Computer Security Symposium 2017
23 - 25 October 2017

検知システムの高精度化に関する提案

山本 匠[1]　西川 弘毅[1]　木藤 圭亮[1]　河内 清人[1]

概要：システムの故障の検知や攻撃活動の検知など，検知技術は様々な場面で利用されている．しかし検知技術の開発の時点で十分な数かつ十分なバリエーションの正常なサンプルと非正常なサンプル（故障や攻撃）を用意することは難しく，閾値などのパラメータを適切に設定することができない．それゆえ検知技術の実運用の際にしばしば誤検知や検知漏れが発生してしまう．そこで本研究では，正常なサンプルや非正常なサンプルを自動生成する技術を提案する．特に検知技術の評価に有用と考えられる，誤検知や検知漏れのサンプルを自動生成する技術のコンセプトについて本稿で述べる．プロトタイプの実装と評価については今後の課題とする．

キーワード：検知技術，検知システム，誤検知，検知漏れ，評価，マルウェア，攻撃

Proposal of an enhancement technique for detection systems

Takumi Yamamoto[1]　Hiroki Nishikawa[1]　Keisuke Kito[1]　Kiyoto Kawauchi[1]

Abstract: Detection systems are being used for various kinds of purposes such as detecting malfunctions in a system being monitored or detecting malicious activities in an enterprise network. However at developing detection systems, preparation of sufficient number of normal samples and abnormal (e.g., malfunctions and malicious activities) samples to represent all possible future events coming to the detection system is nearly impossible, which results in unignorable number of false positives (given a sample is normal but the sample is identified as abnormal) and false negatives (given a sample is abnormal but the sample is identified as normal) making the system less effective. Therefore, we propose an enhancement technique for detection systems, in which normal samples and abnormal samples are automatically synthesized and generated. In this paper, we focus on a technique generating samples likely to be false positive or false negative, which makes the detection systems being much more effectively and exhaustively evaluated. Implementations and evaluations of the proposed technique are left as future works.

Keywords: detection technique, detection system, false positive, false negative, evaluation, malware, attack

1. はじめに

システムの故障などの異常の検知や，マルウェア感染や攻撃活動の検知など，検知技術は様々な場面で利用されている．異常検知に関しては，プラントや工場などのシステムを監視するソリューションとして利用されることが多い[1,2,3]．正常なオペレーションに影響を与えないよう異常を適切に検知しかつ誤報の少ない検知技術が望まれる．

一方，不正検知技術については，これまで企業内のネットワークを狙った攻撃を対象とする技術が多かった[4]．しかし，近年，工場やプラントなどの制御システムへのサイバー攻撃が増加している．下水処理場のシステム[5]，人工衛星のシステム[6]，路面電車のシステム[7]，ドローンのシステム[8]など，様々な制御システムが攻撃の対象となったことが報告されている．また 2015 年には，ウクライナの変電所へのサイバー攻撃により大規模な停電が引き起こされている[9]．大きな被害にはならなかったものの，2010 年に発生した Stuxnet によるイランの原子力施設の破壊は世間に衝撃を与えた[10,11,12]．このように制御システムへの攻撃が増加しており，制御システムを対象とする検知技術の必要性が高まっている[13,14]．

また，昨今の攻撃者は，標的の組織やシステムに関する多くの知識を有していることが報告されている．例えば，文献[10]によれば，Stuxnet の開発者は SCADA や攻撃対象の制御システムに関する豊富な知識を持っていたと言われている．また IPA の報告によれば，2016 年の 10 大脅威の 1 つに「内部不正による情報漏えいとそれに伴う業務停止」が挙げられている[15]．このことからも，標的組織の情報を熟知した攻撃者が今後増えていくことが予想され，通常運転のシステムの動作に隠れるよう作りこまれた攻撃やセキュリティシステムを回避するような攻撃が増えていくと考えられる．そのため，攻撃検知についても，不正を適切に検知しかつ誤報の少ない検知技術が望まれる．

このように異常や不正を検知する高精度な検知技術が様々な場面で求められている．検知技術の精度を測る尺度としては，本来検知すべきではない正常な事象を検知してしまう誤報や誤検知（False Positive）と，本来検知すべき事象を検知しない検知漏れ（False Negative）の 2 つが良く注目される．誤検知と検知漏れの頻度は少なければ少ないほど高精度な検知技術と言える．

