確率統計に基づいた故障木とテストによる機械学習システムの系統的評価手法

青木 利晃[1,a)]　川上 大介[2,b)]　千田 伸男[2,c)]　冨田尭[1,d)]

概要：近年，深層学習を代表とする機械学習が脚光を浴びており，様々なシステムへの応用が検討されている．応用対象は，エンターテイメントシステムや事務処理システムなどにとどまらず，自動車の自動運転などの高安全システムなども検討されている．高安全システムでは，安全性に関する十分なテストを実施することが必要である．一方，機械学習システムの安全性をテストする手法は，まだ確立していない．そこで，本論文では，機械学習を用いて実現された分類器に焦点を当て，その安全性を系統的に評価する手法を提案する．提案手法は，データセットに基づいた安全分析とテスト結果の統計的評価方法から構成される．安全分析では，よく知られている FTA(Fault Tree Analysis) をデータセットを取り扱えるように拡張した．そして，安全分析の結果，期待される認識率の水準を求め，テストを実施する．テストでは，この水準を満たしているか統計的に評価を実施する．提案手法は，CNN(Convolutional Neural Network) による手書き文字の分類器に適用し，評価を行った．

Systematic Evaluation of Machine Learning Systems by Fault Tree and Testing based on Stochastic and Statistic Method

Toshiaki Aoki[1,a)]　Daisuke Kawakami[2,b)]　Nobuo Chida[2,c)]　Takashi Tomita[1,d)]

Abstract: Recently, machine learning, especially, deep learning, accumulates interests and applying it to various kinds of systems is being studied. The targets of applications include not only entertainment systems but also safety critical systems such as autonomous driving of car. In a safety critical system, it is indispensable to ensure the safety of the system before using and deploying it in our society. However, how we ensure the safety of systems which use machine learning has not been established yet. In this paper, we propose a method to systematically evaluate the safety of such systems. The proposed method consists of the safety analysis based on datasets and the statistical evaluation of testing results. In the safety analysis, widely known and used FTA (Fault Tree Analysis) is extended to deal with the datasets and it allows us to obtain expected recognition rates of the systems. Then, we conduct the testing of systems in order to check whether it is met or not. The satisfaction of the expected recognition rates is ensured by statistic evaluation of the testing result. We also conducted experiments using a handwritten character recognition system which is realized by CNN (Convolutional Neural Network) to show the feasibility and effectiveness of our method. We focus on supervised classifiers realized by machine learning in this paper but we believe that it can be extended to the other kinds of machine learning applications.

Keywords: Machine Learning, Fault Tree, Testing, Stochastic/Statistic Method

[1] 北陸先端科学技術大学院大学
JAIST, 1-1 Asahidai, Nomi, Ishikawa, JAPAN
[2] 三菱電機株式会社
Mitsubishi Electric Corporation
[a)] toshiaki@jaist.ac.jp
[b)] Kawakami.Daisuke@ab.MitsubishiElectric.co.jp
[c)] Chida.Nobuo@ab.MitsubishiElectric.co.jp
[d)] tomita@jaist.ac.jp

1. 背景

近年，深層学習を代表とする機械学習が脚光を浴びており，様々なシステムへの応用が検討されている．応用対象は，エンターテイメントシステムや事務処理システムなど

にとどまらず，自動車の自動運転などの高安全システムなども検討されている．高安全システムでは，安全性に関する十分な検証を実施することが必要である．一方，機械学習システムの安全性を検証する手法は，まだ確立していない．そこで，本論文では，機械学習システムの安全性を系統的に評価する手法を提案する．機械学習では，まず，学習用のデータセットを用いて学習を行い，重みなどの値を決定してモデルを作成する．実際のシステムでは，この学習済のモデルを用いる場合が多い．実際にデータを入力してモデルに基づいて計算し，結果を獲得するのである．提案手法は，この学習済のモデルを用いたシステムを系統的に評価するものである．

本論文では，機械学習で実現された分類器のテストに焦点を当てる．分類器のテストでは，オラクル問題が存在する．オラクル問題とは，期待値がわからないか，それを求めるのが困難である，という問題である．分類器のテストでは，期待する分類結果が期待値である．しかしながら，そのような期待値は，すべての入力に対しては，あらかじめ，決めることはできない．一方，機械学習システムの入力の一部を作成して，テストすることは可能である．例えば，犬と猫の画像を分類をする分類器では，人間が見て犬と判断する画像と猫と判断する画像のデータセットを作成して，それらを入力として分類器により分類し，人間の判断と合致するかテストするのである．実際，多くの機械学習に関する論文では，手持ちのデータセットを学習用と評価用の 2 つに分解して用いている．

分類器の評価では，答えが与えられている，すなわち，ラベル付きデータセットを評価用データセットとして用いるが，十分なサイズのデータを収集するのは困難である．そのため，評価用データセットは，実際に分類器に入力されるデータ数と比べて，圧倒的に小さくならざるをえなく，十分に事前に評価を行うことができないことが多い．また，期待する認識率など，特定の基準を満たしているか評価を行うが，この基準は，評価用データセットの内容によって，異なると考えられる．例えば，訓練データセットと同等のものの場合，高い認識率が期待されるが，ノイズが多く入った悪状況を想定したデータセットでは，あまり高い認識率は期待できない．よって，実際に分類器を使う状況を想定して，評価用データセットを作成し，系統的に評価する必要がある．最近では，Python[2] などのプログラミング言語や OpenCV[1] などの画像処理ツールを使用することにより，簡便に，データセットの処理を行えるようになってきた．また，adversarial examples などの誤認識を誘発する画像を自動生成する手法も提案されている [8], [9], [10]．そこで，本論文では，シミュレーションや乱数で状況に応じた評価用のデータセットを自動生成することを想定する．そして，自動生成されたデータセットを用いて，テストにより機械学習システムを系統的に評価する手法を提案する．提案手法は，個々の評価用データセットによるテスト結果を統計的に評価する手法と，評価用データセットとテスト結果全体を系統的に分析・評価する手法により構成される．

