Vol.2020-CSEC-91 No.21
Vol.2020-SPT-40 No.21
Vol.2020-EIP-90 No.21
2020/11/26

機械学習ベースのNIDSにおける動的な判別器生成に関する検討と予備評価

林 はるか[1,a)]　佐藤 秀哉[1]　小林 良太郎[1]

概要：IPA が発表している情報セキュリティ 10 大脅威において，組織を狙った標的型攻撃による脅威が昨年度と共に 1 位となっており，標的型攻撃が日本の組織において大きな脅威として認識されている．その背景から，近年では機械学習型の NIDS が研究されており，当研究室でも，設置組織の通信データを取得し機械学習に利用する NIDS システムを提案してきた．このシステムでは，判別器を生成するときの条件（特徴量，アルゴリズム，学習データ）は事前にある特定の組に決められている．しかし，より高い精度とより短い学習時間を達成するために必要な判別器生成条件は，設置組織内において日々変化していく．そこで本研究では，判別器生成条件を複数用意し，その中から動的に適切と考えられるものを 1 つ選択し，判別器を動的に生成する手法の検討と予備評価を行う．

キーワード：機械学習，動的生成，NIDS，組織内ネットワーク

A Study and Preliminary Evaluation of Dynamic Discriminator Generation in Machine Learning Based NIDS

Abstract: In the IPA's Top 10 Threats to Information Security, the threat of targeted attacks on organizations is Targeted attacks have been recognized as a major threat to Japanese organizations, as it was ranked first in the last year. With this background, machine learning NIDS has been studied in recent years, and we have also proposed a NIDS system that acquires communication data of installed organizations and uses them for machine learning. In this system, the conditions for generating the discriminator (features, algorithm and training data) are it is predetermined for a particular pair. However, the discriminator generation conditions required to achieve higher accuracy and shorter learning time change day by day within the installation organization. Therefore, in this study, we prepared several discriminator generation conditions, selected one of them dynamically appropriate, and we study and preliminary evaluation of a method to generate a discriminator dynamically.

Keywords: Machine learning, Dynamic generation, NIDS, Internal network

1. はじめに

IPA（情報処理推進機構）では毎年，前年に発生した社会的に影響が大きかったと考えられる情報セキュリティの事案から，その年の脅威となりうる候補を決定し情報セキュリティ 10 大脅威として発表している [1]．10 大脅威は毎年多少なりとも順位に変動があるが，組織を狙った標的型攻撃による脅威は昨年度と共に 1 位となっている．

標的型攻撃とは，とある組織の情報を窃取することが目的の攻撃で，目的を達成するために下調べを行い，多くの攻撃手法を使い分けたり執拗に攻撃してくるといった特徴がある．その背景から，近年では機械学習型の NIDS が研究されてきた [2], [3], [4]．しかし，多くの研究が公開されているデータセットを用いて行われており，そのデータセットは攻撃者も使用することが可能となっている．そこで当研究室では，設置組織の通信データを取得し機械学習に利用する NIDS システムを提案してきた [5].

このシステムでは，特徴量，アルゴリズム，学習データといった判別器を生成するときの条件は事前にある特定の組に決められている．しかし，より高い精度とより短い学

[1] 工学院大学 Kogakuin Uniersity
[a)] j117230@ns.kogakuin.ac.jp

Vol.2020-CSEC-91 No.21
Vol.2020-SPT-40 No.21
Vol.2020-EIP-90 No.21
2020/11/26

習時間を達成するために必要な判別器生成条件は，設置組織内において日々変化していく．

そこで本研究では，条件の異なる判別器を複数用意し，判定日のデータの傾向に基づいて動的に使い分ける手法の検討と予備評価を行う．また本論文での動的とは，事前に作成した判別器をデータによって使い分けることを指しており，システム内に 1 つの判別器のみを実装している従来のシステムを静的なシステムとして定義する．

設置組織における正常通信と，その組織に向けられた悪性通信にあたるデータを取得し，データに合わせた判別器で判定を行うことで，設置組織に合わせた検知が可能となる．本研究では，

