Vol.2021-DPS-186 No.60
Vol.2021-CSEC-92 No.60
2021/3/16

疑似ログ生成によるアノマリ検知技術強化に関する考察

山本 匠[1]　岩崎 亜衣子[1]，小林 創[1]，西川 弘毅[1,2]
河内 清人[1]　吉村 礼子[1]

概要：実際の攻撃ログを大量に用意することが困難なことから教師無し学習によるアノマリ検知技術が注目されている．一方でアノマリ検知技術においても多少の攻撃データがあれば，利用するパラメータや特徴の選択を適切化することができ，攻撃データのバリエーションが多ければより様々な攻撃に対応した調整が可能になると考えられる．本研究では，少量の実攻撃ログの近傍にあるログを疑似的に生成し，検証データに含めパラメータの調整を行うことで，アノマリ検知システムの性能改善が可能かを確認する．また同様のアプローチで偽陽性ログの近傍にあるログを生成し，学習データや検証データに含めることで性能改善が可能かを確認する．単純なアノマリ検知システムと公開攻撃データを使って本手法の有効性を確認する．

キーワード：機械学習，アノマリ検知，検知回避，誤検知，疑似ログ，サイバー攻撃

1. はじめに

特定の企業・組織を狙った標的型攻撃が依然として深刻である[1][2]．近年でも，日本国内の政府機関や企業が標的型攻撃の対象となり被害を受けており，対策が求められている[3]．また，制御システムのネットワーク化に伴い，発電プラントやガスプラントなどの重要インフラへのサイバー攻撃が脅威となりつつあり，国家の安全保障を揺るがす重大な懸念事項となっている[4][5]．世界的に注目の集まるオリンピック・パラリンピック競技大会においては，攻撃者の恰好のターゲットとなることが予想される．大会期間中にサイバー攻撃により重要インフラが機能停止すれば大会運営に大きな支障が出る．

一方，セキュリティ監視の現場においては，専門的な知識を必要とするスタッフが不足していることが常態化してしまっているのが現状である．経済産業省からの調査報告によると，2016 年時点で 132,060 人の情報セキュリティ人材が不足しており，2020 年には 193,010 人の不足になると予想されている[6]．そのため，少ないスタッフでもサイバー攻撃を高精度かつ効率よく検知することができる技術が必要である．

サイバー攻撃を監視する技術としては，既知の攻撃手口や攻撃者のふるまいを検知するルールを使ったルールベースの検知技術が従来からよく知られている[7]．しかし昨今，攻撃の高度化や未知攻撃の増加により，あらかじめルールを定義することが困難となり，Security Operation Center（SOC）の監視要員を悩ましている．また対象システムごとルールを手作業で調整する必要がありルールベースの検知技術の限界が近づいている．そのため，あらかじめルールを定義する必要のない，もしくは，正常と異常とを識別する境界が自動的に決められる高度な検知技術が望まれる．

これを実現する技術として Artificial intelligence（AI）など機械学習を利用した攻撃検知技術が昨今注目を集めている．AI はあらかじめ用意された単数もしくは複数のクラスのデータを学習し，クラス間を切り分ける境界を自動的に見つけ出す．クラスごとのデータを大量に用意することができれば，AI は適切に境界を見つけることができる．AI をサイバー攻撃の監視に応用することができれば，これまで専門的な知識やスキルを持つスタッフが行ってきたルールの定義や更新を AI が代替してくれると期待される．

しかし，ネットワークセキュリティにおいては，AI で最も重要なクラスごとのデータを大量に用意することが困難であるという課題がある．特に攻撃に関してはその発生が稀であり，攻撃データを学習用に大量に用意することは非常に難しい．そのため，攻撃データが少ない，または，まったく無い環境においても，効果的に攻撃を異常として検知することができる技術が必要となってくる．

そのような技術の代表としてアノマリ検知技術が知られている．アノマリ検知技術では，監視対象システムの正常な通信やふるまいを正常モデルとして学習し，それから逸脱する通信やふるまいを異常として検知する．正常モデルをルールで定義するホワイトリストも広義の意味ではアノマリ検知技術と言える．

一方で，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[4]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[8]．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，通常業務や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．しかしながら，これらの攻撃も検知