しかし一般的に誤検知と検知漏れの頻度をゼロにすることは困難である．その理由は，検知技術のアルゴリズム

†1 三菱電機株式会社
Mitsubishi Electric Corporation

© 2017 Information Processing Society of Japan

に由来するところもあるが，検知技術の開発の時点で十分な数かつ十分なバリエーションの正常なサンプルと非正常なサンプル（故障や攻撃）を用意することが難しく，閾値などのパラメータを適切に設定することができていないためと考えられる．これは開発で検証に利用した正常なサンプルや非正常なサンプルが少なければ，実運用のときに現れる事象を一様にカバーすることができないためである．また一般的に故障や攻撃は高い頻度で発生する事象ではないため，このような非正常なサンプルを大量に収集することは難しい．

そこで本研究では，この問題を解決するために，正常なサンプルや非正常なサンプルを自動生成する技術を提案する．特に検知技術の評価に有用と考えられる，誤検知や検知漏れのサンプルを自動生成する技術のコンセプトについて本稿で述べる．プロトタイプの実装と評価については今後の課題とする．

2. 関連技術

本章では，セキュリティ評価技術や提案方式に関連のある既存技術について簡単に紹介する．

2.1 セキュリティ評価技術の大別

製品やシステムのセキュリティ評価や脆弱性の有無の検証を行う技術として，Fuzzing，脆弱性スキャナ，ペネトレーションテストが一般的に知られている．本節ではそれぞれの技術について簡単に紹介する．本節の内容は IPA の公開資料に基づいている[16,17].

●Fuzzing（ファジング）

ファジングによる検査は, PC 向けソフトウェアや組込ソフトウェア等に対して脆弱性を発現させやすいデータやファイルを送り込み，脆弱性の有無を検査する方法である．Taof[18]や Peach Fuzzer[19]がオープンソースのファジングツールとしてよく知られている．有償ではあるが制御システム向けのファジングツールとしては Achilles が知られている[20].

●既知の脆弱性検査（スキャナ）

対象とするマシンが使用しているソフトウェアに既知の脆弱性が無いかどうかを調査する検査である．さらに設定ミスや脆弱なパスワードの存在なども確認できる．ファジングとの大きな違いは，既に問題が見つかっているデータに絞って検査（既知の脆弱性の有無そ確認する検査）を行うことである．オープンソースのスキャナとしては OpenVAS[21]が知られている．有償なものとしては Nessus[22]や Nexpose[23]が有名である．

●侵入テスト（ペネトレーションテスト）

侵入テストは組織のサーバやネットワークシステムに対して攻撃者が実際に侵入できるかどうかという点に着目して検査を行う．そのため，運用上のシステムに残存している既知の脆弱性を狙ったり，設計段階での不備を突いたりすることで実行される．オープンソースの侵入テストツールとしては Metasploit Framework[24]が有名である．有償なものとしては Core Impact[25]がよく知られている．

侵入テストでは，ファジングやスキャナによって発見した脆弱性を使って，製品やシステムに侵入することが可能かを確認する．さらに，いくつかの製品やシステムの侵入を経由して，目的の製品やシステムに侵入が可能かについても検査することができる．提案方式は侵入テストに該当する．

2.2 関連する既存技術

提案方式に関連のある既存技術を紹介する．

●文献[26]

本技術では，攻撃データ（バイナリデータ）を効率的に作るために，攻撃データのバイト列そのものを 1 バイトずつ正常データに近づけ，それをシステムに入力し，システムが異常を起こすバイナリデータを特定する．これにより正常データの特徴をよく持つ攻撃データを自動生成し，システムの強化を図る．

本技術はファジングに該当し，システムの異常を見つけることを目的とする技術である．そのため生成された攻撃データが攻撃として成立する（例えば，侵入して不正なプログラムを実行し，インターネット上の攻撃者のサーバと通信をするなど）かまでは確認することができない．

●文献[27]

本技術では，マルウェアなどの不正プログラムに変異を与え，アンチウィルスソフトウェアなどの既存の不正プログラム検知システムでは検出できないような不正プログラムのサンプルを作成する．新しく生成されたサンプルが既存システムで検知されないこと，悪意のある機能を維持していることを検査し，検査をパスしたサンプルを使って，対象の不正プログラム検知システムを強化する．

●文献[28]

文献[27]と類似の研究である. 本研究では, 強力な攻撃者を仮定し, マルウェア検知用の識別器の頑健性を評価する．そのために，識別器を回避するマルウェアの自動生成技術を提案している．本研究では，PDF マルウェアに焦点を置き，検知システムを回避するマルウェアの亜種を，遺伝的アルゴリズムを利用して自動生成する．本研究で対象にしている検知システムは機械学習を利用したものであり，機