A 特徴量の組み合わせによる精度の向上

B アルゴリズムの組み合わせによる精度の向上

以上 2 つの観点による判別器を複数作成し，先ほど述べた手法について検証を行う．

2 章では従来の機械学習型 NIDS について述べ，3 章では本研究の提案手法について述べる．4 章で本研究での評価方法について述べたのち，5 章で本提案手法の為に行った事前検証の概要と結果について述べ，6 章で 5 章の結果を基に作成した判別器について述べる．7 章で本提案手法の評価結果を示し，8 章では評価結果に対する考察を行い，9 章で本論文をまとめる．

2. 従来の機械学習型 NIDS

本章では，今まで当研究室で提案してきた，設置組織内で通常・悪性通信を取得し使用する機械学習型 NIDS の概要と課題について述べる．

2.1 従来システムの概要

従来の機械学習型 NIDS は，自組織内で正常通信と悪性通信を取得し，学習や判定で用いるデータセットとする．そしてこのデータセットを用いて一日単位で定期的に判別器の更新を行う．これにより，データセットの取得環境依存によるスコア低下を避けると共に，最新の攻撃環境を反映させることができる．

図 1 に機械学習型 NIDS のネットワーク図を示す．このシステムの設置組織内のネットワークは，社内ネットワーク，悪性通信収集ネットワーク，解析ネットワークの 3 つに分かれており，これらはルーターとファイヤーウォールを介して外部ネットワークにつながっている．

社内ネットワークで正常通信を取得し，悪性通信収集ネットワークにあるハニーポット等で悪性通信を取得する．また，解析ネットワークでは取得した通信データから特徴量を抽出し，機械学習にかけることで社内ネットワークに潜む悪性通信を検出する．

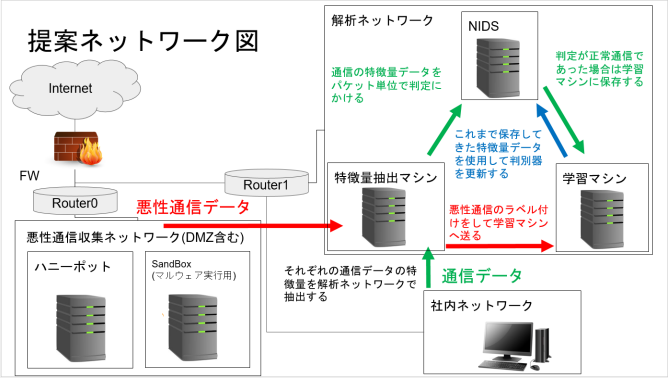

図 1 従来の機械学習型 NIDS のネットワーク図

Fig. 1 Network diagram of a traditional machine learning NIDS

2.2 従来システムの課題

2.1 節で述べたこのシステムでは，機械学習の判別器生成条件を以下に固定して検証を行っている：特徴量は bro-IDS プラグインが生成する 47 種類中，文字データを含まない 26 種類，アルゴリズムはランダムフォレスト，学習データは，判定データの前日にあたるデータ．

しかし 1 章で述べたように，本論文の背景にある標的型攻撃は様々な攻撃手法を用いて情報の窃取を試みてくる為，選択された攻撃手法によって通信データの内容が異なり，判別器生成条件が一定では高い検知精度を維持できないことが予想される．その為，判別器を複数用意しデータによって使い分けることが望ましい．

これらの課題を解決するために，本論文では条件の異なる判別器を複数生成し，動的に使い分ける機械学習プログラムを検討し，評価する．

3. 提案手法

3.1 提案手法の概要

提案手法の概要図を**図 2** に示す．この手法では，大きく分けて 2 つの工程によって動作する．1 つ目は，判別器総当たりプログラムを用いて，判別器生成条件の異なる複数の判別器から，判定データに最適と考えられる判別器を選別する工程．2 つ目は，選択された判別器を用いて学習判定を行い，設置組織に結果を出力する工程である．

まず設置組織が判定したいデータを指定したら，判別器総当たりプログラムは，指定したいデータの前日分に当たるデータを過去データから取得する．

次に，事前に用意されている複数の判別器を用いてそれぞれ検証を行い，一番精度の良かった判別器を選定する．この時の学習データは前日のデータの前半部分を使用し，後半部分を判定用のデータとして用いる．