本論文は，以下のように構成される．2 節では，関連研究について述べる．3 節では，全体のアプローチについて説明する．4 節では，データセットに基づいた安全分析手法，5 節では，テスト結果を統計的に評価する手法について提案する．6 節では，実施した実験について示し，7 節において，それに基づいた議論を行う．8 節において，本論文における貢献をまとめ，今後の課題について述べる．

2. 関連研究

最近になって，機械学習を用いたソフトウェアのテストに関する研究が見受けられるようになってきた．文献 [5], [6] では，機械学習による分類器の誤りを，Metamorphic Testing の手法を用いて検出する手法を提案している．文献 [7] では，Ranking Algorithm(MartiRank) と SVM(SVM-Light) を対象にテストを実施する知見がまとめられている．また，ニューラルネットワークにおいては，adversarial example と呼ばれる，代表的な脆弱性があることが指摘されている [9], [10]．このようなシステムの脆弱性の検出を目的とするテスト手法も提案されつつある [8]．これらの研究は，システムの誤りや潜在的な脆弱性を発見するものであり，妥当性を評価するという我々の目的とは異なる．

文献 [12], [13] では，既存画像を目的にあわせて修正することにより，様々なテスト用の画像データを生成し，機械学習システムの評価している．文献 [14] では，システムの性質を記述するために signal temporal logic という論理を用いて，誤りを発見する手法を提案している．文献 [11] では，ニューロンカバレッジと呼ばれる指標を定義し，それが最大になるよう画像データを生成している．これらの研究は画像データの生成に焦点を当てている．一方，我々の手法は，そのような画像データ生成手法を前提として，確率と統計を用いて系統的に評価する手法を提案している．

提案手法では，安全分析の代表的な手法の 1 つである FTA(Fault Tree Analysis)[3], [4] を用いて，テスト結果を系統的に評価している．FTA で使われる故障木 (Fault Tree) には，様々な解釈が考えられるため，様々な形式化や分析手法が提案されている．文献 [15] では，interval temporal logic により，故障木を形式化している．この形式化は，時間付きの状態モデルを検証することを意図したものである．文献 [16] では，故障木において確率を取り扱う場合について厳密な解釈を与えている．また，確率モデル検査により，故障木上の確率的な特性について分析する手法を提案している．本論文では，これらの研究とは異なり，故障木においてデータセットを取り扱う場合の解釈を与えている．

3. アプローチ

3.1 期待値としての認識率

通常のテストでは，テストケースとして，入力と期待値を作成する．そして，SUT(System Under Test) に入力し，その結果と期待値を比較する．結果が期待値と同じ場合は，テストはパス (pass) し，異なれば，テストは失敗 (fail) する．一方，分類器のテストでは，各々の入力に対する，出力と期待値の比較はあまり意味を持たない．どの程度正しく識別できたか，すなわち，認識率が重要な指標となる．そこで，各々のテストケースではなく，テストケースの集合，すなわち，テストスイート単位でテストを実施する．テストスイートは，データセット，期待値，期待する認識率（期待認識率と呼ぶ）により構成する．期待値は，分類器が識別すべきラベルを表現している．テストでは，データセットに含まれる要素を分類器で識別し，その結果が期待値と同じであれば正解，そうでなければ不正解とする．テストにおいて認識率は，正解数/データセットの要素数，で計算する．そして，その認識率が期待認識率を満たす時，テストはパスし，満たさない時に失敗，とする．

3.2 データセットの母集団と標本

提案手法では，状況に応じてテストスイートを準備する．例えば，画像認識で，紙が汚れている場合は，画像にノイズが入る状況，画像が欠損する状況などが考えられ，それぞれの状況に対してテストスイートを準備する．ここで，本手法では，テストスイートのデータセットを，母集団として取り扱い，その標本を用いてテストを実施するものとする．例えば，上記の状況では，それぞれのテストスイートにおいて，データセットを獲得するために，ドット追加，ドット欠けなどの画像を生成する必要がある．ドット追加，ドット欠けなどの画像のバリエーションは，有限ではあるが，とても多く，すべてをテストすることはできない．そこで，それらの画像を母集団とし，自動生成する画像は，その母集団からの標本と見なす．母集団には，真の認識率が存在し，それを知りたい．そこで，標本を用いたテストを実施して，この母集団の真の認識率を統計的に推定，および，検定するのである．

3.3 データセット用故障木

以上のテストでは，特定の状況に注目して，データセットを自動的に生成し，テストを実施する．しかしながら，状況を場当たり的に列挙すると，思いつきに依存するため，避けられるはずの思いつきの抜けが生じる可能性がある．さらに，場当たり的な列挙に基づいたテスト結果の妥当性は弱く，また，その一連の結果が示唆していることも不明確になる．そこで，系統的に状況を分析，列挙して，一連のテスト結果をまとめて解釈する手法を提案する．図 1 にその概要を示す．この手法では，代表的な安全分析手法 FTA(Fault Tree Analysis) で用いられる故障木 (Fault Tree) を，データセット用に拡張して用いる．この故障木を Fault Tree for Datasets と呼び, 略して，FT4D と表現することにする．FT4D では，FTA と同様，根ノード（トップ事象）を中間ノード（抽象的な事象）に分解し，リーフノード（基本的な事象）まで分解する．個々のノードにはデータセットおよび分類器において期待する認識率 (誤認識率) を割り当てる．ここで，上位のノードのデータセットおよび認識率は，下位のノードのデータセットと認識率により計算できるようにする．そして，リーフノードで上記のテストを実施して期待する認識率を保証することにより，分析したノードすべてにおいても，それらを保証することができるのである．