© 2017 Information Processing Society of Japan

械学習して得られた識別器の分類境界をまたぐようにマルウェアの亜種を生成していく．

●文献[29]

本研究では，人工知能技術により高度化されるマルウェアについて考察を行っている．特に，人工知能技術の一つである進化計算を利用した攻撃自動生成の検討を行っている．進化計算を使うことで攻撃と防御を進化させ続けることができるという仮説のもと，構想が進められている．攻撃を進化させて，それをもとに防御技術の精度向上を目指す点に関して，本研究も同様のアプローチをとっている．

3. 提案方式

3.1 コンセプト

検知漏れや誤検知のサンプルを大量に生成するために，対象システムから収集可能な正常なサンプルと既存の検知システムを利用する．検知漏れサンプルの生成については，対象システムから正常なサンプルを収集し，正常サンプルを表現するモデル（以降，正常モデルと呼ぶ）の学習を行う．また正常サンプルと既存の検知システムを利用して誤検知サンプルを表現するモデル（以降，誤検知モデルと呼ぶ）の学習も行う．攻撃などの非正常サンプルを生成するツールを利用して，正常モデルに近くなるようサンプルを生成することで検知漏れを起こす可能性のあるサンプルを生成する．一方，誤検知サンプルの生成については，正常サンプルを生成するツールを利用して，誤検知モデルに近くなるようサンプルを生成することで誤検知を起こす可能性のあるサンプルを生成する．以下ではより詳細な手順を記載する．

3.2 手順

3.2.1 モデルの作成

●正常モデルの学習

正常モデルの学習手順は以下のとおりである．

step1. 対象のシステムから正常サンプルを収集する
step2. 正常サンプルから特徴ベクトルを抽出する
step3. 正常サンプルの特徴ベクトルを使って正常モデルを学習する

まず step1 では，対象のシステムから真に正常なサンプルを収集する．サンプルとは，例えば，通信パケット，プロキシログ，システムログ，メール，ファイルなどである．サンプルに対応するセンサを対象システムに設置し，サンプルを収集する．この時点では，正常ではないサンプル（非正常なサンプル）は対象システムに含まれていないものとする．実システムではなく，仮想的に実システムを模擬した模擬環境から正常サンプルを収集しても良い．

なお，別の環境から収集した正常サンプルを利用しても良い．この場合，環境に依存する情報を対象のシステムの環境に合わせるように編集が必要である．例えばサンプルがログの場合，タイムスタンプ，IP アドレス，ホスト名，ドメイン名などが，対象システムの環境とは異なる可能性がある．そこで，収集したログの情報と整合するよう，タイムスタンプ，IP アドレス，ホスト名，ドメイン名などの情報を修正する．

step2 では，サンプルの生データを前処理し，あらかじめ決めた特徴情報を抽出する．例えば，プロキシログの場合，一定期間におけるある送信元とある送信先の通信に関して，通信の頻度，やりとりするデータサイズ，データに含まれる文字列の頻度などの情報を，要素として並べたものを特徴ベクトル C =(c1,c2,…,cn)とすることができる．

step3 では，収集した正常サンプルのセットを学習データとして，機械学習などの技術を用い，正常サンプルを表現する特徴空間をモデル（正常モデル）として学習する．機械学習を利用する場合，正常サンプルかどうかを識別するための 1 クラス識別器を利用する．与えられたサンプルが正常モデルと似ているかを判断するために識別器の識別スコアを利用する．スコアは与えられた特徴ベクトルが正常サンプルとどれだけ似ているかを表す値（類似度）であり，正常サンプルのモデルに近ければ高い値になり，似てなければ低い値になる．機械学習の識別器の場合，スコアは予測値の確率にあたる．

もし，攻撃などの非正常サンプルも収集できるならば，非正常サンプルのセットも利用し，正常サンプルか，非正常サンプルかを分類する 2 クラス識別器を利用しても良い．2 つのセットの大きさに差がある場合は，サンプル数を調整したり，誤答の際のペナルティを調整したりするなど，不均衡データに対して良く利用されるアプローチを採用する[30].

