を分析して対応を盛り込む。 ●反復訓練、品質確認・検証の作業で出力される推論結果について、機微情報の取り扱い、新しい機微情報の生成を確認し、脅威と脆弱性を分析して対応を盛り込む。	
	4.2.1 図 13	訓練用前処理の一例	前処理の各工程の中で機微情報の取り扱い、新しい機微情報の生成を確認し、脅威と脆弱性を分析して対応を盛り込む。	

	4.2.1.1	ML 要求分析フェーズ	●学習データを情報資産として計上し、入出力データの性質の整理・具体的な構築への要求として再整理することで表れるデータ間の関係(例：相関)のうち機微なものを被害の対象として想定し、脅威と脆弱性を洗い出して対応を検討する。 ●データの性質や具体的なデータセットそのものに基づいて決める品質要求について、攻撃の被害を想定して脅威と脆弱性を洗い出して対応を検討する。	
	4.2.1.2	訓練用データ構成フェーズ	●データセットを情報資産として計上し、データセットが保管しているデータ種別間の関係（例：相関）やデータの性質（例：公平性やプライバシー）を精査して攻撃を受けた場合の被害を想定する中から脅威と脆弱性を洗いだし対応を検討する。 ●前処理後のデータの間で新しい関係（例：公平性やプライバシー）が生成していないか精査し、攻撃を受けた場合の被害を想定する中から脅威と脆弱性を洗い出し対応を検討する。	
	4.2.1.3	反復訓練フェーズ	●機械学習モデル・ハイパーパラメタ・実装用学習モデルを情報資産として計上し、データセットが保管しているデータ種別間の関係（例：相関）やデータの性質（例：公平性やプライバシー）を精査して攻撃を受けた場合の被害を想定する中から脅威と脆弱性を洗いだし対応を検討する。●前処理後のデータの間で新しい関係（例：公平性やプライバシー）が生成していないか精査し、攻撃を受けた場合の被害を想定する中から脅威と脆弱性を洗い出し対応を検討する。	

	4.2.1.4	品質確認・検証フェーズ	テスト用データセット・テスト結果(テストの確証を含む)・誤推論を起こしたデータを情報資産として取り扱い、攻撃の参考となる情報の改ざんや窃盗による被害を想定し、脅威と脆弱性を洗い出して対応を検討する。	
	4.2.2	システム構築・統合検査フェーズ	機械学習利用システムのうち機械学習要素以外の部分については、ISO/IEC 27000 シリーズ、ISO/IEC 15408 Common Criterial、IPA が提供するガイドライン・フレームワークやツール類を使用してセキュリティリスクアセスメントを実施し、その結果に基づいて対応を図る。	●ビジネス分野に特化した要件については各種 ISAC、ISAC が設立されていない場合は PCI DSS の要件などを参考にアセスメント方法を検討する。 ●生命、安全、財産に関わる場合はガイドライン・フレームワークの選択に注意する。
	4.3	品質監視・運用フェーズ	●学習データに含まれるデータ間の関係(例：相関)や性質(例：公平性やプライバシー)、ハイパーパラメタを情報資産とし、被害を想定した定期的な監視を行って対応を図る。 ●データ間の関係と性質の観点についても定期的な見直しを行う。	
	5.1.1.1	安全性機能が必要な場合	安全性の検討の中で重篤な被害が想定される事象は攻撃対象となった場合の被害が大きくなるため、優先的な対応を図る。	AI・機械学習を用いたシステムに関する安全を検討する。
	5.1.1.3	リスクシナリオ検討	高リスクの事象については、攻撃対象となった場合の被害が大きくなるため、優先的な対応を図る。	AI・機械学習に深く関連するリスクを洗い出し対応の仕方を検討する。
	5.1.1.4 5.1.1.5	利用時品質特性	採用する品質メトリクスが測る観点について被害が想定される場合は、脅威と脆弱性の分析を行って対応を図る。	外部品質特性を流用する場合は外部品質特性に関する被害の想定と脅威・脆

				弱性の分析と対応に従う。
	5.1.2	物的・人的損害リスク	高リスクの事象については、攻撃対象となった場合の被害が大きくなるため、優先的な対応を図る。	AI・機械学習に深く関連するリスクを洗い出し対応の仕方を検討する。
	5.2.1	開発依頼者、開発協力者の双方で合意形成	合意形成時に AI・機械学習に深く関係する攻撃事例から当該ビジネスに関係のあるものを抽出して、セキュリティ上のリスクをリストアップし、リスク対策の検討と実施・リスクの転嫁・リスクの承認を選択する。	●リスクの転嫁：セキュリティを担うシステムや機能を他者のもので実施してセキュリティ要件を他者に委ねるもの(例：AWS 上でのソリューション開発とセキュリティ対策) ●リスクの承認：経営判断として当該のセキュリティリスクを受け入れ対策も転嫁も実施しないもの
	5.2.2	作業内容の明確化	作業工程の進捗のうち、以下の段階においてはセキュリティリスクアセスメントが実施され、対策の実施・転嫁・承認などが判断されることを確認する。 ●要件分析結果の承認 ●ソフトウェアの外部仕様承認時 ●ソフトウェアの動作検証仕様書承認時 ●ソフトウェアの検証結果承認時	特に AI・機械学習に深く関連する KPI の項目について確認する。
	5.2.3 3)	運用時品質管理方法	●学習データに含まれるデータ間の関係(例：相関)や性質(例：公平性やプライバシー)、ハイパーパラメタを情報資産とし、被害を想定した	

			定期的な監視を行って対応を図る。 ●データ間の関係と性質の観点についても定期的な見直しを行う。	
	5.3	差分開発	既存のソフトウェア部品を再利用する場合、セキュリティリスクアセスメントについても新しいシステムが利用される状況(文脈・コンテキスト)において実施する必要がある。これはアセスメントの結果実施するリスク対策についても同様である。	
	6	品質管理の特性軸	問題領域分析の十分性、データ設計の十分性、データセットの被覆性・均一性、データの妥当性、機械学習モデルの正確性・安定性、プログラムの信頼性、運用時品質の異時性に対して開発者が決めた品質レベルの設定・設計の根拠について、最も重篤な被害が起こることを想定し、攻撃シナリオに基づいた脅威と脆弱性の洗い出しを行って対応を実施する。	各観点において、開発者が決めるレベル(LV)の設定とその根拠に基づいた脅威と脆弱性の洗い出しと対応を図る。
	6.1.2.1	「属性」の候補	●属性が機微情報(例：肌の色、人種など)でないかを確認し、機微である場合には、当該属性の使用の取りやめ、匿名化など対策の要否を検討する。●属性間の関係（例：相関）や属性の性質（例：公平性やプライバシー）を精査し、攻撃を受けた場合の被害を想定する中から脅威と脆弱性を洗いだし対応する。	
	6.1.3	リスク回避性レベル	リスクが高まる要因は攻撃者の攻撃対象となった場合の被害が大きくなる原因となるため、要求事項に沿った分析の結果を参考に脅威と脆弱性を洗い出し対応する。	

	6.2	データ設計の十分性	設計された網羅性を侵害する攻撃を想定して脅威と脆弱性を洗い出し対応する。	
	6.3	データセットの被覆性	セキュリティ要件のうち完全性が損なわれる攻撃シナリオを想定して、攻撃を誘発した原因を列挙して脅威と脆弱性を分析し、対応する。	
	6.5.2.1	ラベリングのポリシーの統一・精査	攻撃によってデータがポリシーに反する状態に陥る攻撃シナリオを想定し、関係する脅威と脆弱性性を洗い出して対応する。	
	6.5.2.2	データセットの整合性チェック・再チェック	機能要件や使用環境が変わることに対応して実施する評価や、作業を外注する場合のプロセスについて、セキュリティリスクアセスメントの対象とすることを視野に入れる。	情報セキュリティの観点とサプライチェーンセキュリティの観点を入れるとよい。
	6.5.2.3	ロングテールの扱いと、計測ミス・外れ値の判断	保管されているデータが、ロングテール・計測ミス・外れ値の取り扱いポリシーに反する状態に陥る攻撃シナリオを想定して、関係する脅威と脆弱性を洗い出して対応する。	
	6.5.2.4	データ汚染への対応	●データが汚染される攻撃シナリオを想定して関係する脅威と脆弱性に対応する。 ●データ汚染の検出方法を検討しておく。 ●サイバーセキュリティ以外のセキュリティ要件についてシステム構成と開発対象システムの利用シーンに対応する。	利用シーンへの対応には、開発システムが利用されるビジネスモデルに応じたガイドライン（ISAC や省庁が提供するもの）を利用するとよい。ガイドラインが提供されない場合は金融や重要インフラに向けたガイドラインを参考にするとよい。
	6.5.2.5	最新性	訓練データの最新性が侵害される攻撃シナリオを想定して関係する脅威と脆弱性を洗い出して対応する。	