1 三菱電機株式会社
Mitsubishi Electric Corporation.
2 静岡大学
Shizuoka University

できるようアノマリ検知システムの閾値を単純に調整してしまうと，正常なイベントも異常として誤検知する偽陽性の可能性が増えてしまう．

これまで述べてきた課題を解決するために，本研究では，機械学習を利用したアノマリ検知技術をベースに，少量の実攻撃ログの近傍に存在しうる攻撃ログを疑似的に生成し，検証データに含めアノマリ検知システムのパラメータ調整を行うことで，アノマリ検知システムの性能改善を図る．また同様のアプローチでアノマリ検知システムが誤検知した正常ログ（偽陽性ログ）の近傍に存在しうる正常ログを疑似的に生成し，学習データや検証データに含め，アノマリ検知システムのパラメータ調整を行うことで，アノマリ検知システムの性能改善を図る．これにより，システムの正常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作りこまれた攻撃に対して，誤検知を抑えながら追従することが期待できる．

本稿の構成は，まず 2 章にて本研究で扱う用語について説明し，3 章で関連研究について紹介し，4 章で提案手法の概要を説明する．5 章で提案手法の評価を行い，6 章で本研究をまとめる．

2. 本研究で扱う基本的な用語

表 1 にて本稿で扱う用語について，あらかじめ説明する．

表 1　用語と説明

用語	説明
モデル	機械学習を使って大量のデータを分析し得られたデータのパターンや傾向．本稿ではアノマリ検知技術において，正常データのパターンだけを学習したものを正常モデルとも呼ぶ．
ログ	Web アクセス，ファイルアクセス，認証試行などといったイベントに関する情報がテキストに逐次記録されているもの．通常，1 行につき 1 イベントの情報が構造化されて記録されるものが多い．
学習データ	機械学習アルゴリズムを利用して，データのパターンを分析しモデルを作成する際に利用するデータ．本稿において，学習データはログである．
検証データ	学習データとは異なる期間に取得したデータで，モデルのハイパーパラメータなどを調整するために用いるデータである．時系列データの場合，学習データよりも後に取得されていることが想定される．本稿において，検証データはログである．
評価データ	学習データおよび検証データとは異なる期間に取得したデータで，学習されたモデルを評価するために用いるデータである．同モデルは学習データと検証データを使ってパラメータフィッティングがされている．時系列データの場合，学習データおよび検証データよりも後に取得されていることが想定される．本稿において，評価データはログである．
ラベル付きデータ	ラベルが付与された学習データ，検証データ，テストデータ．特に学習データにラベルが付与されている場合は，教師データとも呼ばれる．攻撃検知技術の場合，例えば正常や攻撃といったラベルが付与される．本研究で扱う検証データと評価データにはラベルがついている．
教師あり学習	ラベル付きデータを用いて学習し，未知のデータがどのラベルに分類されるかを予測する手法．例えば正常か攻撃のラベルがついたデータを学習し，新たなデータが正常と攻撃のどちらかを予測する．
教師なし学習	学習データの中から何かしらの構造や法則を学習する．この学習結果を用いて，未知のデータがその構造や法則に当てはまるかを判定できる．正常データの一貫したふるまいを学習し観測データが正常データに属するか否かを判断するアノマリ検知技術も教師無し学習である．
真陰性（True Negative）	検知システムによって正常なデータが正常と判定されること．
真陽性（True Positive）	検知システムによって異常なデータが異常と判定されること．
偽陰性（False Negative）	検知システムによって異常なデータが正常と判定されること．検知漏れとも呼ぶ．
偽陽性（False Positive）	検知システムによって正常なデータが異常と判定されること．誤検知とも呼ぶ．
検知率	真に異常なデータのうち，正常と判定されたデータの割合．本稿では True Positive Rate（TPR）と表現することもある．
誤検知率	真に正常なデータのうち，異常と判定されたデータの割合．本稿では False Positive Rate（FPR）と表現することもある．

ROC カーブ	検知システムの判定閾値を変化させながら横軸に FPR，縦軸に TPR をプロットした曲線（Receiver Operating Characteristic curve）.
AUC（Area Under the Curve）	ROC カーブの下側の面積. 1.0 に近ければ近いほど 2 値分類器の性能が高い．0.5 はランダムな 2 択と同等の性能. AUC が 0.7 より大きいと，良いモデルとみなされ，0.9 を超える AUC を持つモデルは優秀とされる[9].