●誤検知モデルの学習

誤検知モデルの学習手順は以下のとおりである．

step1. 対象のシステムから正常サンプルを収集する
step2. 既存の検知システムを利用して，誤検知サンプルを収集する
step3. 誤検知サンプルから特徴ベクトルを抽出する
step4. 誤検知サンプルの特徴ベクトルを使って誤検知モデルを学習する

まず step1 では，正常モデルの学習と同様に，対象のシステムから真に正常なサンプルを収集する．

step2 では，収集した正常サンプルと既存の検知システムを利用して，既存の検知システムが誤検知を起こす正常なサンプルを収集する．

step3 で，正常モデルの学習と同様に，サンプルの生データを前処理し，既定の特徴を抽出する．

最後に step4 で，誤検知を起こさない正常なサンプルの

© 2017 Information Processing Society of Japan

セットと誤検知を起こす正常なサンプルのセットを学習データとして，機械学習などの技術を使い，誤検知を起こす正常なサンプルを表現するモデルを学習する．この場合，誤検知を起こさない正常サンプルか，誤検知を起こす正常サンプルかを分類するための 2 クラス識別器を利用する．学習データの大きさに差がある場合は，不均衡データに対して良く利用されるアプローチ[30]を採用し対応する．

図 1〜図 6 にサンプルの特徴空間のイメージを示す．図 1 の W と B の空間がそれぞれ正常なサンプルの集合と非正常なサンプルの集合を表す．図 2 の NM は正常モデルが表現するサンプルの特徴空間である．必ずしも正確に正常なサンプルを学習できるわけではないため，NM には非正常なサンプルが多少は含まれることになる．D はある検知システムが検知するサンプルの集合である．B の中で D に覆われていない空間が検知漏れの集合として表される．検知システムは正常なサンプルを誤って検知してしまうこともあるため，D には正常なサンプルも含まれる．つまり，図 3 に示す FP が誤検知サンプルの集合を表す．そして，図 4 に示す FPM が，誤検知サンプルから生成した誤検知モデルが表現するサンプルの空間である．必ずしも正確に誤検知サンプルを学習できるわけではないため，FPM には誤検知とならない正常サンプルや非正常サンプルが多少は含まれることになる．

図 1　正常なサンプルと非正常なサンプルの特徴空間（イメージ図）

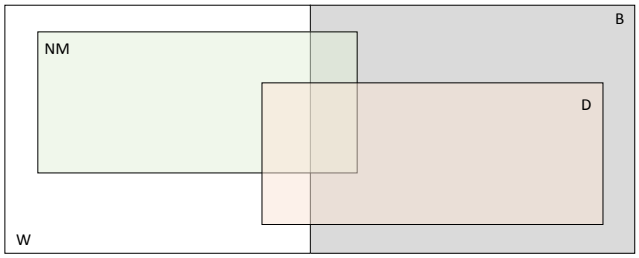

図 2　正常モデルが表現するサンプルの空間と誤検知サンプルの空間（イメージ図）

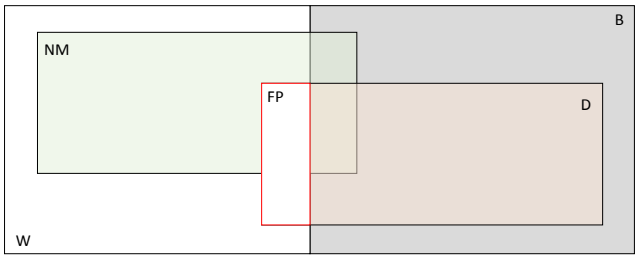

図 3　誤検知サンプルの集合（イメージ図）

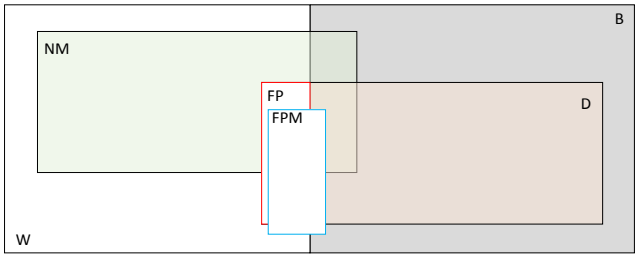

図 4　誤検知モデルが表現するサンプルの空間（イメージ図）

3.2.2 検知漏れサンプルの生成

検知漏れサンプルの生成手順は以下のとおりである．

step1. 攻撃などの非正常サンプルを生成する

step2. 攻撃サンプルを正常モデルに近くなるよう調整する

step3. 調整した攻撃サンプルが既存の検知システムで検知されるかを確認する

step4. 検知されない攻撃サンプルの攻撃機能の有無を確認する

step5. 既存の検知システムで検知されず，攻撃機能も有するサンプルを収集する

step1 では，既存の攻撃生成ツールなどから対象のシステムを攻撃するための攻撃サンプルを生成する．攻撃サンプルの生成は，仮想環境で模擬的に用意した対象システムで実行することを想定する．攻撃生成ツールは，攻撃サンプルの特徴を操作するための入力パラメータを有する．パラメータとは，例えば，エクスプロイトコードを実行する対象（IP アドレス，ポート番号，サービス），侵入後にインストールするマルウェアの種類，攻撃者の C&C サーバの IP アドレス，通信の暗号化方法，C&C 通信の頻度やサイズ，情報漏えいのための POST のサイズと頻度など様々な情報である．