	6.5.2.6	工程管理を行う体制・仕組みづくり	サプライチェーン全体に適用するセキュリティリスクアセスメントルールを制定し、ルールに則った定期的な監査と監査証跡の保管を契約に盛り込む。	
	6.5.3	要求事項	本表の 6.5.2 の各指摘事項について反映する。	
	6.6.2	指標の相対的な振る舞い	●学習データセットに含まれる入力に対する振る舞いについて、学習モデルに及ぶ被害（改ざんなど）を確認する手段を準備し監視する。 ●学習モデルに改ざんなどの被害が発生した場合の原因を想定して対応を準備しておく。	学習データセットに含まれる入力に対する振る舞いについて証跡（評価指標の記録）を残すとよい。
	6.7.2	安定性の評価と向上	●データセットに含まれない入力に対する反応について、学習モデルに及ぶ被害（改ざんなど）を確認する手段を準備し監視する。 ●学習モデルに改ざんなどの被害が発生した場合の原因を想定して対応を準備しておく。	データセットに含まれない入力に対する反応を反復訓練フェーズ・品質確認評価フェーズ・品質監視運用フェーズそれぞれについて証跡を残すとよい。
	6.8.1	オープンソースの実装	公開されている脅威や脆弱性情報への対応を図る。	CWE や CVE を参照するとよい。
	6.9.2	品質劣化	運用時の品質劣化の原因が攻撃者による攻撃である場合の対応プロセスとシステム基盤を作っておく。	
	7.1.2	リスク要因の推定	リスクの項目は被害の程度によって分類し、重篤な被害が出るものについては、攻撃者の被害の対象として想定し、攻撃シナリオを列挙して脅威と脆弱性を洗い出し対応を検討する。	
	7.1.3	最終的な実装形態をある程度	Forecast を行うべき観点を運用監視にも適用し、当初のビジネスの想	

		想定した分析	定からの変化(コンセプトドリフト)と攻撃を見分ける方法を合わせて検討する。	
	7.2	データ設計の十分性	6 章「品質管理特性軸」のセキュリティ対応案を参照	
	7.3	データセットの被覆性	6 章「品質管理特性軸」のセキュリティ対応案を参照●データ整理段階における追加的検査やテスト段階での追加的検査によって採用した特徴量と見落とした特徴量の隠れ相関などの性質を精査し反映する。●特徴量の性質の精査は監視処理などで定期的に実施する。	
	7.3	データセットの被覆性	セキュリティ要件のうち完全性が損なわれる攻撃シナリオを想定して、攻撃を誘発した原因を列挙して脅威と脆弱性を分析し、対策を検討すること。	
	7.4	データセットの均一性	6 章「品質管理特性軸」のセキュリティ対応案を参照	
	7.5.1	データ収集ポリシー	6.5.2.1 章のセキュリティ対応案を参照	
	7.5.1	データ収集ポリシーと合わせて要件定義が更新されていないか、その更新に合わせてここまでの段階（内部品質 A-1～B-2 など）で確認した内容についての再確認が必要で無いかなどのアセスメント	ポリシーや要件定義が更新されている場合は、セキュリティ上被害を受ける想定も変化している恐れがあるので確認し、変更が必要なら脅威・脆弱性の分析と対策内容にも反映する。	
	7.6.1.3	ファズ・テスティング	入力データが脅威・脆弱性に関連する場合は、ファジングデータを入	

			力するテストも必要に応じて実施する。	
	7.6.1.5	テスト入力の自動生成	必要に応じ、AI・機械学習に深く関係する攻撃のうち、Model Evasion、Model Extraction、Adversarial Example を対象とする検証項目を入れる。	
	7.6.2.2	正則化	ドロップアウトでは、非活性とするニューロンの割合がハイパーパラメータとなる。このようなネットワーク構造を決めるハイパーパラメータを情報資産として挙げる。	7.5.2 章の他の項目についても左記と同様の検討を行う。
	7.6.2.3	敵対的データ生成	必要に応じて敵対的データ生成による攻撃を想定した攻撃シナリオから脅威と脆弱性を想定し、対策を検討する。	
	7.6.2.6	敵対的訓練	必要に応じて敵対的訓練を利用した攻撃を想定した攻撃シナリオから脅威と脆弱性を想定し、対策を検討する。	
	7.7.3	脆弱性情報リスト	CVE や CWE は一般情報セキュリティの脅威・脆弱性情報として大変参考になるが、AI・機械学習に深く関連する観点が入っていないため、別途開発するシステムに応じた AI・機械学習に関するセキュリティ観点を設定しシステムの評価に用いる。	
	7.7.5	ソフトウェア更新	近年の高度な攻撃手法のひとつとしてソフトウェア更新に関するものがある。マルウェアの混入・トランザクションの奪取によって、機械学習要素に注目した攻撃が行われる可能性も想定されるので、攻撃シナリオを想定し検証しておく。	
	7.8.1	モニタリング	攻撃を受けた場合に重篤な被害が出る可能性のある特徴量間の関係（例：隠れ相関）特に機微である場合に着目したセキュリティ観点での	

			監視も合わせて実施する。	
	7.8.2	コンセプトドリフト	モニタリングと同様	
	7.8.3	再学習	モニタリングと同様	
	8.1.1	倫理性と公平性	倫理性と公平性を侵害された場合の被害と攻撃シナリオを想定して、原因に繋がる脅威と脆弱性を洗い出し、対策を設ける。	
	8.1.2	社会的要請	法規制・社会原則・ガイドライン・国際標準については、保証を請求されたりビジネスへの波及が大きな場合のる被害を想定して、攻撃シナリオの想定に組み込んでおく。(攻撃シナリオの想定には 8.4 章を参照するとよい。)	

10. （参考）関連する文書類に関する情報

本章の内容は参考（informative）である。

10.1 他のガイドライン類との相互関係

10.1.1 経済産業省の AI 契約ガイドライン

経済産業省が発表した「AI・データの利用に関する契約ガイドライン」[28] は、受発注・準委託などの関係により連携・分担して機械学習 AI などを含むシステムを開発する際における事業者間の契約に関する留意点をまとめたものである。同契約ガイドラインは主に開発の事業者間での契約の在り方や責任の所在などを明確にするものである一方で、本ガイドラインは最終システムの利用者に対してサービス提供者が提供する品質の在り方を整理したものである。本ガイドラインの立場からは、本ガイドラインの規定する製品・サービスの利用時品質は、契約ガイドラインに役割として掲げられた開発のステークホルダ全員が共同してシステムに対して作り込み、製品・サービスの利用者に対して提供するものである。この品質を具体的にどのように事業者間で分担して実現するか、その際の契約や責任の分担をどのようにするかは、基本的には本ガイドラインの直接の対象外であり、契約ガイドラインなどに基づきステークホルダ間で合意をすれば良いものである。一方で、そのステークホルダ間の分担の際に着目し情報交換する品質上の技術的事項については、本ガイドラインがその検討の土台となり得る。一例としての役割分担の可能性については、5.2 節にも若干の分析を行った。