最後に，多数決プログラムは選定された判別器を使用して学習判定を行い，結果を設置組織に向けて出力する．この時に使用するデータは，設置組織が指定したデータを判

Vol.2020-CSEC-91 No.21
Vol.2020-SPT-40 No.21
Vol.2020-EIP-90 No.21
2020/11/26

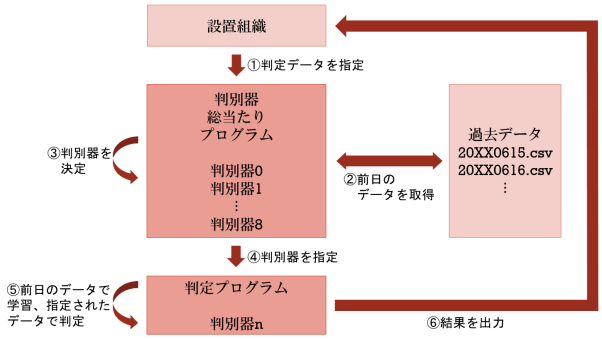

図 2　提案手法の概要
Fig. 2　Overview of the Proposed Methodology

定データ，判定データの前日 1 日分を学習データとして用いる．

なおこの提案手法で使用するプログラムでは，複数のアルゴリズムで結果を出し，その結果を多数決させることで最終的な結果を出す多数決アルゴリズム (voting) という方法を使用している．

また 1 章で述べたように，標的型攻撃の攻撃者は事前に下調べや準備を行って攻撃を行ってくる．これは機械学習を用いた準備も同様で，高精度な判別器を作成し，検知されにくい手法を用意してくることが考えられる．その為本研究では，攻撃者が使用できる公開データセットと使用できない非公開データセットの 2 種類のデータセットを使用し評価を行うことで，作成される判別器の違いを確認した．本研究で使用したデータセットは，公開データセットの例として Kyoto 2016 Dataset，非公開データセットの例として本大学で取得した Closed Dataset を使用した．

3.2　判別器の作成方法

図 2 のプログラム内で使用する判別器は，約 1 ヶ月分のデータに対して，特徴量とアルゴリズムの総当たりをそれぞれ行い，その結果の中から精度の良かった組み合わせを使用して作成したものである．

まず，1 日分のデータに対して総当たりを行った際に精度が一番高かった組み合わせを約 1 か月分集計し，それぞれどんな特徴量やアルゴリズムを使用しているのかを確認した．そして，この際に使用頻度が高いと判断されたものを判別器に使用した．

次に，いくつの特徴量やアルゴリズムを使用した際に一番精度が高い結果となったかを確認した．それらの個数の中で一番少なかった個数，一番多かった個数，中間の個数をそれぞれ判別器に使用した．

結果として本研究では，組み合わせの異なる判別器を 9 つ作成し評価を行った．また，総当たりを行った際の概要は後の 5 章にて述べる．

4.　評価方法

4.1　検証環境

本研究では正常通信の取得と本提案手法の評価を行った際の環境と，悪性通信の取得と特徴量の抽出を行った環境が異なる為，以下にそれぞれ示す．

4.1.1　正常通信の取得環境と提案手法の検証環境

本研究では，著者の使用している PC の通信データを正常通信として採取し使用した．使用した PC のスペックは CPU が intel Core i5-7300U，メモリは 8GB，OS は Windows 10 Home 64 ビットである．

通信データを取得する際に使用したソフトウェアは wireshark ver. 3.2.7 (v3.2.7-0-gfb6522d84a3a) で，pcap 形式のファイルとして取得した．

また機械学習は Anaconda の Spyder ver. 4.1.5 というソフトウェアを使用し，オフライン上で評価を行った．この評価では機械学習によく用いられている言語である Python を使用した．Python のバージョンは Python3.7.9 である．また，機械学習のライブラリはオープンソースである scikit-learn を使用した．ほかにもデータの読み込みなどでは Pandas，総当たりを行う際に複数のデータを指定するために os，総当たりそのものを行う為に itertools のライブラリも使用した．