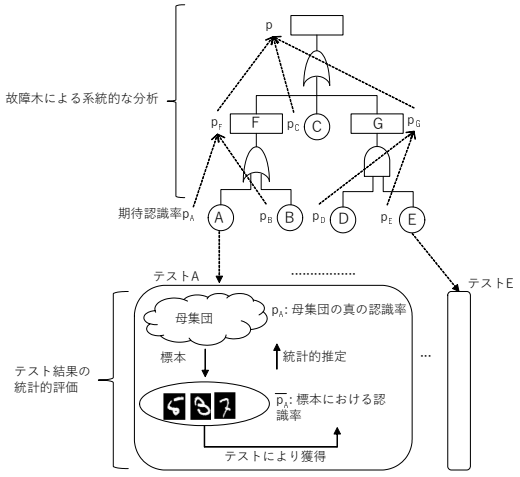

図 1 アプローチ概要

4. データセットに基づいた系統的分析手法

4.1 Fault Tree for Datasets

4.1.1 障害データセットと全体データセット

図 2 は，故障木を用いて，障害により誤認識の要因を含むデータセットを分析した例の一部である．紙が汚れて誤認識する状況を考える．このとき，文字にドットが追加される場合と，欠損する場合が考えられる．よって，E_1，E_2，E_3 を，それぞれ，紙が汚れる，ドット追加，ドット欠けを表現するイベントすると，E_1 は，E_2 と E_3 に OR ゲートで分解される．Fault Tree for Datatets(以下では，FT4D と呼ぶ) は，イベントの意味を，その障害により影響与えられたデータセットとして定義する．例えば，イベント E_1 は，紙が汚れた場合の文字のデータセットを表現しているものとする．イベント E_2, E_3 は，同様に，紙が汚れてドット追加した画像，および，ドット欠けした画像のデータセットにより表現する．イベント E_1, E_2, E_3 のデータセットを，それぞれ，D_{soil}，D_{add}，D_{loss} で表現し，

障害データセットと呼ぶことにする．これらのデータセットの間には，$D_{soil} = D_{add} \cup D_{loss}$ の関係がある．

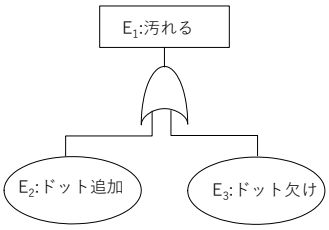

図 2 故障木 (Fault Tree) の例

障害により影響を与えられたデータは，対象システムが入力とするデータの一部である．多くの場合の入力は，障害が発生していない正常データであろう．そこで，対象となるシステムの入力全体を表現するデータセット U を導入する．U は，正常データ，および，障害により影響が与えられたデータを含んでいる．ここで，故障率という概念を導入する．故障率は，イベントが表現する障害が発生する割合であり，従来の FTA と同様の意味である．故障率は，データセット全体 U に対する，障害データセット D の割合 $|D|/|U|$ で定義する．例えば， イベント E_1, E_2, E_3 の故障率は，それぞれ，$|D_{soil}|/|U|$，$|D_{add}/|U|$，$|D_{loss}|/|U|$ であり，以下では，pf_1, pf_2, pf_3 として参照する．

4.1.2 誤認識率

障害データセットのデータの内，誤認識するデータの集まりを誤認識データセットと呼ぶことにする．イベントに対する故障データセット D と誤認識データセット E の間には，$E \subseteq D$ が成立する．イベント E_1, E_2, E_3 の誤認識データセットを，それぞれ，E_{soil}，E_{add}，E_{loss} で表現する．ここで，誤認識率を，全データセット U に対する，誤認識データセット E の割合 $|E|/|U|$ で定義する．故障木は，正常な動作における障害の割合の分析を行うため，このような誤認識率が適切であると考えられる．例えば， イベント E_1, E_2, E_3 の誤認識率は，それぞれ，$|E_{soil}|/|U|$，$|E_{add}|/|U|$，$|E_{loss}|/|U|$ である．これにより，上位の誤認識率を下位の誤認識率から，以下のように，計算できる．

- $E_{add} \cap E_{loss} = \phi$ の場合
  $\frac{|E_{soil}|}{|U|} = \frac{|E_{add} \cup E_{loss}|}{|U|} = \frac{|E_{add}|}{|U|} + \frac{|E_{loss}|}{|U|}$
- $E_{add} \cap E_{loss} \neq \phi$ の場合
  $\frac{|E_{soil}|}{|U|} = \frac{|E_{add} \cup E_{loss}|}{|U|} \leq \frac{|E_{add}|}{|U|} + \frac{|E_{loss}|}{|U|}$