なお、当該契約ガイドラインにおいては、開発プロセスにおける受発注関係において「非ウォーターフォール型モデルが適合する」とされている。一方で、同ガイドラインの 46 ページにおいても③「開発」のほかに④「追加学習段階」などの前後段階のプロセスに着目していることから、本ガイドラインではシステムの企画から運用後廃棄までの広範囲をシステムライフサイクルプロセスの全体として捉え、モデルとしてはウォーターフォール型との混合モデルを基本として、28 ページの「図 6: 混合型機械学習ライフサイクルプロセスの概念図」として整理した。契約ガイドラインの推奨する「探索的段階型」の開発プロセスとの関係では、図 6 の中央の開発フェーズと、契約ガイドラインの「開発段階」が基本的

に対応する。

10.1.2　QA4AI ガイドラインとの関係

AI プロダクト品質保証コンソーシアムは、2020 年 8 月に「AI プロダクト品質保証ガイドライン 2020.08 版」[38] を公開している。同ガイドラインは、「AI プロダクトの品質保証に対する共通の指針を与える」ものとして、AI プロダクトの品質保証において考慮すべき軸として

A)　「Data Integrity」
B)　「Model Robustness」
C)　「System Quality」
D)　「Process Agility」
E)　「Customer Expectation」

の 5 つの軸を提案し、それぞれの軸に対するチェックリストや具体的な品質管理技術のリストなどを提示している。

本ガイドラインとの関係では、これらの品質保証軸のうち A) B) C) の 3 つは、本ガイドラインの 1.7 節（18 ページ）に掲げる内部品質に対応していると考えられる。従って、同ガイドラインが具体的に列挙する技術も、7 章で提示する内部品質特性毎の品質管理手法と対応し、相互に補完しつつ実際に品質を実現するエンジニアに対する示唆を与えるものと位置づけられる。

また D) E) の保証軸は、本ガイドラインでは明確な品質管理軸として設定していないが、本ガイドラインが想定する適用プロセス（5.1 節）やそれを含む顧客とのビジネスプロセスなどを通じて実現されるものと考えられる。

全体として、QA4AI コンソーシアムのガイドラインは、実際に機械学習 AI を作るエンジニアにとって機械学習要素の内部品質を改善し作り込むための技術の可能性を見つけ出すための参考書（リファレンス）として有益である一方、本ガイドラインは機械学習 AI を含むシステム全体を企画開発する企業などが、システム全体の利用時品質をライフサイクルプロセス全体を通じて確保するための必要事項を網羅的に分析し可能な限り列挙することを意図しているもので、相互に補完関係にあると考えられる。

表 7: QA4AI ガイドライン（2020 年 8 月版）との関係の分析

・

本ガイドラインの内部品質特性	QA4AI ガイドラインのチェックリスト
A-1 問題領域分析の完全性	2.2.1 Data Integrity (b) 学習データの妥当性 (b.i) 2.2.2 Model Robustness ⑲数理的多様性、意味的多様性、社会的文化的多様性 などを考慮し、十分に多様なデータで検証を行ったか
A-2 問題に対する被覆性	2.2.1 Data Integrity (a) 学習データの量の十分性 (a.i) (a.ii), (a.iii) (b) 学習データの妥当性 (b.i)
B-1 データセットの被覆性	2.2.1 Data Integrity (d) 学習データの適正性 (f) 学習データの性質の考慮
B-2 データセットの均一性	2.2.1 Data Integrity (d) 学習データの適正性
B-3 データの妥当性	2.2.1 Data Integrity (b) 学習データの妥当性 (b.ii), (b.iii) (c) 学習データの要件適合性 (c.ii) (e) 学習データの複雑性 (g) 学習データの値域の妥当性
C-1 モデルの正確性	2.2.1 Data Integrity (i) 検証用データの妥当性 2.2.2 Model Robustness (a) モデルの制度の十分性 (b) モデルの汎化性能の十分性 (c) モデルの評価の十分性 (d) 学習過程の妥当性 (e) モデル構造の妥当性 2.2.3 System Quality (a) システムによる提供価値の適切性 (a.ii)
C-2 モデルの安定性	2.2.2 Model Robustness (b) モデルの汎化性能の十分性 (d) 学習過程の妥当性

	(e) モデル構造の妥当性 (f) モデルの検証の妥当性 (g) モデルの頑健性
D-1 プログラムの信頼性	2.2.1 Data Integrity (k) データ処理プログラムの妥当性 2.2.2 Model Robustness (k) プログラムとしてのモデルの適切性
1.7.9 運用時品質の維持	2.2.1 Data Integrity (j) オンライン学習の影響の考慮 2.2.2 Model Robustness (i) モデル更新に対する検証の十分性 (j) モデルの陳腐化への考慮 2.2.3 System Quality (i) システムの品質低下への考慮
外部品質に対応する項目	2.2.3 System Quality (a) システムによる価値提供の適切性 (c) システム評価単位の妥当性 (d) 事故による影響の抑制 (e) 事故発生の回避性 (f) AI の影響度の抑制

10.2 AI の品質に関する国際的取り組みとの関係

現在、AI の品質については、以下に述べるようなさまざまな国際的な取組がなされている。本ガイドラインの検討においては、これらの取り組みを踏まえて活用できる部分を取り入れる。また、本ガイドラインにより得られた優れた知見は国際標準化へ積極的に提案していく。

10.2.1　品質、安全性

ISO/IEC JTC 1/SC 42/WG 3 では品質や安全性に関するアドホック検討が立ちあげられ、特に AI の品質特性、品質保証技術について、既存標準（ISO/IEC 25000 (SQuaRE) [7][8], ISO/IEC/IEEE 29119-4 (試験技術) [11］の他、機能安全（IEC 61508 [12], ISO 26262 [9]）などの既存標準とのギャップ分析が実施されている。SC42 ではこれ以外にもデータ品質などについて、様々な議論が立ち上がりつつある。

EU においては、2018 年に欧州委員会が、適切な倫理的・法的枠組みの確保を含む欧州 AI 政策指針［23］を発表し、AI-HLEG（AI 専門家グループ）[25］を設置した。2019 年に AI-HLEG が、人間の尊重や公平性などを 4 倫理原則とする欧州 AI 倫理ガイドライン［26］を策定した。2020 年に欧州委員会が、高リスク産業領域かつ高リスク用途の AI システムに訓練データや精度などに 6 要件を求める欧州 AI 白書［24］を発表した。欧州 AI 白書へのパブコメを経て、2021 年 4 月に欧州委員会が、データや精度などに 8 要件を求める欧州 AI 法案［22］を発表した。欧州委員会の発議を受け、欧州議会で審議される過程で、欧州 AI 法規制の内容は変化する可能性がある。世界初の法的拘束力のある AI ハードロー案である欧州 AI 法案が、最終的に保護されるべき目的（ゴール）を策定し、本ガイドラインが、ゴールを達成する技術的手段を提供できる可能性があるが、それを実現するのは、今後の将来課題である。欧州 AI 法案は、データとデータガバナンス要件（Data and data governance）で、高品質の訓練・バリデーション・テストデータを開発に用いること、及び、地理的・行動的・機能的使用環境の特徴・特性・要素を考慮することを求めている。欧州 AI 法案は、これらの要件が関係する、本ガイドラインの B-1: データセットの被覆性と B-2: データセットの均一性の、必要性を指摘している。また、欧州 AI 法案は、精度・頑健性・サイバーセキュリティ要件（Accuracy, robustness and cybersecurity）で、ライフサイクルを通じて一貫した性能発揮、精度の水準及び基準の宣言、敵対的データなどに対するサイバーセキュリティの担保を求めている。欧州 AI 法案は、これらの要件が関係する、本ガイドラインの C-1: 機械学習モデルの正確性と C-2: 機械学習モデルの安定性の、必要性を指摘している。