4.1.2　悪性通信の取得環境と特徴量の抽出環境

悪性通信を採取するためのハニーポットは，正常通信を採取するネットワークとは別のネットワークを用意し，その中に設置することで採取した．

ハニーポット用の PC は CPU が intel(R) Core(TM) i7-7700，メモリは 8GB，OS が ubuntu18.04 の物を用意した．

用意したハニーポットは honeytrap である．honeytrap の設定で，11211 や 23 などの複数のポートを開放し，解放したポートと設定を変更するための ssh 用のポート以外の通信は遮断した．また，honeytrap は Docker を使用して構築している．Docker のバージョンは，Docker version 19.03.13，build 4484c46d9d である．攻撃によってはコンテナが破壊される可能性があるため，破壊されたり，何かファイルがコンテナ上にダウンロードされた際は，ダウンロードされたファイルを保存しつつ，コンテナの再構築を行った．そして tcpdump ver 4.9.3 を使用し，悪性通信を正常通信と同じ pcap 形式のファイルで取得した．

また，採取した正常通信と悪性通信の特徴量の抽出は，bro-IDS ver. 2.5.3 という，wireshark や tcpdump と同じネットワーク調査ツールのプラグインを使用して行う．この際に入力された pcap ファイルは，特徴量が抽出されると，機械学習に使用できる csv 形式のファイルとして出力される．

Vol.2020-CSEC-91 No.21
Vol.2020-SPT-40 No.21
Vol.2020-EIP-90 No.21
2020/11/26

表 1　評価指標
Table 1　Evaluation Index

		実際の結果	
		正常通信 Positive	悪性通信 Negative
判定結果	正常通信 Positive	TP (True-Positive)	FP (False-Positive)
	悪性通信 Negative	FN (False-Negative)	TN (True-Negative)

4.2　評価方法

本節では，通信データを分類した際の予測結果の精度を評価する手法について述べる．

機械学習を用いたクラス分類の精度を評価するには，混同行列を用いて正しく識別できた件数と誤って識別した件数を比較することが一般的である．混同行列を作成した際の対応表を**表 1** に，その混同行列に割り振られた件数を使用して分類精度を数値化する式を 1 と 2 に示す [6].

本研究で数値化する内容としては，正確に正常通信と悪性通信を分類できたかを確認するために全体正解率（Accuracy=ACC）として式 1 で数値化する．また，悪性通信を正常通信と判定してしまうことは避けたいため，偽陽性率（False Positive Rate=FPR）として式 2 で数値化する．

$$ACC = \frac{TP + TN}{TP + FP + TN + FN} \tag{1}$$

$$FPR = \frac{FP}{FP + TN} \tag{2}$$

5.　判別器生成条件のための事前検証

本章では，提案手法の判別器に使用する特徴量やアルゴリズムを選定するために行った総当たりの検証方法や結果について述べる．

5.1　Kyoto 2016 Dataset

Kyoto 2016 Dataset は，京都大学に設置されているハニーポットの通信データを使用して作成されたものであり，KDD Cup 1999 Data の特徴量の一部である 14 種類と，独自に追加した 10 種類の特徴量を合わせた 24 種類で作成されている [7].

本研究では数値データのみを扱う為，24 種類の特徴量のうち，文字列データを含まない 14 種類を使用する．なお，この 14 種類は判定に用いるラベルデータを含まないものである．

また Kyoto 2016 Dataset には，正常通信のほかに既知の悪性通信と未知の悪性通信という 3 種類のラベルが存在するが，本研究では Closed Dataset に合わせて正常通信と既知の悪性通信の 2 種類のみを使用する．

5.2　Closed Dataset

Closed Dataset は，本大学生の通信データと大学内に設置したハニーポットを使用して収集したデータに bro-IDS で特徴量を付けたものであり，特徴量は bro-IDS プラグインによって 47 種類が生成されている [8].