3. 関連研究

提案方式に関連する既存技術について簡単に紹介する．

3.1 既存技術

●文献[10]

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[11]

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[12]

文献[11]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[13]

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．

3.2 著者らの既存研究

検知を回避する攻撃ログを生成する既存の手法として，著者らは，過去に文献[14]および文献[15]を発表している．文献[14]では，セキュリティ製品の評価のために，正常な状態に良く似た特徴を持つよう作られた巧妙な攻撃のログを自動的に生成する技術を提案されている．文献[15]では，セキュリティ製品の評価のために，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検知（偽陽性）と，本来検知すべき事象を検知しない検知漏れ（偽陰性）を自動生成する技術が提案されている．両技術ともに，正常データのふるまいを学習した正常モデルの決定境界を越えるように攻撃の特徴ベクトルを変更していき，境界を越えた特徴ベクトルに対応する特徴を持つように模擬環境上で攻撃を生成する．環境や攻撃などの制約や攻撃機能の有無などを確認することでリアリティのある巧妙な攻撃ログを生成する．

これらの技術は，アノマリ検知システムが扱う特徴空間上で検知システムの決定境界をまたぐように特徴ベクトルを修正し，検知を回避するサンプルを探索する．特徴空間が非線形かつ高次元な攻撃検知技術になればなるほど，特徴空間上の表現（特徴ベクトル）から実空間の情報（ログ）に逆変換することは困難であり，逆変換できたとしても場合によってはログとして不自然なものや実際の攻撃として成立しないような疑似攻撃ログが生成されることが懸念される．

4. 提案方式

4.1 コンセプト

提案方式は，少数の実攻撃ログをもとに，アノマリ検知システムによる検知を回避しうる攻撃ログを疑似的に生成し，当該アノマリ検知システムの精度の改善を図るものである．また，アノマリ検知システムによって誤検知とされた正常ログ（偽陽性ログ）をもとに，偽陽性ログを疑似的に生成し，当該アノマリ検知システムの精度の改善を図るものでもある．

3.2 節で述べたように，特徴空間が非線形かつ高次元な攻撃検知技術になればなるほど，特徴空間上の表現（特徴ベクトル）から実空間の情報（ログ）に逆変換することは困難であり，逆変換できたとしても場合によってはログとして不自然なものや実際の攻撃として成立しないような疑似攻撃ログが生成されることが懸念される．

そこで本研究では，特徴空間上で特徴ベクトルを修正して疑似的に検知を回避する攻撃ログや偽陽性ログを探すのではなく，実空間上でログの各項目を修正した後，特徴ベクトルに変換し，特徴空間上で検知結果に変化が生じるかを確認する．このままでは場当たり的になるため，新たに

作る特徴空間上で修正対象のログ（攻撃ログもしくは偽陽性ログ）近傍の真陰性の正常ログを特定し，真陰性の正常ログに多く見られる特徴の傾向を含むように，修正対象のログを修正していく．

以降では，プロキシログを例に主に説明を行うが，提案コンセプトはプロキシログ以外のログにも適応可能である．またアノマリ検知システムが扱う入力データの最小単位を 1 サンプルと呼び，ログの 1 イベントを 1 サンプルと仮定して説明を行う．正常サンプルは正常ログに含まれる 1 イベント（レコード），攻撃サンプルは攻撃ログに含まれる 1 イベント（レコード）を表す．

4.2 疑似攻撃ログの生成手順

修正対象の攻撃サンプル近傍にある真陰性の正常サンプルを特定し，近傍の真陰性の正常サンプルに多く見られる特徴を含むよう攻撃サンプルを修正する手順を説明する．

① 回避対象のアノマリ検知システムを使い，修正対象の攻撃サンプルが正常か異常かを判定する．正常と判定された場合は，同攻撃サンプルを，検知を回避する攻撃として，登録する．異常と判定された場合は，修正対象の攻撃サンプルとして以降の手順を実施する．

② アノマリ検知システムによって正常と判定された正常サンプル（真陰性の正常サンプル）を使い，修正対象の攻撃サンプル近傍にある真陰性の正常サンプル（近傍真陰性正常サンプル）を抽出する（詳細は 4.2.1）. 真陰性の正常サンプルはあらかじめ用意しておいてもよい．

③ 抽出した近傍真陰性正常サンプルの特徴の傾向を算出する（詳細は 4.2.2）.

④ 得られた傾向をもとに，近傍真陰性正常サンプルの特徴値を多く含むように攻撃サンプルの各フィールドを修正する（詳細は 4.2.3）.