10.2.2　透明性（transparency）

EU においては、欧州 AI 倫理ガイドラインや欧州 AI 法案が透明性への要求条件をまとめ

ており、EU 加盟国を中心に国際的な影響力を持つと考えられる。2019 年 4 月に AI-HLEG は、透明性を担保するためのチェックリストを提示し、実際に企業との実証実験を通じて検証する作業を行い、2020 年に、透明性や説明責任などの 7 要件を含む信頼できる AI の自己評価リスト［27］を発表した。欧州 AI 法案では、透明性とユーザへの情報提供要件（Transparency and provision of information to users）として、動作の透明性（処理過程を利用者が理解・制御できるように、透明性のある動作を設計・開発すること）が求められている。

一方 IEEE では現在、IEEE P7001（transparency of autonomous systems）[17] を検討しており、用語の定義や考え方において今後の標準化に一定の影響力を持つ可能性がある。5 種類の関係者（ユーザ、事故調査委員会他）に対し 6 種の透明性レベル（0〜5）を規定しており（数字が大きくなると必ずしも基準が厳しくなるものではない)、パイロット認証プロジェクト ECPAIS にて適合性認証の実証が進められている。

ISO では、既に述べた ISO/IEC JTC 1/SC 42 の WG 3（trustworthiness）において、文書 TR24028 [6] にて透明性に関する用語や概念を記載している。

10.2.3　公平性（バイアス）

EU では、バイアス含む AI の ELSI 問題に関するハイレベルな原則をまとめている。また、欧州 AI 法案では、データとデータガバナンス要件（Data and data governance）として、データセットが関連性・代表性・無誤差・適切な統計的特性を持つこと、及び、バイアスの監視・検知・補正を求めている。

前述した IEEE P7003 (algorithmic bias) [19] においては、アルゴリズムの開発において「負のバイアス」（人種や性別など法的に禁じられている差別、法的ではない差別を共に想定）を特定し、システムの立案から運用にいたるライフサイクルでバイアスを許容範囲に抑え込む方法論を規定しており、適合性認証を実現するためのパイロットプロジェクト ECPAIS (Ethics Certification Program for Autonomous and Intelligent Systems）での実証実験が進められている。

ISO/IEC JTC 1/SC 42/WG 3（trustworthiness）においても、TR 24028 (Overview of trustworthiness in Artificial Intelligence) [6], TR 24027 (Bias in AI systems and AI aided decision making) [5] などの文書の作成作業が現在進められている。

10.2.4 その他の機械学習品質マネジメントの観点

IEEE P7000 シリーズにてプライバシー［18］や Nudge（行動の誘導）他の検討が、また ISO/IEC JTC1/SC42 ではガバナンス他の検討が行われている。欧州 AI 法案では、既出の要件の他に、リスク管理システム（risk management system）、技術文書の公開（Technical documentation）、運用時の記録保持（Record-keeping）、人間による監視（Human oversight）の要件を求めている。

11. （参考）分析に関する情報

本章では、主に 1.7 節および 6 章に掲げた内部品質の特性軸を導き出すに至る分析の過程を参考情報として整理する。

本章の内容は参考（informative）である。

11.1 リスク回避性に対する内部品質特性軸の分析

まず初めに、リスク回避性の外部品質軸について、品質の劣化モードの特定を行った。この品質劣化は、「機械学習要素による個々の推論結果（極端に言えば、ニューラルネットワークの結果のベクトル値）が［正しくない・思わしくない・望ましくない］」ということに起因する。そこで抽象的な機械学習要素を想定し、機械学習要素が「望ましくない」答えを出す可能性について、2 分木的かつ抽象的な故障モードの分析を、Fault Tree Analysis (FTA) に類似した考え方に基づいて行った。その分析結果を図 21 に示す。

この分析はあくまで故障モードの網羅的な原因分解を目的としたものであり、実際に判断に失敗したケースについて、これらのどれが原因であったかについては、いわゆるオラクルの視点がなければ原因が特定できないことがある点は注意が必要である。しかし一方で、対策の観点からは、例えオラクルの視点がなくても、全ての故障原因に対して対策を行ってしまえば、全ての故障モードの可能性を減らすことができると言うことができる。もちろん実際にはこれらの各項を完璧に実現することは不可能であるし、また訓練入力データに含まれる誤差や、超多次元空間データにおける数学的課題などの諸々のギャップを意図的に無視しているが、全体として品質向上プロセスの方向性としてこれらの項に着目すれば、ギャップがあっても品質の向上に十分寄与することを意図している。この図の性質 1〜7 を、故障原因から「故障しないための特性」へと反転させたものが、内部品質の 6.1〜6.9（6.4 を除く）に対応している。

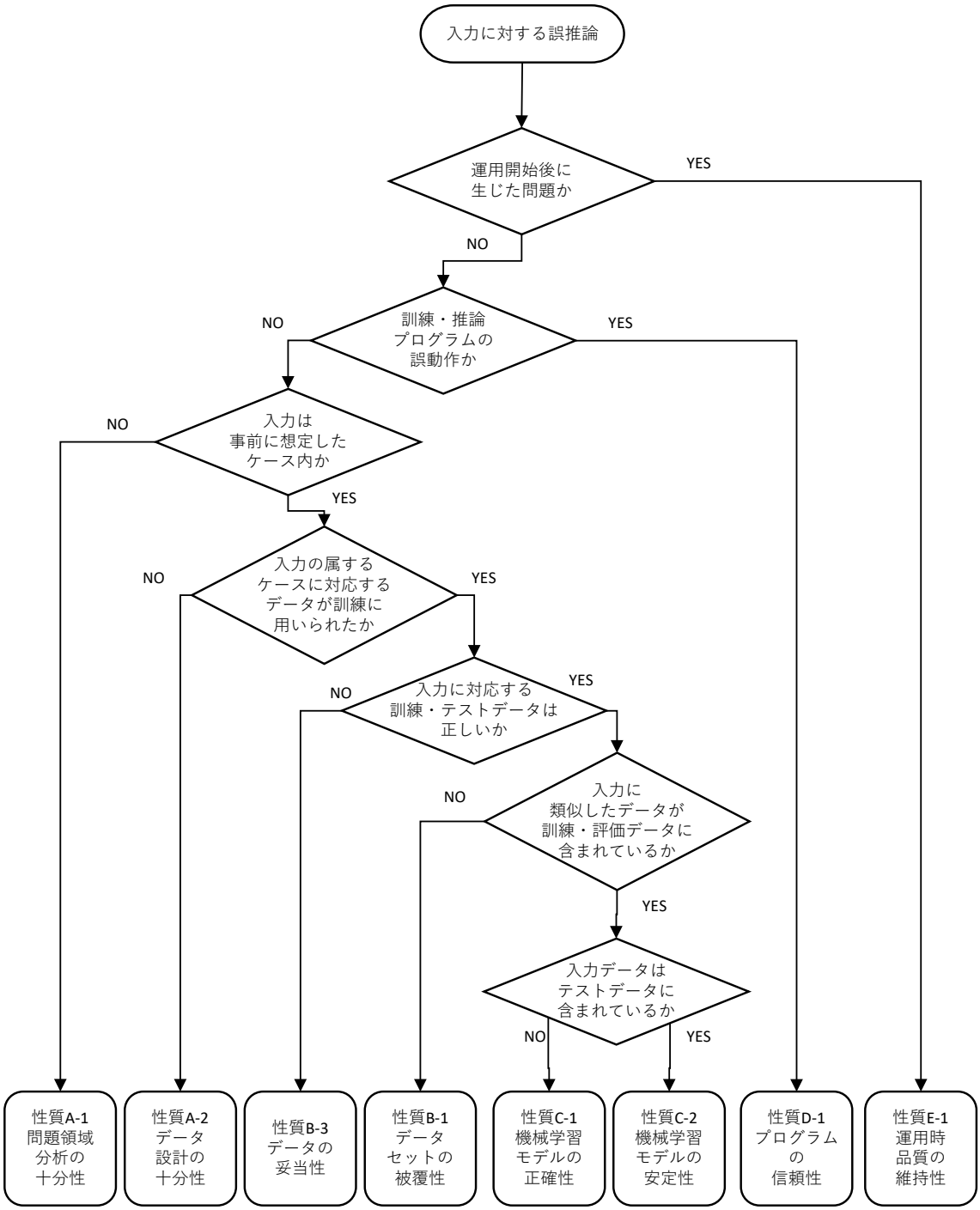

図 21: 機械学習に対する故障モード解析例

11.2 AI パフォーマンスに対する品質管理軸の分析

前節の「リスク回避性」に対する内部品質管理軸を踏まえて、続いて AI パフォーマンス

に対しても同様の分析を行った。

AI パフォーマンスも極論としては「機械学習要素による個々の推論結果（極端に言えば、ニューラルネットワークの結果のベクトル値）が［正しくない・思わしくない・望ましくない］」ということにその劣化が起因し、リスク回避性との差異はその個々のケースの重み付けの差異による全体の評価関数の違いと考えられる。このことから、基本的には同じ内部品質特性軸をそのまま利用できると考えられる。