step2 では，攻撃サンプルから特徴ベクトル C =(c1,c2,…,cn)を作成する．そして特徴ベクトルの要素を変更できるかを確認する．具体的には，与えられた特徴ベクトルである C =(c1,c2,…,cn)の各要素がとりうる範囲（LBi ≦ ci ≦ UBi）を定義しておき，その範囲の中で得られる可能な特徴ベクトル全てに対して確認を行う．LBi と UBi はそれぞれ ci の下限と上限である．変更できる場合は，要素を変更し新たな特徴ベクトル C' を用意する．正常モデル（識

© 2017 Information Processing Society of Japan

別器 E）を使い，スコア S(*C'*)を算出する．算出されたスコア S(*C'*)が規定の閾値以上の場合，特徴ベクトルは正常モデルに近いと判断する．

次に step3 では，対応する攻撃生成ツールを利用して，正常モデルに近い特徴ベクトル *C'* に対応する攻撃を実行する．特徴ベクトルの各要素（プロキシログを使った不審通信検知の場合，例えば，1 回あたりの Post の サイズ平均 [bytes/time]，Post の周期 T [sec]などの通信の特徴）を調整可能な攻撃ツールである．対象のシステムにおいて検知したい事象（攻撃，故障，異常）に合わせて，攻撃ツールはあらかじめ用意する．

step4 では，実行された攻撃が攻撃機能を維持しているかを確認する．検知システムの監視対象がログの場合，ログを発生させた攻撃が，基本機能（例えば，ファイル操作，ユーザ認証，プログラム起動，外部への情報アップロードなど）を行っているかを監視する．基本機能の有無を判断するために Syslog や通信ログなどのログをパージングし，該当の操作に関するログが存在するかを確認する．

検知システムの監視対象がメールの場合，生成された不正メールが基本機能（例えば，メールを送りつけられた人物が，誤って不正メールの文面にある URL や添付ファイルを実際にクリックしてしまう）を確認する．組織の不審メール対応訓練の一環として，生成された不正メールを組織の人間に送り，不正メールの文面にある URL や添付ファイルを実際にクリックするかを監視する．添付ファイルには，クリックすると特定の URL にアクセスするようにプログラムされたスクリプトが記述されている．ドキュメントファイルと誤認するように PDF などのドキュメント用のアイコンを添付ファイルには利用する．

検知システムの監視対象が通信の場合，生成された攻撃通信が基本機能（例えば，RAT の操作，C&C サーバとのやりとり，ファイルアップロードなど）を行っているかを監視する．

検知システムの監視対象がファイルの場合，生成された不正ファイルが基本機能（例えば，プログラムの実行，ファイルの削除，C&C サーバとの通信，ファイルアップロードなど）を行うかを確認する．

最後に step5 で，検知システムに検知されず，攻撃機能も維持しているサンプルを収集する．

図 5 に検知漏れサンプル生成のイメージを示す．検知システムに検知される攻撃サンプルの特徴ベクトルを，正常モデルに含まれ，検知システムが検知しない領域に収まるよう徐々に変化させていく．

なお，step2 で，特徴ベクトルを調整する際に，特徴ベクトルに対する制約を設定しても良い．ここで言う制約とは，監視対象としている攻撃において必須の機能（条件）である．

プロキシログを使った不審通信検知を例に説明する．特徴ベクトルに以下の要素を含むとする．

c1: 1 回あたりの POST データの サイズ平均 [bytes/time]
c2: POST の周期 T [sec]

不審通信は，攻撃者のサーバとのやりとりや機密データの送信が目的である．そのため，攻撃機能としての必須の条件として，例えば「一定時間当たりの POST データサイズが規定の閾値より大きい」，すなわち，「$c1 \times (1/c2) > \theta$ [sec]」を制約として定義することができる．

制約を満たさない特徴ベクトルについては，検知システムに検知されるかどうかや攻撃機能の有無を確認するまでもなく不適であり，検知漏れサンプル生成の処理を効率化することができる．