⑤ 回避対象のアノマリ検知システムを使って修正した攻撃サンプルが正常か異常かを判定する．修正した攻撃サンプルが正常と判定された場合は, 同攻撃を，検知を回避する疑似攻撃サンプルとして登録する．特徴の修正量が規定値を越えるまで④⑤を繰り返し，検知を回避する疑似攻撃サンプルを作成･収集する．

⑥ 元となる攻撃サンプルを変えながら，規定の回数①～⑤を繰り返す．

近傍の抽出, 傾向の算出, 特徴の修正について 4.2.1, 4.2.2, 4.2.3 にてそれぞれ補足する．

4.2.1 近傍抽出

近傍抽出は，X 個の攻撃サンプルおよび Y 個の真陰性正常サンプルから規定の特徴を抽出し，特徴情報を機械学習アルゴリズムで処理しやすい形式（特徴ベクトル）に変換する．Y は X に比べて十分大きな数である．プロキシログにおいては，ドメイン，メソッド，ステータスコードなどのカテゴリデータを，例えば，One-hot エンコーディングや，頻度にもとづいた数値表現[16]に変換する．数値データは，正規化をし，特徴種別間で大きさをそろえておく．攻撃サンプルの特徴ベクトルと真陰性正常サンプルの特徴ベクトルを用い，X 個の攻撃サンプルに対して近傍の K 個の真陰性正常サンプル（近傍真陰性正常サンプル）を特定する．利用する特徴や特徴表現は対象のアノマリ検知システムと異なっていても良い．実際には，X 個の攻撃サンプルのそれぞれから K’個の近傍真陰性正常ログを抽出し 1 つにまとめる（K≧K’）.

4.2.2 傾向の算出

傾向の算出は，得られた K 個の近傍真陰性正常サンプルと，X 個の真陽性の攻撃サンプルとから，規定の特徴を抽出し，特徴情報を機械学習アルゴリズムで処理しやすい形式（特徴ベクトル）に変換する．利用する特徴や特徴表現は対象のアノマリ検知システムや近傍抽出（4.2.2）と異なっていても良い．

得られた特徴ベクトルを用いて，K 個の真陰性の正常サンプルと，X 個の真陽性の攻撃サンプルとを切り分ける識別器（C）を学習する．識別器が K 個の真陰性の正常サンプルと X 個の真陽性の攻撃サンプルを分類した際の特徴の重要度（feature importance）を算出し，重要度の大きい上位 N 件の特徴を抽出する（F_1～F_N）．重要度を算出できる識別アルゴリズムの 1 つとしてランダムフォレストがある．

得られた上位 N 件の特徴に対して K 個の真陰性の正常ログにおける統計情報を取得する．カテゴリデータの場合は例えば最頻値（モード，mod_i，i=1～N），数値データの場合は例えば平均値（μ_i，i=1～N）と標準偏差（σ_i，i=1～N）を算出する．

4.2.3 特徴の修正

以下の手順で特徴 F_1～F_N のそれぞれに対して修正の候補値を算出する．

① 真陽性の攻撃サンプルの特徴 F_i に対応するフィールドから特徴 F_i の実際の値（d_i）を取得する．

② 特徴 F_i がカテゴリデータの場合，頻度にもとづいた数値表現（d_i）に変換しておく．頻度はアノマリ検知システムの機械学習のモデルを学習した際の学習データなどからあらかじめ算出しておく．また K 個の近傍真陰性正常サンプルにおける統計情報（s_i）を参照する．特徴 F_i がカテゴリデータの場合 s_i は最頻値（mod_i）である．特徴 F_i が数値データの場合 s_i は平均値（μ_i）である．

③ d_i が s_i より大きい場合，d_i が s_i に Δ_i ずつ近づくよう（小さくなるよう）に d_i を更新し，更新した値をリスト $list_i$ に追加する．d_i が s_i より大きい間繰り返す．d_i が s_i より小さい場合，d_i が s_i に Δ_i ずつ近づくよう（大きくなるよう）に d_i を更新し，

更新した値をリスト $list_i$ に追加する．d_i が s_i より小さい間繰り返す．d_i が s_i と等しい場合，何もしない．Δ_i は規定の値である．

④ 全ての F_1～F_N（i=1~N）に対して①～③を繰り返す．

修正対象の特徴 F_i（i=1~N）から作成されたリスト $list_i$（i=1~N）をもとに，特徴の修正候補の全組合せを作り，それぞれの組に対応するサンプルを生成する．リスト $list_i$（i=1~N）の長さがそれぞれ len_i（i=1~N）の場合，生成されるサンプルの種類は $N=\Pi len_i$ となる．特徴 F_i（i=1~N）に対応しないサンプルのフィールド（修正対象ではないフィールド）はオリジナルの攻撃サンプルと同じ値を保持する．