本研究では 47 種類の特徴量のうち，数値データのみの 26 種類を使用する．また，bro-IDS では正常通信と悪性通信を判定するためのラベルが存在しない為，本大学生の通信データが正常通信，ハニーポットで収集された通信データが悪性通信であると仮定した上でラベル付けを行い，Closed Dataset として付け加えた．

5.3　特徴量の検証方法と結果

5.3.1　特徴量の検証方法

どの特徴量が判別器を作成する際に重要となるかを確認するために，まずはアルゴリズムと学習データを一定に固定した状態で使用できる特徴量を総当たりさせ，4.2 節で述べた全体正解率や偽陽性率を確認した．また，総当たりさせると検証数が多く，結果の比較が困難になる為，全体正解率が 99%を越えるものだけを確認した．同様に検証時間短縮の面から，検証精度を下げると確認された特徴量は除外し，アルゴリズムは固定した．なお，この検証の際のアルゴリズムは，どちらのデータセットでも検証時間が短く，かつ精度が高めであったランダムフォレストを使用した．

Kyoto 2016 Dataset の学習データは 2007 年 1 月 1 日に固定し，判定データを 2 日に設定し検証を行った後，同様に判定データを 3 日から 30 日までに変えながら検証を行った．また，Kyoto 2016 Dataset 内で 2007 年 1 月 10 日のデータは存在しない為，10 日を除いた 28 日分を判定データとして使用した．

Closed Dataset の学習データは 2020 年 5 月 15 日に固定し，判定データを 18 日から 6 月 29 日までに変えながら検証を行った．また，Closed Dataset の正常通信は研究を行った日付にのみ採取している為，休日を除いた 28 日分のデータを判定用として使用した．この時の悪性通信は，正常通信と同日の物を使用した．

また Closed Dataset はデータ数が多く，検証に時間がかかってしまう為，データ数を Kyoto 2016 Dataset に合わせて 5 万件から 6 万件ほどに削減して使用した．

5.3.2　特徴量の検証結果

まず，Kyoto 2016 Dataset の特徴量で総当たりを行った際の結果を**図 3** に示す．

この結果は 14 種類の特徴量で総当たりを行い，その際に検証精度を下げた 3 つの特徴量を除いた 11 種類を使用したものである．

この図の横軸は検証に使用した判定データの日付，左側の縦軸は，11 種類の特徴量を総当たりさせた総数 2047 件のうち，全体正解率が 99%を越えた件数を示しており，青

Vol.2020-CSEC-91 No.21
Vol.2020-SPT-40 No.21
Vol.2020-EIP-90 No.21
2020/11/26

の棒グラフと対応している．また，右側の縦軸は全体正解率が 99%を越えなかった件数を示しおり，オレンジの折れ線グラフと対応している．

この図 3 から，全体正解率が 99%を超えた件数が全体の約 30%から 50%程度確認できる日と，全体の 10%未満しか確認出来ない日の 2 つに大きく分かれることが確認できる．

また総当たりの結果から，Duraton や Srv_serror_rate，Source_Port_Number の 3 種類は偽陽性率を下げる傾向にあり，逆に Dst_host_srv_count は偽陽性率を上げる傾向にあった，また，Destination_Port_Nimber は全体正解率を上げる傾向にあることも分かった．なお一番精度の高かった組み合わせで使用された特徴量の個数としては，3 個から 10 個の特徴量を使用した組み合わせが確認されたが，6 個の特徴量を使用した日が一番多く，8 日存在した．次いで 9 個使用した日が 5 日，5 個使用した日が 4 日という結果になった．

次に Closed Dataset の特徴量で総当たりを行った際の結果を図 **4** に示す．この結果は，26 種類の特徴量で総当たりを行い，その際に検証精度を下げた特徴量を除いた 11 種類を使用したものである．

この図 4 の，全体正解率が 99%未満の件数を示すオレンジの折れ線グラフから，Kyoto 2016 Dataset で総当たりを行った時よりも 99%を越えた精度を出しやすいことが分かる．また Kyoto 2016 Dataset の時とは異なり，全体正解率が 99%を越えた件数が 30%に達しなかった日付は 2 日しかなく，それ以外は約 40%以上の件数を確認出来た．

また総当たりの結果から，src_bytes や Kyoto 2016 Dataset には存在しない，wrong_fragment という特徴量が全体正解率を上げる傾向にあることが分かった．なお一番精度の高かった組み合わせで使用された特徴量の個数としては，3 個から 9 個まで組み合わせという Kyoto 2016 Dataset と似たような結果が確認されたが，3 個と 7 個，8 個の特徴量を使用した日が一番多く，各 6 日存在した．次いで 6 個使用した日が 4 日という結果になった．