4.1.3 基本誤認識率

テストは基本イベントに基づいて実施する．このテストの詳細は 5 節で述べるが，ここでは，誤認識率ではなく，故障データセット D に対する誤認識データセット E の割合 $|E|/|D|$ を確認することになる．この割合のことを，基本誤認識率と呼ぶことにする．基本誤認識率から誤認識率を計算することも可能である．図 2 において，E_2 が基本イベント，すなわち，最下位のイベントとする．ここで，テストを実施して，それぞれ，基本誤認識率 $pb_2 = |E_{add}|/|D_{add}|$ が確認されたとする．この基本誤認識率から誤認識率は以下のように計算できる．

$$\frac{|E_{add}|}{|U|} = \frac{|D_{add}|}{|U|} \cdot \frac{|E_{add}|}{|D_{add}|} = pf_2 \cdot pb_2$$

また，下位の誤認識率から上位の誤認識率が計算できるので，基本イベントの基本誤認識率が保証されれば，すべての誤認識率を保証することできる．

4.2 Formal Definitions

FT4D の文法を以下で定義する．

定義 1 FT4D の文法

FT4D は，以下の BNF で定義される $NODE$ である．

$$\begin{aligned} NODE &::= BE | FE(GATE) \\ GATE &::= OR(NODE, \cdots, NODE) \mid \\ &\quad\ AND(NODE, \cdots, NODE) \end{aligned}$$

ここで，BE は基本事象の名前，FE は故障事象の名前である．

全体データセット，故障データセット，誤認識データセットは以下のように定義される．

定義 2 全体データセット，故障データセット，誤認識データセット

全体データセット (universal data set) を U で表現する．ノードに対する故障データセット (fault data sets) と誤認識データセット (error data sets) は，それぞれ，以下の関数 D と E で表現する．

$$D : NODE \rightarrow 2^U$$
$$E : NODE \rightarrow 2^U$$

また，これらのデータセットは，以下の制約を満たす．

$$\forall n \in NODE.E(n) \subseteq D(n)$$

FT4D では，AND ゲートと OR ゲートの 2 種類が存在する．それぞれのゲートは，故障データセットと誤認識データセットが，どのように合成されるかで意味付けされる．

定義 3 ゲートの意味

それぞれのゲートを含むノード $n \in NODE$ では，故障データセットに関して，以下の関係が成立するものとする．

$$\begin{aligned} D(n) &= \textstyle\bigcup_{1 \leq i \leq m} D(n_i) \quad if \quad n = FE(OR(n_1, \cdots, n_m)) \\ D(n) &= \textstyle\bigcap_{1 \leq i \leq m} D(n_i) \quad if \quad n = FE(AND(n_1, \cdots, n_m)) \end{aligned}$$

それぞれのゲートを含むノード $n \in NODE$ では，誤認識データセットに関して，以下の関係が成立するものとする．

$$\begin{aligned} E(n) &= \textstyle\bigcup_{1 \leq i \leq m} E(n_i) \quad if \quad n = FE(OR(n_1, \cdots, n_m) \\ E(n) &= \textstyle\bigcap_{1 \leq i \leq m} E(n_i) \quad if \quad n = FE(AND(n_1, \cdots, n_m) \end{aligned}$$

FT4D のノードには，故障率，誤認識率，基本誤認識率

を割り当てる．

定義 4　故障率，誤認識率，基本誤認識率

故障率 σ_{pf}, 誤認識率 σ_{pe}，基本誤認識率 σ_{pb} を，以下により定義する．

$\sigma_{pf}(n) = \frac{\lfloor D(n) \rfloor}{\lfloor U \rfloor}$, $\sigma_{pe}(n) = \frac{\lfloor E(n) \rfloor}{\lfloor U \rfloor}$, $\sigma_{pb}(n) = \frac{\lfloor E(n) \rfloor}{\lfloor D(n) \rfloor}$

定理 1　故障率の制約

FT4D における，それぞれのゲートを含むノード $n \in NODE$ に対して以下の制約が成立する．

- $n = FE(OR(n_1, \cdots, n_m)) \wedge \forall i, j (1 \leq i, j \leq m \wedge i \neq j).D(n_i) \cap D(n_j) = \phi$ の場合．
  $\sigma_{pf}(n) = \sum_{i=1}^{i=n} \sigma_{pf}(n_1, \cdots, n_m)$
- $n = FE(OR(n_1, \cdots, n_m)) \wedge \exists i, j (1 \leq i, j \leq m \wedge i \neq j).D(n_i) \cap D(n_j) \neq \phi$ の場合．
  $\sigma_{pf}(n) \leq \sum_{i=1}^{m} \sigma_{pf}(n_1, \cdots, n_m)$
- $n = FE(AND(n_1, \cdots, n_m))$ の場合．
  $\sigma_{pf}(n) \leq \min_{1 \leq i \leq m} \sigma_{pf}(n_i)$

OR ゲートについては，例を用いて，すでに説明したため，AND ゲートの場合のみ，以下で証明を与える．

$n = AND(OR(n_1, \cdots, n_m))$ の時，以下が成立する．

$\sigma_{pf}(n) = \frac{|D(n)|}{|U|} = \frac{\left|\bigcap_{1 \leq i \leq m} D(n_i)\right|}{|U|} \leq \frac{\min_{1 \leq i \leq m} |D(n_i)|}{|U|}$
$= \min_{1 \leq i \leq m} \sigma_{pf}(n_i)$