4.3 疑似偽陽性ログの生成

詳細は割愛するが 4.2 節の疑似攻撃ログの生成手順と同様のアプローチで，疑似的に偽陽性ログを生成する．4.2 との違いは，攻撃サンプルをもとにするのではなく，アノマリ検知システムによって誤検知された正常サンプル（偽陽性サンプル）をもとに疑似的に偽陽性のサンプルを生成する．修正対象の偽陽性サンプルの近傍にある真陰性の正常サンプルに多く見られる特徴を含むよう偽陽性サンプルを修正する．4.2 節のステップ⑤において，修正した偽陽性サンプルが異常と判定された場合には，同サンプルを，疑似偽陽性サンプルとして登録する．

学習データまたは検証データにおいて，誤検知を起こしやすい正常サンプルの割合を多くすることで，誤検知を起こしやすい正常サンプルの特徴を反映した学習になることが期待できる．

5. 評価

提案方式の実現性および有効性を確認するため評価実験を行った．

5.1 実験環境

表 2 に実験に利用したマシンの情報を記載する．

表 2 実験に利用したマシンの情報

OS	Ubuntu 20.04 LTS
CPU	Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz
RAM	32GB
Python	3.8.3
scikit-learn	0.23.1

5.2 実験に利用したデータ

5.2.1 正常データ

攻撃が含まれていない正常データとして従業員 1000 人規模の大規模組織で過去に収集されたプロキシログを利用した．連続する平日 3 日分を学習データとし，学習データよりも後の平日 1 日分のログを検証データ，検証データよりも後の平日 1 日分のログをテストデータとして利用する．平日 1 日分のプロキシログのイベント（レコード）件数は約 5,000,000 件である．数が多すぎるため本実験では 30,000 件にランダムサンプリングしている．便宜上，学習データのことを train_data と表記する．

5.2.2 攻撃データ

検証データおよびテストデータに含める攻撃データとして，CTU University で公開しているマルウェアの通信データ[17]を利用した．CTU データセットでは，Bot の実行ファイルや実際の通信の PCAP ファイルなどが公開されている．本実験においては，2018 年以降に取得されたと考えられる通信データの中で[a]，ログフォーマット形式（拡張子が.weblog）に変換されたデータが公開されていて，かつ，10 件以上イベントを含むデータを利用した[b].

2018 年 2 月～3 月の間に取得されたと考えられる攻撃データを検証データに，2018 年 4 月～10 月の間に取得されたと考えられる攻撃データをテストデータに混ぜ込んだ．混ぜ込む際，攻撃ログの種類ごとランダムに IP アドレスを選択し，相対的な時間間隔は維持したまま業務時間中に検証データとテストデータに挿入した[c]. なお処理時間を軽減するために，午前 10 時～11 時の間の検証データとテストデータを本実験で利用した．便宜上，ここで得られた検証データとテストデータのことをそれぞれ val_data と test_data と表記する．val_data および test_data はラベル付きデータあり，各イベント（サンプル）には，正常サンプルもしくは攻撃サンプルであることを示すラベルが付与されている．

5.3 基本モデル

4.2 節に記載の回避対象のアノマリ検知システムとして基本モデルを説明する．本研究では，提案方式の実現性と有効性を確認するために，複雑な特徴や深層学習を利用したアノマリ検知システムではなく，単純な特徴と従来の機械学習アルゴリズムを利用してアノマリ検知システムのモデルを用意した．利用した機械学習アルゴリズムは Local Outlier Factor（LOF）である．抽出した特徴は，5.2.1 に記載の正常データ（プロキシログ）に含まれるフィールドの情報のうち，例えば，ドメイン，拡張子，ステータスコード，レスポンスサイズ，アクセス間隔などの情報である．イベントごと抽出した特徴を機械学習アルゴリズムに入力できる形式に変換して特徴ベクトルとした．

LOF は Python のオープンソース機械学習ライブラリである scikit-learn で実装されているものを利用した[18].

学習データと検証データに train_data と val_data をそれぞれ利用し，グリッドサーチでハイパーパラメータを調整し

a ファイル名に付与されている日付をもとに取得日を判断した
b 大量のリクエストを出しているマルウェア（例、1 日に 10 万件以上）のデータについては本評価のデータから除外した。
c 業務時間帯のうち 8:30AM～10:30AM のランダムな時間帯に攻撃が開始するように攻撃ログを挿入した．この時間帯を選んだ理由として，業務を開始しメールを見始める時間帯と考えたためである．