ただし、リスク回避性においては、それぞれの想定リスクケースに対して対策としての学習データを十分に割り当てることに重点を置いており、学習データ全体としてのバランスを意図的に考慮していない。これは、特に発生頻度が稀ながら重篤なリスクケースに対して、均一に学習データをサンプリングしたのでは十分な学習が得られないことを想定しているが、AI パフォーマンスの観点からは、このような「意図的に偏って強化した学習データ」は全体性能の劣化を及ぼすことがあることが既に知られている。このことから、AI パフォーマンスを主に想定した内部品質軸として、6.4 に掲げる「データの均一性」を追加した。この結果が、6 章に掲げた 9 つの内部特性である。

12. 図表

本章の内容は参考（informative）である。

12.1 外部品質特性と内部品質特性の対応表

表中の数値・記号は、それぞれ該当する節の「品質レベルごとの要求事項」に掲げられたチェック項目のレベルを示す。記号「+」は、追加的な検討を今後行うことを示す。太字の項目は、特に注意を要する項目を示す。

AISL	0	0.1	0.2	1	2	3	4
A-1 問題領域分析の十分性		1	2	3	+	+	+
A-2 データ設計の十分性		1	2	3	+	+	+
B-1 データセットの被覆性		1	2	3	+	+	+
B-2 データセットの均一性		S1	S2	S2	+	+	+
B-3 データの妥当性		1	2	3	+	+	+
C-1 モデルの正確性		1	2	3	+	+	+
C-2 モデルの安定性		1	2	3	+	+	+
D-1 プログラムの健全性		1	2	3	+	+	+
E-1 初期品質の維持性		1	2	3	+	+	+

AIPL・AIFL	PL 0	PL 1	PL 2	FL 0	FL 1	FL 2
A-1 問題領域分析の十分性		1	2		1	2
A-2 データ設計の十分性		1	2		1	2
B-1 データセットの被覆性		1	2		1	2
B-2 データセットの均一性		E1	E2		**E2**	E2
B-3 データの妥当性		1	2		1	2
C-1 モデルの正確性		1	2		1	2
C-2 モデルの安定性		1	2		1	2
D-1 プログラムの健全性		1	2		1	2
E-1 初期品質の維持性		1	2		**2**	**3**

13. 参考文献

13.1 国際規格

[1] ISO 13849-1:2015: Safety of machinery — Safety-related parts of control systems — Part 1: General principles for design.

[2] ISO/IEC/IEEE 15288:2015: Systems and software engineering — System life cycle processes.

[3] ISO/IEC 15408-1:2009: Information technology — Security techniques — Evaluation criteria for IT security — Part 1: Introduction and general model.

[4] ISO/IEC DIS 22989: Information technology — Artificial intelligence — Artificial intelligence concepts and terminology.

[5] ISO/IEC DTR 24027: Information technology — Artificial Intelligence (AI) — Bias in AI systems and AI aided decision making.

[6] ISO/IEC TR 24028:2020: Information technology — Artificial intelligence — Overview of trustworthiness in artificial intelligence.

[7] ISO/IEC 25010:2011: Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models.

[8] ISO/IEC 25012:2008: Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Data quality model.

[9] ISO 26262-1:2018: Road vehicles — Functional safety — Part 1: Vocabulary.

[10] ISO/IEC 27000:2018: Information technology — Security techniques — Information security management systems — Overview and vocabulary.

[11] ISO/IEC/IEEE 29119-4:2015: Software and systems engineering — Software testing — Part 4: Test techniques.

[12] IEC 61508-1:2010: Functional safety of electrical/electronic/programmable electronic safety-related systems - Part 1: General requirements.

[13] IEC 61508-3:2010: Functional safety of electrical/electronic/programmable electronic safety-related systems - Part 3: Software requirements.

[14] IEC 61508-4:2010: Functional safety of electrical/electronic/programmable electronic safety-related systems - Part 4: Definitions and abbreviations.

[15] IEC 62278:2002: Railway applications - Specification and demonstration of reliability, availability, maintainability and safety (RAMS).

[16] IEC TS 62998-1:2019: Safety of machinery - Safety-related sensors used for the protection of persons.

[17] IEEE P7001 - IEEE Draft Standard for Transparency of Autonomous Systems.

[18] IEEE P7002 - IEEE Draft Standard for Data Privacy Process.

[19] IEEE P7003 - Algorithmic Bias Considerations. An active project.

13.2 国・国際機関の指針等

[20] 内閣府 統合イノベーション戦略推進会議. 人間中心の AI 社会原則. 2019 年 3 月. https://www8.cao.go.jp/cstp/aigensoku.pdf

[21] Organisation for Economic Co-operation and Development (OECD). *OECD Principles on Artificial Intelligence*. May 2019. https://www.oecd.org/going-digital/ai/principles/

[22] European Commission. *Regulation of the European parliament and of the council laying down harmonized rules on artificial intelligence (Artificial Intelligence Act) and amending certain union legislative acts*. April 2021. https://digital-strategy.ec.europa.eu/en/library/proposal-regulation-laying-down-harmonised-rules-artificial-intelligence-artificial-intelligence

[23] European Commission. *Communication Artificial Intelligence for Europe*. 2018. https://digital-strategy.ec.europa.eu/en/library/communication-artificial-intelligence-europe.

[24] European Commission. *The White Paper on Artificial Intelligence – A European approach to excellence and trust*. February 2020. https://ec.europa.eu/info/publications/white-paper-artificial-intelligence-european-approach-excellence-and-trust_en

[25] European Commission. High-Level Expert Group on Artificial Intelligence. 2018. https://digital-strategy.ec.europa.eu/en/policies/expert-group-ai

[26] The High-Level Expert Group on Artificial Intelligence, European Commission. *Ethics guidelines for trustworthy AI*. April 2019. https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai

[27] The High-Level Expert Group on Artificial Intelligence, European Commission. *The Assessment List for Trustworthy Artificial Intelligence (ALTAI)*. July 2020. https://ec.europa.eu/digital-single-market/en/news/assessment-list-trustworthy-

artificial-intelligence-altai-self-assessment

[28] 経済産業省. AI・データの利用に関する契約ガイドライン. 2018 年 6 月. https://www.meti.go.jp/press/2018/06/20180615001/20180615001-3.pdf

[29] 経済産業省 AI 社会実装アーキテクチャ検討会. 我が国の AI ガバナンスの在り方 ver1.0. 2021 年 1 月. https://www.meti.go.jp/press/2020/01/20210115003/20210115003-1.pdf

[30] The IEEE Global Initiative on Ethics of Autonomous and Intelligent Systems. *Ethically Aligned Design: A Vision for Prioritizing Human Well-being with Autonomous and Intelligent Systems, First Edition*. IEEE, 2019. https://standards.ieee.org/content/ieee-standards/en/industry-connections/ec/autonomous-systems.html

13.3 公的規格・フォーラム標準等

[31] National Institute of Standards and Technology (United States of America). Draft NIST IR 8269: *A Taxonomy and Terminology of Adversarial Machine Learning*. October 2019. https://nvlpubs.nist.gov/nistpubs/ir/2019/NIST.IR.8269-draft.pdf

[32] National Institute of Standards and Technology (United States of America). NIST SP 800-30: *Guide for Conducting Risk Assessments*. September 2012. https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-30r1.pdf

[33] National Institute of Standards and Technology (United States of America). *Cyber Security Framework Version 1.1*. April 2018. https://doi.org/10.6028/NIST.CSWP.04162018

[34] MITRE. *Adversarial ML Threat Matrix*. https://github.com/mitre/advmlthreatmatrix

[35] MITRE, *Common Platform Enumerations*. http://cpe.mitre.org/

[36] MITRE, *Common Vulnerability Enumerations*. https://cve.mitre.org/

[37] Payment Card Industry Security Standards Council. Payment Card Industry Data Security Standard (PCI DSS). https://www.pcisecuritystandards.org/.