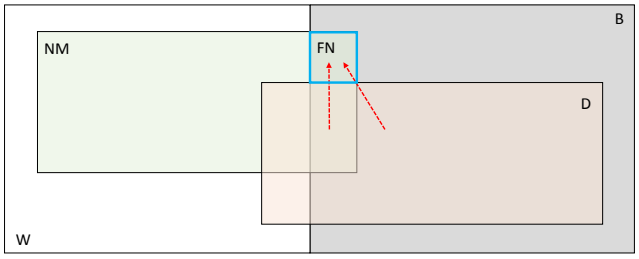

図 5　検知漏れサンプル生成のイメージ

3.2.3 誤検知サンプルの生成

誤検知サンプルの生成手順は以下のとおりである．

step1. 正常または非正常サンプルを生成する
step2. 生成したサンプルが誤検知モデルに近くなるよう調整する
step3. 調整したサンプルが既存の検知システムで検知されるかを確認する
step4. 検知されるサンプルの攻撃機能の有無を確認する
step5. 既存の検知システムで検知され，攻撃機能を有さないサンプルを収集する

step1 では，正常なサンプルを生成するツールなどから対象のシステム上で正常なサンプルを生成する．サンプルの生成は，仮想環境で模擬的に用意した対象システムで実行することを想定する．サンプル生成ツールは，サンプルの特徴を操作するための入力パラメータを有する．パラメータとは，例えば，通信をする対象（IP アドレス，ポート番号，サービス），実行するプログラムの種類，通信の暗号化方法，情報をアップロードするための POST のサイズと頻度など様々な情報である．サンプル生成ツールは，攻撃生成ツールの攻撃機能を無効化したものと捉えてよい．

step2 では，サンプルから特徴ベクトル *C* =(c1,c2,…,cn)を作成する．そして特徴ベクトルの要素を変更できるかを確認する．変更できる場合は，要素を変更し新たな特徴ベク

© 2017 Information Processing Society of Japan

トル C' を用意する．誤検知モデル（識別器 E'）を使い，スコア S'(C')を算出する．算出されたスコア S'(C')が規定の閾値以上の場合，特徴ベクトルは誤検知モデルに近いと判断する．

次に step3 では，対応するサンプル生成ツールを利用して，誤検知モデルに近い特徴ベクトル C' に対応するサンプルを実行する．検知漏れサンプルの生成と同様に実施する．

step4 では，実行されたサンプルが攻撃機能を持ってしまっているかを確認する．確認の仕方は検知漏れサンプルの生成と同じである．

step5 で，既存の検知システムで検知され，攻撃機能を持たないサンプルを収集する．

図 6 に誤検知サンプル生成のイメージを示す．正常なサンプルや攻撃サンプルを，攻撃機能を持たないことを確かめながら，誤検知モデルが表現する領域に収まるよう徐々に変化させていく．

なお，検知漏れサンプル生成のときと同様に，step2 で，特徴ベクトルを調整する際に，特徴ベクトルに対する制約を設定しても良い．ここで制約とは，監視対象とする攻撃において必須の機能（条件）の否定である．

検知漏れサンプル生成の例を再び利用すると，攻撃機能としての必須の条件の否定として，「一定時間当たりの POST データサイズが規定の閾値以下」，すなわち，「c1×(1/c2) ≦ θ [sec]」が制約となる．

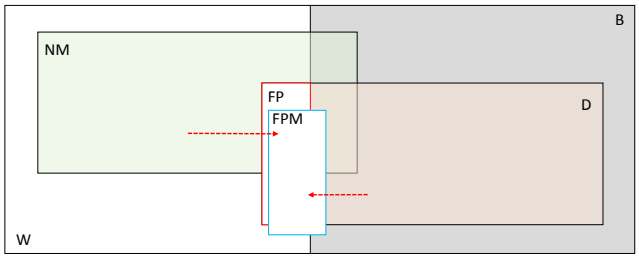

図 6　誤検知サンプルの生成のイメージ

4. 考察

4.1 効果

本来，正常サンプルに比べ，非正常なサンプルは圧倒的に少ない．そのため，これまでは検知システムを適切に評価することが難しかった．提案方式により人工的にではあるものの，非正常なサンプルである検知漏れサンプルや正常なサンプルではあるが検知システムがエラーを起こすような誤検知サンプルを大量に生成することできる．これにより，より多くのサンプルを使って検知システムを評価することができるため，検知システムの強化が期待される．