て作成したモデルを基本モデルと呼ぶ．グリッドサーチで調整したハイパーパラメータは n_neighbors と cotamination である．

5.4 疑似ログの生成

本実験で用意した疑似攻撃ログと疑似偽陽性ログについて説明する．

5.4.1 疑似攻撃ログ

基本モデルによる val_data に対する誤検知率が 1%未満となるよう閾値 θ_0 を設定し，val_data および test_data から真陽性の攻撃サンプルを抽出した．さらに 4.2 節の手順に従い，真陽性の攻撃サンプルを修正し，基本モデルと閾値 θ_0 を使って，疑似攻撃サンプルを抽出し，疑似攻撃ログを用意した．便宜上 val_data と test_data から作った疑似攻撃ログをそれぞれ fn_val と fn_test と呼ぶ．fn_val および fn_test において，疑似攻撃サンプルに付与されるラベルはオリジナルの攻撃サンプルと同じである．

なお本実験では，攻撃サンプル 1 件に対し 20 件の近傍真陰性正常サンプルを抽出し（X=1，K=K'=20），当該 20 件の近傍真陰性正常サンプルの情報を使って，当該攻撃サンプルを修正した．なお 4.2 のステップ①において，オリジナルの攻撃ログ（修正する前）から正常と判定された攻撃ログについては，今回の実験では対象外とした．実運用では，検知できない攻撃は，そもそも気づくことができないため，得ることが難しいと判断したためである．

4.2.1 の近傍抽出において利用した機械学習アルゴリズムは，最近傍法で，LOF と同様 scikit-learn で実装されているものを利用した[19]．ディフォルト設定のハイパーパラメータを利用した．

4.2.2 の傾向算出において利用した機械学習アルゴリズムはランダムフォレストで，LOF と同様 scikit-learn で実装されているものを利用した[20]．ディフォルト設定のハイパーパラメータを利用した．傾向算出には特徴寄与度（feature_importance）が 0 より大きい特徴全てを利用した．

4.2.3 のステップ③における Δ_i は，修正対象の特徴が数値データのとき，4.2.2 の手順で算出した近傍真陰性正常サンプルの標準偏差（σ_i）をもとに $\Delta_i=6*\sigma_i/10$ と定義した．修正対象の特徴がカテゴリデータの場合は，$\Delta_i=1$ と定義し，特徴値を Δ_i だけ大きく（小さく）する場合，当該カテゴリにおいて当該カテゴリデータの次に頻度が高い(低い)カテゴリデータを利用した．

5.4.2 疑似誤検知ログ

4.3 節の手順に従い疑似偽陽性ログを生成した．細かな設定は 5.4.1 に記載の通りである．

基本モデルによる val_data に対する誤検知率が 1%未満となるよう閾値 θ_1 を設定し，train_data および val_data から偽陽性の正常サンプルを抽出した．さらに 4.3 節の手順に従い，偽陽性の正常サンプルを修正し，基本モデルと閾値 θ_1 を使って，疑似偽陽性サンプルを抽出し，疑似偽陽性ログを用意した．便宜上 train_data と val_data から作った疑似偽陽性ログをそれぞれ fp_train と fp_val と呼ぶ．疑似偽陽性サンプルに付与されるラベルはオリジナルの偽陽性の正常ログと同じである．

5.5 改良モデル

疑似ログを含めた新たな学習データと検証データを用いて作成した改良モデルを説明する．

5.5.1 疑似ログを利用したデータ

新たな学習データ，検証データ，テストデータについて説明する．

●疑似偽陽性ログを混ぜた学習データ

正常データのみからなる学習データ（train_data）に疑似偽陽性ログ（fp_train）を混ぜたデータを train_data_fp とする．train_data_fp における fp_train の割合が大きくなりすぎないよう fp_train のデータサイズが train_data のサイズの 10%を越える場合はランダムにサンプリングを行った．

結果として，イベント総数が 31444 件で，そのうち 1444 件が疑似偽陽性サンプルとなった．

本データを学習データに利用することで，誤検知しやすい正常データに対してより重きをおいて学習することが予想される．

●疑似偽陽性ログを混ぜた検証データ

攻撃データと正常データからなる検証データ（val_data）に疑似偽陽性ログを混ぜたデータを val_data_fp とする．val_data_fp における fp_val の割合が大きくなりすぎないよう fp_val のデータサイズが val_data のサイズの 10%を越える場合はランダムにサンプリングを行った．結果として，イベント総数が 23122 件で, そのうち 2112 件が疑似偽陽性サンプルとなった．

本データを検証データに利用することで，誤検知されやすい正常データを正常として正しく判定するようアノマリ検知システムのハイパーパラメータが調整されることが期待される．