[38] AI プロダクト品質保証コンソーシアム. AI プロダクト品質保証ガイドライン 2020.08 版. 2020 年 8 月. http://qa4ai.jp/download/.

13.4 学術論文等

[39] Martín Abadi, Andy Chu, Ian J. Goodfellow, H. Brendan McMahan, Ilya Mironov,

Kunal Talwar, and Li Zhang. Deep Learning with Differential Privacy. In *Proc. of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS'16)*, pp. 308–318, 2016.

[40] Naveed Akhtar, and Ajmal Mian. Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey. *IEEE Access* 6, pp. 14410–14430, 2018. https://arxiv.org/pdf/1801.00553.pdf

[41] P. Ammann, and J. Offutt. *Introduction to Software Testing*, Cambridge University Press, 2008.

[42] Giuseppe Ateniese, Luigi V. Mancini, Angelo Spognardi, Antonio Villani, Domenico Vitali, and Giovanni Felici. Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. *International Journal of Security and Networks*, 10(3):137–150, 2015.

[43] Guy Barash, Eitan Farchi, Ilan Jayaraman, Orna Raz, Rachel Tzoref-Brill, and Marcel Zalmanovici. Bridging the gap between ML solutions and their business requirements using feature interactions. In *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2019)*, pp. 1048–1058, August 2019. https://dl.acm.org/citation.cfm?id=3340442

[44] E. T. Barr, M. Harman, P. McMinn, M. Shahbaz, and S. Yoo. The Oracle Problem in Software Testing: A Survey. In *IEEE Transactions on Software Engineering*, 41(5):507–525, 2015.

[45] Akhilan Boopathy, Tsui-Wei Weng, Pin-Yu Chen, Sijia Liu, and Luca Daniel. CNN-Cert: An Efficient Framework for Certifying Robustness of Convolutional Neural Networks. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI 2019)*, pp.3240–3247, 2019.

[46] Flavio Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R. Varshney. Optimized Pre-Processing for Discrimination Prevention. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, December 2017. https://papers.nips.cc/paper/6988-optimized-pre-processing-for-discrimination-prevention.pdf.

[47] Nicholas Carlini, and David Wagner. Towards Evaluating the Robustness of Neural Networks. In *Proceedings of the 38th IEEE Symposium on Security and Privacy (SP)*, pp. 39–57, 2017.

[48] Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian Molloy and Biplav Srivastava. Detecting Backdoor Attacks on Deep Neural Networks by Activation Clustering. In *Proceedings of the Workshop on Artificial Intelligence Safety 2019*, 2019. http://ceur-ws.org/Vol-2301/paper_18.pdf

[49] T. Y. Chen, S. C. Chung, and S. M. You. Metamorphic Testing – A New Approach for Generating Next Test Cases, Technical Report HKUST-CS98-01, Department of

Computer Science, The Hong Kong University of Science and Technology, 1998.

[50] T. Y. Chen, F.-C. Kuo, H. Liu, P.-L. Poon, D. Towey, Y. H. Tse, and Z. Q. Zhou. Metamorphic Testing: A Review of Challenges and Opportunities. *ACM Computing Surveys*, 51(1):1–27, 2018.

[51] S. Chiappa, and W. S. Isaac. A Causal Bayesian Networks: Viewpoint on Fairness. In *Privacy and Identity Management: Fairness, Accountability, and Transparency in the Age of Big Data*, IFIP Advances in Information and Communication Technology, vol. 547, 2019.

[52] Jeremy M Cohen, Elan Rosenfeld, and J. Zico Kolter. Certified Adversarial Robustness via Randomized Smoothing. *The 36th International Conference on Machine Learning (ICML 2019)*, 2019.

[53] George Deckert, NASA Hazard Analysis Process. Johnson Space Center, National Aeronautics and Space Administration, 2010. https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20100040678.pdf.

[54] Ann-Kathrin Dombrowski, Maximilian Alber, Christopher J. Anders, Marcel Ackermann, Klaus-Robert Müller, and Pan Kessel. Explanations can be manipulated and geometry is to blame. In *Proceedings of the Annual Conference on Neural Information Processing Systems 2019 (NeurIPS'19)*, pp. 13567–13578, 2019. https://arxiv.org/pdf/1906.07983.pdf

[55] Yizhen Dong, Peixin Zhang, Jingyi Wang, Shuang Liu, Jun Sun, Jianye Hao, Xinyu Wang, Li Wang, Jin Song Dong, and Dai Ting. There is Limited Correlation between Coverage and Robustness for Deep Neural Networks. arXiv:1911.05904 [cs.LG]. https://arxiv.org/abs/1911.05904

[56] S. Elbaum and D. S. Rosenblum. Known Unknowns – Testing in the Presence of Uncertainty, In *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE 2014)*, pp. 833–836, 2014.

[57] 江間 有沙 (ed.). 小特集「AI 原則から実践へ：国際的な活動紹介」. 人工知能 36(2), 人工知能学会, 2021 年 3 月.

[58] E. R. Faria, J. Gama, and A. C. Carvalho. Novelty detection algorithm for data streams multi class problems, In *Procedings of the 28th annual ACM symposium on applied computing*, pp. 795–800, 2013.

[59] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS '15)*, pp. 1322–1333, 2015. https://dl.acm.org/doi/10.1145/2810103.2813677

[60] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, A survey on concept drift adaptation. In *ACM computing surveys*, 46(4):1–37, April 2014.

[61] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems 27 (NIPS 2014)*, pp. 2672–2680, 2014.

[62] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. In *Proc. of the 3rd International Conference on Learning Representations (ICLR '15)*, 2015. https://arxiv.org/pdf/1412.6572.pdf

[63] F. Harel-Canada, L. Wang, M. A. Gulzar, Q. Gu, and M. Kim. Is Neuron Coverage a Meaningful Measure for Testing Deep Neural Networks? In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2020)*, pp. 851–862, 2020.

[64] Christina Ilvento. Metric Learning for Individual Fairness. arXiv:1906.00250 [cs.LG]. https://arxiv.org/abs/1906.00250

[65] L. Inozemtseva, and R. Holmes. Coverage is not Strongly Correlated with Test Suite Effectiveness, In *Proceedings of the 36th International Conference on Software Engineering (ICSE 2014)*, pp. 435–455, 2014.

[66] Matthew Jagielski, Alina Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru and Bo Li. Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning. In *Proceedings of the 2018 IEEE Symposium on Security and Privacy (S&P '18)*, pp. 19–35, 2018.
https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8418594.

[67] J. Jia, A. Salem, M. Backes, Y. Zhang, and N. Z. Gong. MemGuard: Defending against Black-Box Membership Inference Attacks via Adver- sarial Examples. In *Proc. of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS'19)*, pp. 259–274, 2019.

[68] Mika Juuti, Sebastian Szyller, Samuel Marchal, and N. Asokan. PRADA: Protecting Against DNN Model Stealing Attacks. In *Proc. of the IEEE European Symposium on Security and Privacy (Euro S&P '19)*, pp.512–527, 2019.
https://arxiv.org/pdf/1805.02628.pdf

[69] Guy Katz, Clark Barrett, David Dill, Kyle Julian, and Mykel Kochenderfer. Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks. In *International Conference on Computer-Aided Verification (CAV)*, 2017.