5.4 アルゴリズムの検証方法と結果

5.4.1 アルゴリズムの検証方法

次にアルゴリズムでも総当たりを行い結果を確認した．アルゴリズムは scikit-learn の cheat-sheet[9] で公開されているものから，プログラムの実行時間やデータの種類を考慮して 8 種類を選出した．なお cheat-sheet の Ensemble Classfiers は，同じく scikit-learn で公開されているアルゴリズムから考慮した [10]．選出したアルゴリズムを**表 2** に示す．また 3.1 節で述べたように，本研究の提案手法では voting という多数決アルゴリズムを使用している為，この検証では，総当たりで指定されたアルゴリズムで多数決を取った際の結果を，その組み合わせの精度として比較した．

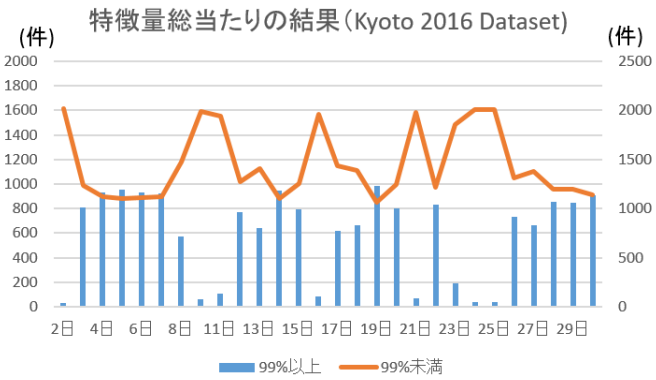

図 3 特徴量総当たりの結果 (Kyoto 2016 Dataset)
Fig. 3 Results of feature brute force(Kyoto 2016 Dataset)

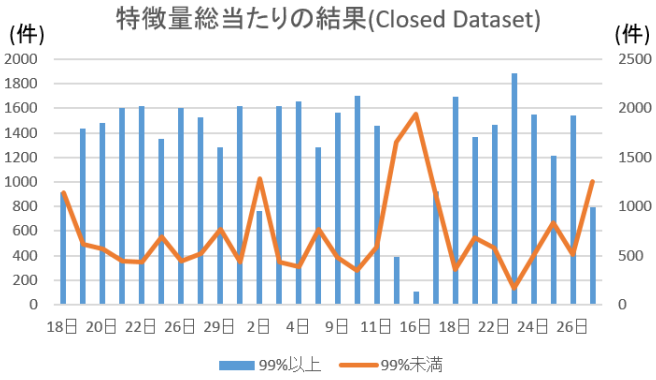

図 4 特徴量総当たりの結果 (Closed Dataset)
Fig. 4 Results of feature brute force(Closed Dataset)

表 2 アルゴリズム一覧
Table 2 List of Algorithms

多数決アルゴリズム	voting
多数決に使用するアルゴリズム	AdaBoost(ada)
	Bagging(bag)
	ExtraTree(et)
	GradientBoosting(gb)
	RandomForest(rf)
	Kneighbors(knn)
	LogisticRegression(logit)
	DecisionTree(tree)

ここでの特徴量は，Kyoto 2016 Dataset と Closed Dataset のどちらも，特徴量総当たりの際に使用した 11 種類を全て使用し，学習データと判定データは特徴量の時と同様に検証を行った．その際にも 8 種類のアルゴリズムを総当たりさせた総数 255 件のうち，全体正解率が 99%を越えるものだけを比較した．

5.4.2 アルゴリズムの検証結果

まず，Kyoto 2016 Dataset のアルゴリズムで総当たりを行った際の結果を図 **5** に示す．

この図 5 から，アルゴリズムに関しても 5.3.2 と同様に

Vol.2020-CSEC-91 No.21
Vol.2020-SPT-40 No.21
Vol.2020-EIP-90 No.21
2020/11/26

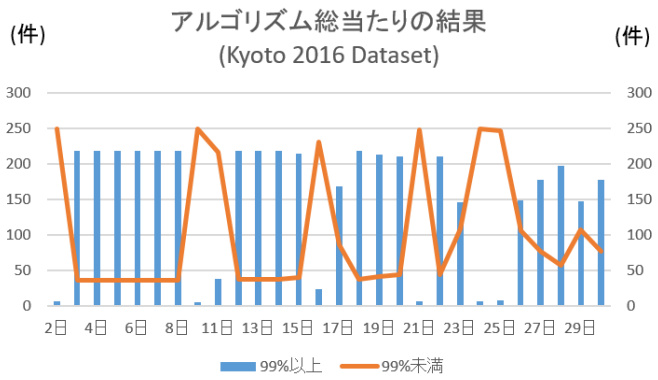

図 5 アルゴリズム総当たりの結果 (Kyoto 2016 Dataset)
Fig. 5 Results of the Algorithmic brute force (Kyoto 2016 Dataset)