定理 2　誤認識率の制約

FT4D における，それぞれのゲートを含むノード $n \in NODE$ に対して以下の制約が成立する．

- $n = FE(OR(n_1, \cdots, n_m)) \wedge \forall i, j (1 \leq i, j \leq m \wedge i \neq j).D(n_i) \cap D(n_j) = \phi$ の場合．
  $\sigma_{pe}(n) = \sum_{i=1}^{m} \sigma_{pe}(n_1, \cdots, n_m)$
- $n = FE(OR(n_1, \cdots, n_m)) \wedge \exists i, j (1 \leq i, j \leq n \wedge i \neq j).D(n_i) \cap D(n_j) \neq \phi$ の場合．
  $\sigma_{pe}(n) \leq \sum_{i=1}^{m} \sigma_{pe}(n_1, \cdots, n_m)$
- $n = FE(AND(n_1, \cdots, n_m))$ の場合．$\sigma_{pe}(n) \leq \min_{1 \leq i \leq m} \sigma_{pe}(n_i)$

OR ゲートについては，例を用いて，すでに説明したため，AND ゲートの場合のみ，以下で証明を与える．

$n = AND(OR(n_1, \cdots, n_m))$ の時，以下が成立する．

$\sigma_{pe}(n) = \frac{|E(n)|}{|U|} = \frac{\left|\bigcap_{1 \leq i \leq m} E(n_i)\right|}{|U|} \leq \frac{\min_{1 \leq i \leq m} |E(n_i)|}{|U|}$
$= \min_{1 \leq i \leq m} \sigma_{pe}(n_i)$

定理 3　基本故障率，誤認識率，故障率の関係

基本イベント $n \in NODE(BE)$ に対して，$\sigma_{pe}(n) = \sigma_{pf}(n) \cdot \sigma_{pe}(n)$ が成立する．

例を用いて，すでに説明したため，証明は省略する．

図 3 に FT4D の例を示す．我々の文法で表現すると，$A(OR(B, C(AND(D, E))))$ になる．

$\sigma_{pf}(A) \sim \sigma_{pf}(E)$ を，それぞれ，イベント $A \sim E$ の故障率とする．故障率の決定は，従来の FTA と同様である．すなわち，これらの故障率は，故障木を用いた分析の際，決定するものとする．イベント B, D, E は基本イベントである．基本イベントに対しては，テストを実施し，基本誤認識率を獲得する．テストの詳細は 5 節で説明する．獲得したイベント B, D, E の基本誤認識率を，それぞれ，$\sigma_{pb}(B), \sigma_{pb}(D), \sigma_{pb}(E)$ とする．イベント B, D, E の誤認識率 $\sigma_{pe}(B), \sigma_{pe}(D), \sigma_{pe}(E)$ は定理 3 により，以下のように計算できる．

$\sigma_{pe}(B) = \sigma_{pf}(B) \cdot \sigma_{pb}(B)$
$\sigma_{pe}(D) = \sigma_{pf}(D) \cdot \sigma_{pb}(D)$
$\sigma_{pe}(E) = \sigma_{pf}(E) \cdot \sigma_{pb}(E)$

さらに，イベント A, C の誤認識率を，$\sigma_{pe}(A), \sigma_{pe}(C)$ は，以上で獲得したイベント B, D, E の誤認識率 $\sigma_{pe}(B), \sigma_{pe}(D), \sigma_{pe}(E)$ と定理 2 より，以下のように計算できる．

$\sigma_{pe}(C) = \min\{\sigma_{pe}(D), \sigma_{pe}(E)\}$
$\sigma_{pe}(A) = \sigma_{pe}(B) + \sigma_{pe}(C)$

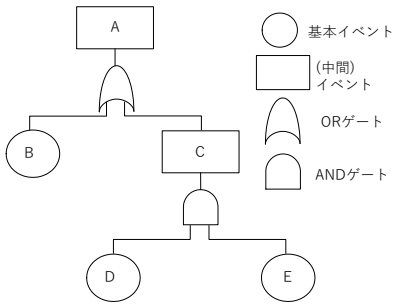

図 3　FT4D の例

よって，FT4D では，基本イベントに関して，テストを実施して誤認識率を求めると，上位のイベント，および，最上位のイベントの誤認識率を計算することができる．

5. テスト結果の統計的評価

5.1 評価用データセットと期待認識率

4 節で提案した，FT4D の基本イベントの個々の状況について，評価用データセットを用いてテストを実施する．テストスイートのデータセットは，非常に多くの要素を含む場合や，無限の場合が考えられる．例えば，100*100 ドットの 256 階調のモノクロ画像を対象として，「A」という画像に，3 ドットの任意の濃さのノイズを任意の場所に入れた画像を考える．この場合，データセットの要素数は，10^{10} 以上になる．よって，分類器のテストでは，すべての要素をテストして，認識率を求めることが困難な場合が多いと考えられる．そこで，テストスートを母集団として，母集団の認識率を標本から推定するアプローチをとる．D をデータセットの母集団，r を期待値，p_e を期待する認識率とすると，テストスイートを，3 つ組 (D, r, p_e) で表現する．そして，テストスイートの母集団分布を $D(r, p)$ で

表現する．ここで，r は既知の期待値，p は未知の真の認識率である．テストスイートのデータセットの標本集合を標本データセットと呼び，以下の $\overline{D}$ で定義する．

$\overline{D} = \{x_1, x_2, \cdots, x_n\}$

ここで，x_i は $X_i \overset{i.i.d}{\sim} D(r,p)$ の実現値で，$x_i \in D$ である．4 節で述べた，基本イベントに関するテストでは，その故障データセットが母集団 D であり，基本誤認識率は，「1 − 認識率」である．