[70] T. Kamishima, S. Akaho, H. Asoh, and J. Sakuma. Fairness-aware classifier with prejudice remover regularizer. In *Machine Learning and Knowledge Discovery in Databases*, Lecture Notes in Computer Science, 7524:35–50, 2012.
https://link.springer.com/content/pdf/10.1007%2F978-3-642-33486-3_3.pdf

[71] J. Kim, R. Feldt, and S. Yoo. Guiding Deep Learning System Testing Using Surprise Adequacy, In *Proceedings of the 41st International Conference on Software Engineering (ICSE '19)*, pp. 1039–1049, 2019.

[72] Rob Kohavi. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. *The 14th International Joint Conference on Artificial Intelligence*, 2(12):1137–1143, 1995.

[73] F. Kamiran, A. Karim, and X. Zhang. Decision theory for discrimination-aware classification. In *Proceedings of the IEEE International Conference on Data Mining (ICDM 2012)*, 2012.

[74] Faisal Kamiran, and Toon Calders. Data preprocessing techniques for classification without discrimination. In *Knowledge and Information Systems*, 33(1):1–33, 2012.

[75] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet Classification with Deep Convolutional Neural Networks. In *Procedings of the Advances in Neural Information Processing Systems 25 (NIPS 2012*), pp.1097–1105, 2012.

[76] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial Examples in the Physical World. In *Proc. of the 5th International Conference on Learning Representations (ICLR) Workshop*, arXiv:1607.02533 [cs.CV], July 2016, https://arxiv.org/pdf/1607.02533.pdf

[77] Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified Robustness to Adversarial Examples with Differential Privacy. *The IEEE Symposium on Security and Privacy* (SP), 2019.

[78] S. Lee, J. Kim, J. Jun, J. Ha, and B. Zhang. Overcoming catastrophic forgetting by incremental moment matching. In *Proc. of the Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pp. 4652–4662, 2017.

[79] Yiming Li, Baoyuan Wu, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor Learning: A Survey. arXiv:2007.08745 [cs.CR], July 2020, https://arxiv.org/pdf/2007.08745

[80] P. Lindstrom, B. M. Namee, and S. J. Delany. Drift detection using uncertainty distribution divergence. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining Workshops*, pp. 604–608, 2011.

[81] X. Liu, M. Cheng, H. Zhang, and C. Hsieh. Towards robust neural networks via random self-ensemble. In *Proceedings of the European Conference on Computer Vision (ECCV 2018)*, 2018.

[82] Lei Ma, Felix Juefei-Xu, Fuyuan Zhang, Jiyuan Sun, Minhui Xue, Bo Li, Chunyang Chen, Ting Su, Li Li, Yang Liu, Jianjun Zhao, and Yadong Wang. DeepGauge: Multi-Granularity Testing Criteria for Deep Learning Systems, In *Proceedings of the 33rd IEEE/ACM International Conference on Automated Software Engineering (ASE 2018)*, pp. 120–131, 2018.

[83] Shiqing Ma, Yingqi Liu, Guanhong Tao, Wen-Chuan Lee, and Xiangyu Zhang. NIC: Detecting Adversarial Samples with Neural Network Invariant Checking. In *Network and Distributed Systems Security Symposium (NDSS)*, 2019.

[84] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian

Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks. In *the Sixth International Conference on Learning Representations (ICLR 2018)*, 2018.

[85] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A Survey on Bias and Fairness in Machine Learning. arXiv:1908.09635 [cs.LG], 2019.

[86] B.P. Miller, L. Fredricksen, and B. So. An Empirical Study of the Reliability of UNIX Utilities, *Communications of the ACM*, 33(12):32–44, 1990.

[87] 中島 震. データセット多様性のソフトウェア・テスティング, コンピュータ・ソフトウェア, 35(2):26–32, 2018.

[88] Shin Nakajima. Distortion and Faults in Machine Learning Software, In *Proc. 9th International Workshop on SOFL + MSVL for Reliability and Security*, pp. 29–41, 2020, and also arXiv:1911.11596 [cs.LG], 2019.

[89] 中島 震. ソフトウェア工学から学ぶ機械学習の品質問題, 丸善出版, 2020.

[90] 中島 震. 訓練済み機械学習モデル歪みの定量指標, 電子情報通信学会ソフトウェアサイエンス研究会, 2020.

[91] Shin Nakajima. Statistical Partial Oracle for Machine Learning Software Testing, In *Proceedings of the 10th International Workshop on SOFL + MSVL for Reliability and Security*, 2021.

[92] Shin Nakajima and Tsong Yueh Chen. Generating Biased Dataset for Metamorphic Testing of Machine Learning Programs, In *Proceedings of The 31st IFIP International Conference On Testing Software And Systems (IFIP–ICTSS 2019)*, pp. 56–64, 2019.

[93] 中谷 多哉子, 中島 震. ソフトウェア工学. 放送大学大学院教材, 放送大学教育振興会, 2019 年 3 月.

[94] Steven Nowlan, and Geoffrey Hinton. Simplifying neural networks by soft weight-sharing. *Neural Computation*, 4(4), 1992.

[95] L. Oneto, N. Navarin, A. Sperduti, and D. Anguita. Recent Trends in Learning From Data. *Tutorials from the INNS Big Data and Deep Learning Conference (INNSBDDL2019)*, January 2020.

[96] Tinghui Ouyang, Vicent Sanz Marco, Yoshinao Isobe, Hideki Asoh, Yutaka Oiwa, and Yoshiki Seo. Corner Case Data Description and Detection. arXiv:2101.02494v2 [cs.LG]. https://arxiv.org/pdf/2101.02494.pdf

[97] Judea Pearl. Understanding Simpson's Paradox. *The American Statistician*, 68(1):8–13, February 2014. Also UCLA Cognitive Systems Laboratory Technical Report R-414, University of California, Los Angeles, December 2013.

[98] Kexin Pei, Yinzhi Cao, Junfeng Yang, and Suman Jana. DeepXplore: Automated

Whitebox Testing of Deep Learning Systems, *The 26th Symposium on Operating Systems Principles (SOSP 2017)*, pp. 1–18, 2017.

[99] G. Pleiss, M. Raghavan, F. Wu, J. Kleinberg, and K. Q. Weinberger. On fairness and calibration. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS '17)*, December 2017.

[100] D. M. dos Reis, P. Flach, S. Matwin, and G. Batista. Fast unsupervised online drift detection using incremental Kolmogorov Smirnov test. In *Proc. of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1545–1554, 2016.

[101] T. S. Sethi, and M. Kantardzic. On the reliable detection of concept drift from streaming unlabeled data. *Expert Systems with Applications*, 82:77–99, 2017.

[102] B. Settles. Active Learning Literature Survey. Technical Report #1648, Computer Science Department, University of Wisconsin-Madison, 2009.

[103] S. Shalev-Shwartz. Online learning and online convex optimization, *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

[104] Reza Shokri, Marco Stronati, Congzheng Song and Vitaly Shmatikov. Membership Inference Attacks Against Machine Learning Models. In *Proceedings of the 2017 IEEE Symposium on Security and Privacy (S&P'17)*, pp. 3–18, 2017.
https://arxiv.org/pdf/1610.05820.pdf

[105] M. S. R. Shuvo and D. Alhadidi. Membership Inference Attacks: Analysis and Mitigation. In *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications* (TrustCom), pp. 1410–1419, 2020. doi: 10.1109/TrustCom50675.2020.00190.
https://ieeexplore.ieee.org/document/9343081

[106] Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. Fooling LIME and SHAP: Adversarial Attacks on Post hoc Explanation Methods. In *Proc. of the AAAI/ACM Conference on AI, Ethics, and Society 2020 (AIES'20)*, pp.180–186, 2020. https://arxiv.org/pdf/1911.02508.pdf

[107] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting, *Journal of Machine Learning Research* 15(56):1929–1958, 2014.

[108] H. Tian, M. Yu, and W. Wang. Continuum: A platform for cost aware, low latency continual learning, In *Proceedings of the ACM Symposium on Cloud Computing*. pp. 26–40, 2018.