●疑似攻撃ログを混ぜた検証データ

攻撃データと正常データからなる検証データ（val_data）に疑似攻撃ログを混ぜたデータを val_data_fn とする．val_data_fn における fn_val の割合が大きくなりすぎないよう fn_val のデータサイズが val_data のサイズの 10%を越える場合はランダムにサンプリングを行った．結果として，イベント総数が 21148 件で，そのうち 138 件が攻撃サンプルで，そのうち 22 件が疑似攻撃サンプルとなった．

本データを検証データに利用することで，正常と判定されやすい疑似攻撃サンプルを攻撃として正しく判定するようアノマリ検知システムのハイパーパラメータが調整されることが期待される．

●疑似攻撃ログを混ぜたテストデータ

攻撃データと正常データからなるテストデータ（test_data）に疑似攻撃ログを混ぜたデータを test_data_fn

Vol.2021-DPS-186 No.60
Vol.2021-CSEC-92 No.60
2021/3/16

とする. test_data_fn における fn_test の割合が大きくなりすぎないよう，fn_test のデータサイズが test_data のサイズの 10%を越える場合はランダムにサンプリングを行った．結果として，イベント総数が 23111 件で，そのうち 2131 件が攻撃サンプルで,そのうち 30 件が疑似攻撃サンプルとなった．

本データをテストデータに利用することで，正常と判定されやすい疑似攻撃サンプルを攻撃として正しく判定することができるかを評価することができる．

5.5.2 改良モデル

表 3 に記載の学習データと検証データの組合せで改良モデルを作成した．比較のために基本モデルについても記載する．

基本モデルは，疑似ログを加える前のオリジナルの学習データ（train_data）と検証データ（val_data）を使って作成されている．

改良モデル 1 は，基本モデルと同じオリジナルの検証データ（val_data）と疑似偽陽性ログ（fp_train）を加えた学習データ（train_data_fp）を使って作成されている．改良モデル 2 は，基本モデルと同じオリジナルの学習データ（train_data）と疑似偽陽性ログ（fp_val）を加えた検証データ（val_data_fp）を使って作成されている．改良モデル 3 は，基本モデルと同じオリジナルの学習データ（train_data）と疑似攻撃ログ（fn_val）を加えた検証データ（val_data_fn）を使って作成されている．

いずれのモデルにおいても，グリッドサーチによって検証データにおける精度が最大となるようハイパーパラメータを調整した．全てのモデルに対して同じテストデータを用いて精度の比較を行った．

表 3　基本方式と改良方式に利用する学習データと検証データ

	基本モデル	改良モデル 1	改良モデル 2	改良モデル 3
学習データ	train_data	train_data_fp	train_data	train_data
検証データ	val_data	val_data	val_data_fp	val_data_fn

5.1 実験結果

表 4 に基本モデルおよび改良モデル 1〜3 の実験結果を記載する．改良モデル 1 および改良モデル 3 において，基本モデルよりも高い精度となることが確認できた．

表 4　実験結果

	基本モデル	改良モデル 1	改良モデル 2	改良モデル 3
学習データ	train_data	train_data_fp	train_data	train_data
検証データ	val_data	val_data	val_data_fp	val_data_fn
テストデータ 1	test_data	test_data	test_data	test_data
AUC	0.787	0.816	0.787	0.832
テストデータ 2	test_data_fn	test_data_fn	test_data_fn	test_data_fn
AUC	0.79	0.818	0.790	0.835

5.2 考察

今回の実験では疑似偽陽性ログを検証データに含めてもその効果を確認することはできなかった（改良モデル 2）が,疑似偽陽性ログを学習データに含めた改良モデル 1 や，疑似攻撃ログを検証データに含めた改良モデル 3 においては基本モデルよりも精度改善を確認することができた．結果として，疑似的に生成した偽陽性ログや攻撃ログを混ぜ込むことで，教師無し学習のアノマリ検知システムの精度の改善が可能であることを示すことができた．試してはいないが，学習データと検証データへの疑似ログの埋め込みの組合せによっては，さらなる精度向上が期待できる．

本研究においては，真陰性の正常ログの特徴を使って疑似的に攻撃ログや偽陽性ログを生成したが，偽陽性の正常ログ（誤検知）や偽陰性の攻撃ログ（検知漏れ）の特徴の傾向を活用することも可能であると考える．例えば，偽陰性の攻撃ログには，アノマリ検知システムが攻撃ログを正常と判定してしまうような特徴が多く含まれていると考えられる．そのため，そのような特徴を修正対象の攻撃ログに含めることで，修正の効率化が期待できる．また偽陽性の正常ログには，アノマリ検知システムが正常ログを異常と判定してしまう特徴が多く含まれていると考えられる．そのため，そのような特徴を修正対象の攻撃ログには含めないようにすることで，修正の効率化が期待できる．