4.2 効率化

ある正常サンプルが既存の検知システムで誤検知されるということは，既存の検知システムが検知の際に注目する非正常な特徴を，同サンプルが多く含んでいると解釈することができる．誤検知されない正常サンプルと，誤検知される正常サンプルの特徴の差異から，既存の検知システムが検知の際に注目する非正常な特徴を推測することで，より効率的に検知漏れサンプルの生成を実現することが期待される．そこで次のように提案方式の改良を考える．

step1. 寄与度を算出できるアルゴリズムで誤検知モデルを作成する
step2. 各クラスへの分類における各特徴の寄与度を計算する
step3. 寄与度の大きな特徴を既存の検知システムが検知時に注目する非正常な特徴として選択する
step4. 検知漏れサンプル生成の際に，step3 で選択した特徴を，検知システムの検知を回避する方向へ調整することで，検知漏れサンプルを生成する

まず step1 では，誤検知サンプル生成のときと同様に，誤検知を起こさない正常サンプルか，誤検知を起こす正常サンプルかを分類するための 2 クラス識別器を学習する．このとき特徴の寄与度を算出できるアルゴリズムを利用する．例えば Random Forest では，識別に利用する各特徴の寄与度を算出することができる．

次に step2 で，各特徴の寄与度を算出する．識別に利用する寄与度が大きいほど識別に重要な特徴であることがわかる．寄与度以外にも，学習済みのモデルの重み係数，オッズ比，感度分析結果などから，識別に強く影響を与える特徴を推測することができる．

step3 で，寄与度が既定の閾値以上の特徴を選択する．これにより，既存の検知システムが検知を起こす（誤検知を起こす）際に影響のある特徴を抽出することができる．

最後に step4 で，抽出した特徴を，3.2.2 節「検知漏れサンプルの生成」における step2 で修正する要素とし，正常なモデルに近づくように特徴を修正する．

この方法により，非正常さの判定に大きな影響を持つ特徴のみに絞ることで，検知漏れサンプルの生成を効率化することができると期待される．

5. まとめと今後の課題

本稿では，検知漏れサンプルや誤検知サンプルを自動的に大量に生成するアイデアを提案した．提案方式により生成されたサンプルを利用して検知システムを評価することで，検知技術をより精度よく設計または検証することができると期待される．

今後は，情報システムや制御システムなどの模擬環境を用い，提案方式の効果を評価する予定である．

© 2017 Information Processing Society of Japan

参考文献

[1]. 花森利弥，西村利浩，システムの異常予兆を検知するリアルタイム監視ソリューション，https://www.fujitsu.com/jp/documents/about/resources/publications/magazine/backnumber/vol67-2/paper04.pdf（2017 年 8 月 21 日確認）.

[2]. 富士電機，ビッグデータ解析を用いたプラント異常検知，http://www.fujielectric.co.jp/about/technology/theme/bigdata_plant.html（2017 年 8 月 21 日確認）.

[3]. 加川 和伸，NEC，プラント故障予兆検知サービスのグローバル展開，http://jpn.nec.com/techrep/journal/g15/n01/pdf/150118.pdf（2017 年 8 月 21 日確認）.

[4]. 三菱電機，数億種類のウイルスの活動を 50 個程度の「攻撃手口」に分類・検知し，情報流出を阻止 「サイバー攻撃検知技術」を開発，http://www.mitsubishielectric.co.jp/news/2016/0217-f.html（2017 年 8 月 21 日確認）.

[5]. Marshall Abrams and Joe Weiss, Malicious Control System Cyber Security Attack Case Study– Maroochy Water Services, Australia, http://csrc.nist.gov/groups/SMA/fisma/ics/documents/Maroochy-Water-Services-Case-Study_report.pdf（2017 年 8 月 21 日確認）.

[6]. DailyWireless.org, SkyNet Satellite Hacked?, http://www.dailywireless.org/2007/05/08/skynet-satellite-hacked/（2017 年 8 月 21 日確認）.

[7]. The Telegraph, Schoolboy hacks into city's tram system, http://www.telegraph.co.uk/news/worldnews/1575293/Schoolboy-hacks-into-citys-tram-system.html（2017 年 8 月 21 日確認）.

[8]. Global Research, Israeli Intelligence Report: US Drone downed by Iran Cyber Attack, http://www.globalresearch.ca/israeli-intelligence-report-us-drone-downed-by-iran-cyber-attack/28114（2017 年 8 月 21 日確認）.

[9]. SANS Industrial Control Systems Security Blog, Confirmation of a Coordinated Attack on the Ukrainian Power Grid, https://ics.sans.org/blog/2016/01/09/confirmation-of-a-coordinated-attack-on-the-ukrainian-power-grid（2017 年 8 月 21 日確認）.