例として，数字 2 の画像にノイズが追加されたデータセットを考える．これは，紙が汚れてノイズが付加された状況を表現するデータセットであり，図 2 の基本イベント E_2 の故障データセットの一部である．ノイズが付加された画像は大量にあるので，すべての画像を列挙できない．この大量のデータセットを母集団 D_{add}，母集団分布 $D_{add}(r_{add}, p_{add})$ で表現する．ここで，期待する認識率が 80%以上，すなわち，基本誤認識率 20%以下であるとすると，テストスートは $(D_{add}, r_{add}, 0.8)$ となる．この母集団の画像を分類器で認識した場合，それは 2 と判別されることを期待する．よって，期待値 r_{add} は 2 である．しかしながら，真の認識率 p_{add} は，存在するが，未知である．よって，標本 $\overline{D}_{add}$ により，p_{add} を推定する．ここで，標本 $\overline{D}_{add}$ として，2 の元画像にランダムにノイズを追加した画像を用いることができる．

テストスイート (D, r, p_e) のテストは，その標本データセット $\overline{D}$ により実施する．そして，標本データセット集合でテストをした結果から，真の認識率 p が，期待認識率 p_e 以上であると推定される時，テストはパスしたものとし，そうでない時，失敗したものとする．ここでの失敗は，真の認識率が期待認識率未満であることを意味するものではない．真の認識率が期待認識率以上であることが統計的に言えないだけであり，保留の意味合いが強い．

5.2 テスト結果の区間推定による統計的評価

標本データセット $\overline{D}$ を用いてテストを実施した結果を，統計的に評価する．統計的な評価では，SMC(Statistical Model Checking)[17], [18] などで用いられている，ベルヌーイ試行に基づいた区間推定の枠組みを使用する．区間推定では，信頼係数 $1-\delta$ にに対して，未知の部数 θ の値が $Pr(L \leq \theta \leq U) = 1-\delta$ となる信頼区間 $[L, U]$ を獲得する．以下では，標本データセットの要素 $x \in \mathcal{D}$ を入力として分類した結果を $CF(x)$ で表現する．標本データセット $\mathcal{D}$ を用いたテストでは，それぞれの要素 $x_i \in \mathcal{D}$ に対して，$CF(x_i) = r$ であることを確認する．つまり，認識結果が期待値にあっているか，あっていないかの試行であり，ベルヌーイ試行である．ベルヌーイ試行の区間推定では，Chernoff-Hoeffding bound[19] から，以下が成立する．

系 1 Chernoff-Hoeffding bound
テストスイート (D, r, p_e) の標本データセットを $\mathcal{D}$ とする．ここで，テストスイートの母集団分布は $D(r,p)$ である．
$x_i \in \overline{D}$ に対して，c_i を以下で定義する．

$$c_i = \begin{cases} 1 & CF(x_i) = r \\ 0 & otherwise \end{cases}$$

標本データセットをテストすることにより，標本認識率 $\overline{p}$ は以下で計算することができる．

$\overline{p} = \frac{1}{n}\sum_{i=1}^{n} c_i$

そして，信頼係数 $1-\delta$，誤差 ϵ の時，以下が成立する．

$n \leq \frac{1}{2\epsilon^2}\ ln\ \left(\frac{2}{\delta}\right)$

$Pr(|\overline{p} - p| \leq \epsilon) \geq 1-\delta$

以上の事実により，テストスイート (D, r, p_e) の標本データセット $\overline{D}$ を用いたテストは以下のように実施できる．

(1) 信頼区間 ϵ と信頼係数 $1-\delta$ を決定する．
(2) 以下で定義される n 個の素を含む標本データセット $\overline{D}$ を準備する．
$n = \left\lceil \frac{1}{2\epsilon^2}\ ln\ \left(\frac{2}{\delta}\right) \right\rceil$
(3) $\overline{D}$ のそれぞれの要素に対してテストを実施する．実施結果は以下で判定する．
- $|\{x \in \overline{D} | CF(d) = r\}| \geq n(p_e + \epsilon)$ の場合，パス．
- それ以外，失敗．

つまり，$(1-\delta) \times 100\%$の確率で認識率が p_e 以上の時，パスであり，それ以外の時は失敗である．

例として，テストスイート $(D, r, 80\%)$ を考える．$\delta = 0.1, \epsilon = 0.05$ の時，$n = 600$ である．よって，600 回中 510 回以上正解であれば，90%の確率で真の認識率 p が 80%以上であることが保証できる．

6. 実験

6.1 実験対象

実験対象として，MNIST を用いた畳み込みニューラルネットワーク (CNN) による手書き数字文字の分類器を選択した．MNIST では，画像サイズは 784 ピクセル (28 × 28) であり，学習用として 60,000 枚，評価用として 10,000 枚の画像が提供されている．また，ニューラルネットワークフレームワーク Caffe を用いて CNN を実現した．CNN の構成は，入力層 784(28·28)，畳み込み層 11520(24·24·20)，プーリング層 2880(12·12·20)，畳み込み層 800(4·4·50)，全結合層 500(500·1)，出力層 10(10·1) である．そして，MNIST の学習用画像を用いて学習済みの分類器を対象として，提案手法を用いて妥当性評価を行った．テストにおいては，Python および OpenCV を用いて標本データセットを自動生成した．FT4D でテストする状況が与えられるので，オリジナルの MNIST の評価用画像を，その状況を表現するように，乱数を用いて加工した．使用したパソコンは，CPU は Intel Core i5-7400，RAM は 8.0GB，OS は Ubuntu 14.04 LTS である．