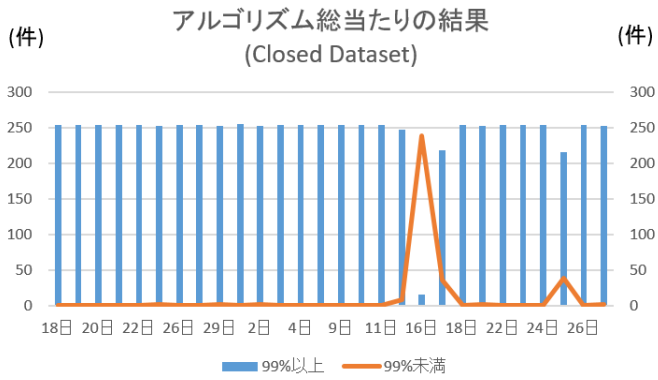

図 6 アルゴリズム総当たりの結果 (Closed Dataset)
Fig. 6 Results of the Algorithmic brute force(Closed Dataset)

全体正解率が 99%を越えた件数が全体の 60%から 85%程度確認出来る日と，全体の 15%未満しか確認できない日の 2 つに分かれた．また，その日付は 5.3.2 の日付と一致した．このことから，Kyoto 2016 Dataset は学習データと判定データの組み合わせにも精度を上下する要素があると考えられる．

また総当たりの結果から AdaBoost は全体正解率を下げる傾向にあり，ExtraTree や GradientBoosting は全体正解率を上げる傾向にあった．なお一番精度の高かった組み合わせで使用されたアルゴリズムの個数としては，1 個から 7 個までのアルゴリズムを使用した組み合わせが確認されたが，3 個のアルゴリズムを使用した日が一番多く，9 日存在した．次いで 5 個使用した日が 8 日，1 個のみ使用した日が 6 日という結果になった．

次に Closed Dataset のアルゴリズムで総当たりを行った際の結果を**図 6** に示す．この図 6 の右軸から，全体正解率が 99%未満であった件数が 1，2 件であることが分かる．同様に，Kyoto 2016 Dataset とは異なり，学習日による差が少ないと考えられる．

また総当たりの結果から，AdaBoost や ExtrTree は全体正解率を上げる傾向にあり，GradientBoosting や DecisionTree は全体正解率を下げる傾向にあった．なお一番精度の高かった組み合わせで使用されたアルゴリズムの個数としては，2 個から 5 個までの組み合わせという Kyoto 2016 Dataset の時よりも少ない結果が確認されたが，3 個のアルゴリズムを使用した日が一番多く，10 日存在した．次いで 4 個使用した日が 9 日，5 個のみ使用した日が 6 日という結果になった．

6. 提案手法で用いられる判別器

本章では**表 3** と**表 4** にて，5 章で述べた総当たりの結果を基に作成した判別器について述べる．表 3 は Kyoto 2016 Dataset の結果を基に作成した判別器，表 4 は Closed Dataset の結果を基に作成した判別器である．またこの判別器は，本研究の提案手法で使用される判別器である (図 2 参照)．

この表の横の欄には判別器の番号，縦の欄には各判別器に使用された特徴量とアルゴリズムの名前が記載されており，縦の欄の順番は 5 章の結果において使用頻度が高かった順に並んでいる．なお判別器に使用する特徴量やアルゴリズムの個数は，5 章の結果に記載した，総当たりの結果によって確認された個数のうち上位 3 種類の個数を使用した．

本研究の提案手法では，図 2 の判別器総当たりプログラムで表 3 や表 4 の判別器 9 種類をそれぞれ判定し，その中で一番精度の良かった判別器を次の判定プログラムで使用するように指定する．そして，指定された判別器を使用して判定プログラムを実行し，結果を設置組織へ出力する．