[10]. CSMonitor.com, How Stuxnet cyber weapon targeted Iran nuclear plant, https://www.csmonitor.com/USA/2010/1116/How-Stuxnet-cyber-weapon-targeted-Iran-nuclear-plant （ 2017 年 8 月 21 日確認）.

[11]. Eset, "Stuxnet Under the Microscope", https://www.esetnod32.ru/company/viruslab/analytics/doc/Stuxnet_Under_the_Microscope.pdf （2017 年 8 月 21 日確認）.

[12]. 日本シーサート協議会，"マルウェア Stuxnet(スタクスネット) について", http://www.nca.gr.jp/2010/stuxnet/（2017 年 8 月 21 日確認）.

[13]. 三菱電機，重要インフラの安定したサービス提供に貢献「サイバー攻撃検知技術」を開発，http://www.mitsubishielectric.co.jp/news/2017/0517.html

[14]. NEC, 未知のサイバー攻撃を自動検知する自己学習型システム異常検知技術（ASI），http://jpn.nec.com/techrep/journal/g16/n01/160111.html（2017 年 8 月 21 日確認）.

[15]. IPA 独立行政法人 情報処理推進機構，"情報セキュリティ 10 大脅威 2016 ～個人と組織で異なる脅威，立場ごとに適切な対応を～", https://www.ipa.go.jp/files/000051691.pdf（2017 年 8 月 21 日確認）.

[16]. IPA 独立行政法人 情報処理推進機構，"情報セキュリティ 10 大脅威 2016 ～個人と組織で異なる脅威，立場ごとに適切な対応を～", https://www.ipa.go.jp/files/000051691.pdf（2017 年 8 月 21 日確認）.

[17]. IPA 独立行政法人 情報処理推進機構，"ファジング：FAQ", https://www.ipa.go.jp/security/vuln/fuzz_faq.html#011（2017 年 8 月 21 日確認）.

[18]. Taof, "The art of fuzzing beta", https://sourceforge.net/projects/taof/files/（2017 年 8 月 21 日確認）.

[19]. Peach Fuzzer, "Discover unknown vulnerabilities", http://www.peachfuzzer.com/（2017 年 8 月 21 日確認）.

[20]. 日本ダイレックス，"Achilles|制御システム・セキュリティ・ロバスト性テストツール", http://www.direx.com/product.seculity.achilles.html（2017 年 8 月 21 日確認）.

[21]. OpenVAS, "Open Vulnerability Assessment System", http://www.openvas.org/（2017 年 8 月 21 日確認）.

[22]. Tenable Network Security, "Nessus Vulnerability Scanner", https://www.tenable.com/products/nessus-vulnerability-scanner（2017 年 8 月 21 日確認）.

[23]. Rapid7, "Top Rated Vulnerability Management Software", https://www.rapid7.com/products/nexpose/（2017 年 8 月 21 日確認）.

[24]. Metasploit，”Penetration Testing Software”, https://www.metasploit.com/（2017 年 8 月 21 日確認）.

[25]. Core Impact, "Penetration Testing & Vulnerability

© 2017 Information Processing Society of Japan

Assessment", https://www.coresecurity.com/core-impact（2017 年 8 月 21 日確認）.

[26]. 日本特許, 富士通株式会社, "テストデータ作成方法, テストデータ作成プログラム及びテストデータ作成装置", 特開 2013-196390，2013-9-30

[27]. 日本特許, サイバーアクティブセキュリティーエルティーディー, "予測的なセキュリティ製品を提供し, 既存のセキュリティ製品を評価する方法と製品", 特表 2016-507115，2016-3-7.

[28]. Weilin Xu, Yanjun Qi, and David Evans, "Automatically Evading Classifiers", Network and Distributed System Security Symposium (NDSS), San Diego, CA 21-24 February 2016, https://www.cs.virginia.edu/~evans/pubs/ndss2016/evademl.pdf（2017 年 8 月 21 日確認）.

[29]. 八槇 博隻, "人工知能技術を用いた標的型サイバー攻撃に関する一考察", 電子情報通信学会総合大会講演論文集 2016 年_情報システム(1), "S-12"-"S-13", 2016-03-01.

[30]. TOM FAWCETT, Learning from Imbalanced Classes, https://www.svds.com/learning-imbalanced-classes/（2017 年 8 月 21 日確認）.

© 2017 Information Processing Society of Japan