6.2 実験結果

まず，ノイズによって手書き数字の認識を誤る状況をルートイベントとして，文字認識の障害となる状況の分析を行い，FT4D を作成した．作成した FT4D は，深さが 3，基本イベントが 21 個，中間イベントが 14 個であった．その一部を図 4 に示す．イベントに付加されている p_f, p_e, p_b は，それぞれ，故障率，誤認識率，基本誤認識率を表現している．ルートイベントにおいて期待する誤認識率は 0.5% であり，故障率は 10% である．すなわち，10%の確率でなんらかの障害が発生し，その障害が発生したとしても，99.5% の確率で正しく認識して欲しい，ということである．

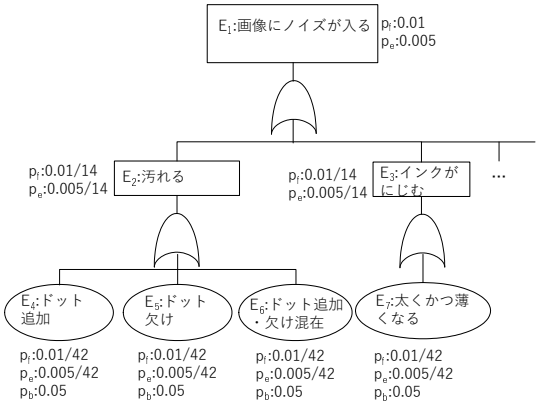

図 4 手書き文字認識の FT4D

次に，基本イベントに関して，テストを実施した．信頼係数 $1-\delta$ は 0.01，誤差 ϵ は，それぞれ，0.05 と 0.025 の 2 つの場合で実施した．標本データセットは，前述のとおり，OpenCV と Python により乱数を用いて自動生成した．例を図 5 に示す．左がオリジナルで右が加工後の画像である．

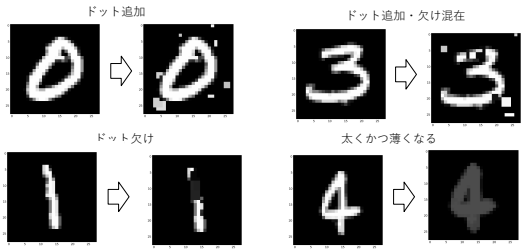

図 5 生成画像の例

図 4 に示した部分のテスト結果を表 1 に示す．左から順に，対象基本イベント，正解した入力の数，テストの判定，テストにかかった時間である．$\epsilon = 0.025$ の場合は，それぞれの基本イベントに関して，4240 回テストを実施し，4134 回以上正しく認識すれば，テストはパスする．しかしながら，すべての基本イベントに関してパスしなかった．これは，ϵ の値が大きいため，多めに正しく認識しなければならいからである．そこで，ϵ の値を 0.005 に設定した．この場合は，105967 回テストを実施し，100672 回以上正しく認識すれば，テストはパスする．結果は，$E_4, E_5,$ はパスしたが，E_6, E_7 はパスしなかった．これにより，E_6, E_7 に関しては，期待する認識率を統計的に保証することは期待できないことが伺える．

E_6 に関しては，FT4D を見直す必要がある．例えば，E_6 が発生する頻度を低くする機構を導入したり，全体の誤認識率の基準を甘くするなどが考えられる．E_6 の想定する発生頻度は，0.1/42 であるが，0.006/42 にまで抑えこむことができれば，基本誤認識率を 0.094 にすることができ，テストをパスすることになる．E_7 に関しても，同様に，対処することが可能である．このように，安全分析とテストは関連づけられており，テスト結果に基づいて，安全水準を調整するなどのトレードオフも行うことができる．

$\epsilon = 0.025$, テスト数: 4240, 4134 $\leq$

イベント	正解数	判定	時間 (秒)
E_4	4101	失敗	47.59
E_5	4050	失敗	46.83
E_6	3889	失敗	48.00
E_7	4015	失敗	44.76

$\epsilon = 0.005$, テスト数: 105967, 100672 $\leq$

E_4	101846	パス	1190.79
E_5	101744	パス	1167.75
E_6	97350	失敗	1209.88
E_7	100607	失敗	1113.84

表 1 テスト結果

7. 議論

6 節の実験で示したように，提案手法では，様々な状況を想定したデータセットを用いて，分類器の系統的に妥当性を評価できた．FT4D を用いて，障害により入力画像が影響を受ける場合を分析し，テスト実施可能な詳細度の基本イベントまで分解する．そして，個々の基本イベントに対して，データセットを作成し，期待する水準の認識率 (誤認識率) が達成できるかテストする．安全分析からテストまで系統的に実施できていることがわかる．

本手法では，安全分析，テストの両方において，定量的な評価が可能である．安全分析では，障害が発生する頻度 (故障率)，誤認識率，基本誤認識率を用いて定量的に分析する．故障木には，データセットに基づいた意味が存在し，イベント間において故障率，誤認識率の計算も可能である．テストにおいても，統計的推定を用いて，定量的な認識水準を保証することができる．また，安全分析とテストは関連づけられており，テスト結果に基づいて，安全水準を調整するなどのトレードオフも行うことができる．

機械学習はデータセットにより学習と評価を行う．データセットの内容によっては，大きく挙動が変わってくる．よって，データセットのバリエーションを扱うことは，重

要で本質的であると考えている．また，システムが正しく認識する水準は，入力するデータセットの内容に大きく依存している．従来の手法のように，大雑把に認識率とデータセットを考えるのではなく，それらのモデル化が必要である．本論文で提案した FT4D は，安全分析の観点から，データセットのバリエーションをモデル化するものである．