今回の実験においては，非常に単純なアノマリ検知システムを使ったため，回避対象のアノマリ検知システム（基本モデル）のそもそもの精度が低く，改良モデルによる精度改善がしやすい条件だったとも考えられる．今後，より複雑な特徴情報や，深層学習などのより高度な機械学習アルゴリズムを利用したアノマリ検知システムに対しても，提案方式の効果が期待できるかについて評価を行っていく．

6. おわりに

本研究では，実攻撃ログを大量に用意することが困難な

環境を想定し，アノマリ検知システムに注目した．アノマリ検知システムを導入する際に利用可能な学習データや検証データに，疑似的に生成した攻撃ログや偽陽性ログを埋め込みことで，アノマリ検知システムの精度改善が可能かを検証した．評価実験により，疑似的に生成した攻撃ログを検証データに含めパラメータ調整を行うことで，アノマリ検知システムの性能改善が可能であることを示した．また，疑似的に生成した偽陽性ログを学習データに含めモデルを学習することでも，アノマリ検知システムの性能改善が可能であることを示した．本手法は教師無し学習のアノマリ検知システムだけではなく，教師有学習を利用した攻撃検知システムの精度改善にも利用可能と考えられる．

参考文献

[1]. 警察庁：令和２年上半期におけるサイバー空間をめぐる脅威の情勢等について, https://www.npa.go.jp/publications/statistics/cybersecurity/data/R02_kami_cyber_jousei.pdf（2021 年 2 月確認）

[2]. 日本経済新聞, 標的型メール攻撃，過去最多ペース，上半期 3900 件, https://www.nikkei.com/article/DGXMZO64479980R01C20A0CR8000（2021 年 2 月確認）

[3]. IPA，情報セキュリティ白書 2020，https://www.ipa.go.jp/files/000087025.pdf（2021 年 2 月確認）

[4]. 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（2021 年 2 月確認）

[5]. IPA, 制御システムのセキュリティリスク分析ガイド補足資料 制御システム関連のサイバーインシデント事例４ ～Stuxnet：制御システムを標的とする初めてのマルウェア～, https://www.ipa.go.jp/files/000080701.pdf（2021 年 2 月確認）

[6]. 経済産業省，IT 人材の最新動向と将来推計に関する調査結果～ 報告書概要版 ～, https://tokiocyberport.tokiomarine-nichido.co.jp/cybersecurity/s/column-detail50（2021 年 2 月確認）

[7]. 三菱電機，サイバー攻撃検知技術, http://www.mitsubishielectric.co.jp/corporate/randd/spotlight/a35/index.html（2021 年 2 月確認）

[8]. 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（2021 年 2 月確認）

[9]. XLSTAT，ROC 曲線, https://www.xlstat.com/ja/solutions/features/roc-curves（2021 年 2 月確認）

[10]. 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

[11]. 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.

[12]. 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（2021 年 2 月確認）

[13]. 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.

[14]. 山本 匠，西川 弘毅，木藤 圭亮，河内 清人，検知技術回避を目的とする攻撃のシミュレート手法の提案，暗号と情報セキュリティシンポジウム(SCIS2017)

[15]. 山本 匠，西川 弘毅，木藤 圭亮，河内 清人，検知システムの高精度化に関する提案，コンピュータセキュリティシンポジウム 2017（CSS2017）

[16]. Steve T.K. Jan，et al，Throwing Darts in the Dark? Detecting Bots with Limited Data using Neural Data Augmentation, Security & Privacy 2020, https://people.cs.vt.edu/vbimal/publications/syntheticdata-sp20.pdf （2021 年 2 月確認）

[17]. Stratosphere. (2015). Stratosphere Laboratory Datasets. Retrieved March 13, 2020, from https://www.stratosphereips.org/datasets-overview（2021 年 2 月確認）

[18]. scikit-learn, sklearn.neighbors.LocalOutlierFactor, https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.LocalOutlierFactor.html（2021 年 2 月確認）

[19]. scikit-learn, sklearn.neighbors.NearestNeighbors, https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.NearestNeighbors.html（2021 年 2 月確認）

[20]. scikit-learn, sklearn.ensemble.RandomForestClassifier, https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html（2021 年 2 月確認）