[109] Y. Tian, K. Pei, S. Jana, and B. Ray. DeepTest: Automated Testing of Deep-Neural-Network-driven Autonomous Cars. In *Proceedings of the 40th International Conference on Software Engineering (ICSE 2018)*, pp. 303–314, 2018.

[110] Vincent Tjeng, Kai Xiao, and Russ Tedrake. Evaluating robustness of neural networks

with mixed integer programming. In *International Conference on Learning Representations*, 2019.

[111] Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. Stealing Machine Learning Models via Prediction APIs. In *Proceedings of the 25th USENIX Security Symposium (USENIX Security '16)*. https://arxiv.org/pdf/1609.02943.pdf

[112] Guillermo Valle-Pérez and Ard A. Louis. Generalization bounds for deep learning. arXiv:2012.04115v2, 2020. https://arxiv.org/abs/2012.04115

[113] A. Vostrikov and S. Chernyshev. Training sample generation software. In *Intelligent Decision Technologies 2019*, Smart Innovation, Systems and Technologies (SIST), 143:145–151, 2019.

[114] J. Wang, J. Chen, Y. Sun, X. Ma, D. Wang, J. Sun, and P. Cheng. RoBOT: Robustness-Oriented Testing for Deep Learning Systems, In *Proceedings of the 43rd International Conference on Software Engineering (ICSE 2021)*, 2021.

[115] Tsui-Wei Weng, Pin-Yu Chen, Lam Nguyen, Mark Squillante, Akhilan Boopathy, Ivan Oseledets, and Luca Daniel. PROVEN: Verifying Robustness of Neural Networks with a Probabilistic Approach. In *Proceedings of the 36th International Conference on Machine Learning (ICML 2019)*, PMLR 97:6727–6736, 2019.

[116] Tsui-Wei Weng, Huan Zhang, Hongge Chen, Zhao Song, Cho-Jui Hsieh, Duane Boning, Inderjit S. Dhillon, and Luca Daniel. Towards Fast Computation of Certified Robustness for ReLU Networks. In *Proceedings of the 35th International Conference on Machine Learning*, PMLR 80:5276-5285, 2018.

[117] E. J. Weyuker. On Testing Non-testable Programs, Computer Journal, 25(4):465–470, 1982.

[118] Eric Wong and J. Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. *The 35th International Conference on Machine Learning (ICML 2018)*, PMLR vol. 80, pp. 5283–5292, 2018.

[119] Min Wu, Matthew Wicker, Wenjie Ruan, Xiaowei Huang, and Marta Kwiatkowska. A Game-Based Approximate Verification of Deep Neural Networks with Provable Guarantees. In *Theoretical Computer Science,* 2019. https://doi.org/10.1016/j.tcs.2019.05.046

[120] Weilin Xu, David Evans, and Yanjun Qi. Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks. In *Network and Distributed Systems Security Symposium* (NDSS), 2018.

[121] B. H. Zhang, B. Lemoine, and M. Mitchell. Mitigating Unwanted Biases with Adversarial Learning. In *AIES '18: Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, December 2018. https://dl.acm.org/doi/pdf/10.1145/3278721.3278779.

[122] Jie M. Zhang, Mark Harman, Lei Ma, and Yang Liu. Machine Learning Testing: Survey, Landscapes and Horizons. arXiv:1906.10742 [cs.LG]. https://arxiv.org/abs/1906.10742

[123] P. Zhang, Q. Dai, and P. Pelliccione. CAGFuzz: Coverage-Guided Adversarial Generative Fuzzing Testing of Deep Learning Systems. arXiv:1911.07931, 2019.

[124] P. Zhang, J. Wang, J. Sun, G. Dong, and X. Wang. White-box Fairness Testing through Adversarial Sampling. In *Proceedings of the 42nd International Conference on Software Engineering (ICSE 2020)*, pp. 1–12, 2020.

13.5 その他

[125] 独立行政法人情報処理推進機構. はじめての *STAMP/STPA*. 2019 年 6 月. https://www.ipa.go.jp/ikc/reports/20190329.html

[126] 独立行政法人情報処理推進機構 セキュリティセンター. つながる世界の品質確保に向けた手引き. 2018 年 6 月. https://www.ipa.go.jp/sec/publish/tn18-001.html

[127] 独立行政法人情報処理推進機構 セキュリティセンター. 共通プラットフォーム一覧 CPE 概説. 2008 年 10 月. https://www.ipa.go.jp/security/vuln/CPE.html

[128] 独立行政法人情報処理推進機構 セキュリティセンター. 共通脆弱性識別子 CVE 概説. 2009 年 1 月. https://www.ipa.go.jp/security/vuln/CVE.html

[129] 国立研究開発法人産業技術総合研究所. 機械学習品質評価・向上技術に関する報告書, デジタルアーキテクチャ研究センター・サイバーフィジカルセキュリティ研究センター・人工知能研究センター テクニカルレポート, DigiARC-TR-2021-02 / CPSEC-TR-2021002, 2021.

[130] IBM Corporation. AI Fairness 360 - Resources. http://aif360.mybluemix.net/resources

14. 主な変更点

14.1 第 2 版（2021 年 7 月）

- 1.5.3 節の公平性の定義を、より詳細に検討し変更した。
- 1.7 節及び 6 章の内部品質特性を A～D にカテゴリ分けし、A-1～E-1 まで付番した。
- 内部品質特性 B-3「データの妥当性」を、特性 C-1「機械学習モデルの正確性」から分離し、内容を拡充した。
- 内部品質特性の一部を改名した。
- 公平性に関する 8 章及びセキュリティに関する 9 章を追加した。

産総研 人工知能研究センター・サイバーフィジカルセキュリティ研究センター
機械学習品質マネジメント検討委員会

メンバリスト（2020 年度）

中島 震	産総研・国立情報学研究所（委員長）
妹尾　義樹	産総研（副委員長）
江川　尚志	日本電気
大岩　寛	産総研
小川　秀人	日立製作所
岡本　球夫	パナソニック
桑島　洋	デンソー
小林　健一	富士通
佐藤　直人	日立製作所
鈴木　知道	東京理科大学
高田　眞吾	慶應義塾大学
土屋　哲	富士通
中島　裕生	テクマトリックス
難波　孝彰	パナソニック
浜谷　千波	アドソル日進
福島　真太朗	トヨタ自動車
山田　敦	日本 IBM
若松　直哉	日本電気

（敬称略・五十音順）

機械学習品質マネジメントガイドライン　執筆者リスト

・ 全体構成・編集	大岩 寛（産業技術総合研究所）
・ 5.2 節	浜谷 千波（アドソル日進）
・ 7.6 節	中島 震（国立情報学研究所）
・ 7.6.2 節	磯部 祥尚（産業技術総合研究所）
・ 7.8 節	小林　健一，大川　佳寛（富士通）
・ 8 章	大岩 寛，浜谷 千波
・ 9.2〜9.3 節	大岩 寛，川本 裕輔（産業技術総合研究所）， 三宅 和公（住友電気工業）
・ 9.4 節	三宅 和公（住友電気工業）
・ 10.2 節	江川 尚志（日本電気），桑島 洋（デンソー）の 寄稿による
・ 内容検討・監修	AI 品質マネジメント検討委員会 各委員 AI 品質マネジメント検討委員会 ガイドライン詳細検討タスクフォース メンバー

ガイドライン詳細検討タスクフォース　メンバーリスト

磯部　祥尚	産総研
江川　尚志	日本電気
大岩　寛	産総研
岡本　球夫	パナソニック
桑島　洋	デンソー
川本　裕輔	産総研
小林　健一	富士通
妹尾　義樹	産総研
土屋　哲	富士通
中島　裕生	テクマトリックス
中島　震	産総研
難波　孝彰	パナソニック
浜谷　千波	アドソル日進
福島　真太朗	トヨタ自動車
三宅　和公	住友電気工業
三宅　武史	サイバー創研

（敬称略・五十音順）

本プロジェクトに関係して産総研に特定集中研究専門員として部分的に在籍しているメンバーについては、それぞれの出身母体で記載した。