本手法は，データセットを自動生成することを想定して提案しているが，そうで無い場合も適用できる．FT4D は，テスト手法とは独立であり，特定の状況を表現するデータセットの分析と，それらによる評価にも用いることができる．テスト手法においても，特定の状況で収集したデータセットは標本であると見なすことができるので，収集できなかったデータを含めデータセット全体 (母集団) における認識率の評価を行うことができる．

一方，提案手法では，画像データのような，比較的単純なデータセットについて取り扱っている．他にも，動画や時系列データなどを取り扱うものもある．現在の FT4D では，AND/OR ゲートによるデータセットの整理で十分であるが，これらのデータセットを扱うためには，新たなゲートや整理手法が必要になるかもしれない．

8. まとめ

本論文では，機械学習を用いて実現されたシステムの安全性を系統的に評価する手法を提案した．提案手法は，安全分析とテストから構成されている．安全分析では，データセットに基づいて故障木を定義しているため，状況を分析しながら，データセットの分類，期待する誤認識率の計算を行うことができる．また，獲得した基本イベントの誤認識率と故障率に基づいて，テストでパスすべき基準（基本誤認識率）を求めることができる．この基準に基づいて，テストを実施する手法も提案した．機械学習システムは，認識率など確率的な指標に基づいて妥当性を判定する．また，想定する入力の数も膨大である．そこで，統計的手法を用いてテスト結果を評価し，期待する水準を統計的に満たすかどうか判定する手法を提案した．

本論文では，教師あり学習に基づいた機械学習による分類器の評価を行った．今後は，教師なし学習に基づいたもの，分類問題以外のもの，などに，段階的に対象を広げて，提案手法を拡張する予定である．また，提案手法に限らず，機械学習システムの妥当性を評価する際には，期待する認識率を設定する必要がある．機械学習において，どれくらいの認識率を期待するのか，なんらかの基準や標準が必要である．高すぎる認識率は達成できないであろうし，低すぎるとチューニングの機会を失うことになる．適切な認識率の決定法についても検討していく．

参考文献

[1] G. Bradski and A. Kaehler:Learning OpenCV, O'Reilly, 2008.
[2] G. van Rossum: Python tutorial, Technical Report CS-R9526, Centrum voor Wiskunde en Informatica (CWI), May, 1995.
[3] W. Vesely, F. Goldberg, N. Roberts, D. Haasl: Fault tree handbook (nureg-0492), 1981.
[4] W. Lee, D. Grosh, F. Tillman, L.C.H.:Fault tree analysis methods and applications - a review, pp. 194–203, 1985.
[5] X. Xie, J.W.K. Ho, C. Murphy, G. Kaiser, B. Xu, and T.Y. Chen : Testing and Validating Ma- chine Learning Classifiers by Metamorphic Testing, JSS, 84(4), pp.544-558, 2011.
[6] S. Nakajima and H. N. BUI:Dataset Coverage for Testing Machine Learning Computer Programs, pp.297–304, APSEC, 2016.
[7] C.Murphy, G.Kaiser and M.Arias: An Approach to Software Testing of Machine Learning Applications, SEKE, 2007.
[8] X. Huang, M. Kwiatkowska, S. Wang and M. Wu: Safety Verification of Deep Neural Networks,Computer Aided Verification, pp. 3–29, 2017
[9] I. J. Goodfellow, J. Shlens and C. Szegedy, Explaining and harnessing adversarial examples, arXiv:1412.6572, 2014.
[10] I. Evtimov, K. Eykholt, E. Fernandes, T. Kohno, B. Li, A. Prakash, A. Rahmati and D. Song. 2017. Robust Physical-World Attacks on Machine Learning Models. arXiv preprint arXiv:1707.08945, 2017.
[11] Kexin Pei, Yinzhi Cao, Junfeng Yang, Suman Jana: DeepXplore: Automated Whitebox Testing of Deep Learning Systems, SOSP, pp.1–18, 2017.
[12] Yuchi Tian, Kexin Pei, Suman Jana, Baishakhi Ray: DeepTest: Automated Testing of Deep-Neural-Network-driven Autonomous Cars, ICSE, 2018.
[13] Tommaso Dreossi, Shromona Ghosh, Alberto L. Sangiovanni-Vincentelli, Sanjit A. Seshia: Systematic Testing of Convolutional Neural Networks for Autonomous Driving, ICML Workshop on Reliable Machine Learning in the Wild, 2017.
[14] Tommaso Dreossi, Alexandre Donze, Sanjit A. Seshia: Compositional Falsification of Cyber-Physical Systems with Machine Learning Components. NFM, pp.357-372, 2017.
[15] F.Ortmeier and G.Schellhorn: Formal Fault Tree Analysis - Practical Experiences, Electronic Notes in Theoretical Computer Science, Volume 185, pp.139–151, 2007,
[16] M.Ammar, K.A.Hoque and O.A.Mohamed: Formal analysis of fault tree using probabilistic model checking: A solar array case study, SysCon, pp. 1-6 , 2016.
[17] T. Herault, R. Lassaigne, F. Magniette and S. Peyronnet: Approximate probabilistic model checking, VMCAI, pp. 73–84, 2004.
[18] C. Jegourel: Rare event simulation for statistical model checking, Theses, Universite de Rennes 1, France, Nov. 2014.
[19] W. Hoeffding: Probability inequalities for sums of bounded random variables, Journal of the American Statistical Association, vol. 58, no. 301, pp. 1330, 1963.