7. 提案手法の評価結果

本章では 6 章で述べた判別器を使用し，提案手法の概要に沿って評価を行った際の結果を，**図 7** と**図 8** にて示す．図 7 は Kyoto 2016 Dataset を使用した際の結果，図 8 は Closed Dataset を使用した際の結果である．

この図は，判別器総当たりプログラムによって一番精度が高いと判定されたプログラムが，多数決プログラムでは全 9 種類の判別器中何位の物であったかを示している．この図の横軸は多数決プログラムで使用した際の順位，縦軸は各順位が 28 回分の検証中で何回観測されたかを示している．またデータは，5 章で使用したデータと同一の物で検証を行った．その為データは 29 日分存在するが，本手法では学習に前日分のデータを使用するため，1 日分少ない 28 個の結果で評価を行った．

図 7 を見ると，6 位と 8 位の観測回数は少ないがほぼ同じような出現回数のように見える．しかしこの順位を 3 個ずつで上位，中位，下位に分けると，上位が約 40%，中位が約 25%，下位が約 35%と上位の判別器を選択している割合が少し高いことが分かる．

Vol.2020-CSEC-91 No.21
Vol.2020-SPT-40 No.21
Vol.2020-EIP-90 No.21
2020/11/26

表 3　判別器 (Kyoto 2016 Dataset)
Table 3　Classifier(Kyoto 2016 Dataset)

		判別器 (Kyoto 2016 Dataset)								
		0	1	2	3	4	5	6	7	8
特徴量	Destination_Port_Number	○	○	○	○	○	○	○	○	○
	Source_bytes	○	○	○	○	○	○	○	○	○
	Destination_bytes	○	○	○	○	○	○	○	○	○
	Same_srv_rate	○	○	○	○	○	○	○	○	○
	Dst_host_srv_serror_rate_rate	○	○	○	○	○	○	○	○	○
	Serror_rate				○	○	○	○	○	○
	Count							○	○	○
	Dst_host_srv_count							○	○	○
	Dst_host_serror_rate							○	○	○
アルゴリズム	GradientBoosting(gb)	○	○	○	○	○	○	○	○	○
	Bagging(bag)		○	○		○	○		○	○
	ExtraTree(et)		○	○		○	○		○	○
	RandomForest(rf)			○			○			○
	DecisionTree(tree)			○			○			○

表 4　判別器 (Closed Dataset)
Table 4　Classifier(Closed Dataset)

		判別器 (Closed Dataset)								
		0	1	2	3	4	5	6	7	8
特徴量	wrong_fragment	○	○	○	○	○	○	○	○	○
	src_bytes	○	○	○	○	○	○	○	○	○
	res_pt	○	○	○	○	○	○	○	○	○
	resp_pt				○	○	○	○	○	○
	srv_serror_rate_100				○	○	○	○	○	○
	Count_100				○	○	○	○	○	○
	srv_count_100				○	○	○	○	○	○
	serror_rate_100							○	○	○
アルゴリズム	AdaBoost(ada)	○	○	○	○	○	○	○	○	○
	ExtraTree(et)	○	○	○	○	○	○	○	○	○
	Kneighbors(knn)		○	○		○	○		○	○
	LogisticRegression(logit)			○			○			○
	RandomForest(rf)			○			○			○

また図 8 を見ると，下位の判別器よりも上位の判別器を多く選択できていることが分かる．割合としても上位が約 45%，中位が約 35%，下位が約 20%と上位の判別器を選択できていることが分かる．

なお紙面の関係上詳細な結果は記載しないが，作成した判別器は多くのデータで全体正解率が 99%以上，偽陽性率が 1%未満という結果を出した．しかし Closed Dataset では，全体正解率が 26%や 11%，偽陽性率が 10%や 38%という結果を出す場合がまれに存在した．

8. 考察

7 章で記載した結果から，今回の提案手法によって検知精度を高める判別器を動的に選択することが可能であると考えられる．

また今回は事前検証の際に，メインの Closed Dataset では学習データと判定データの組み合わせによる違いが確認されなかった為，提案手法で使用する学習データは前日の

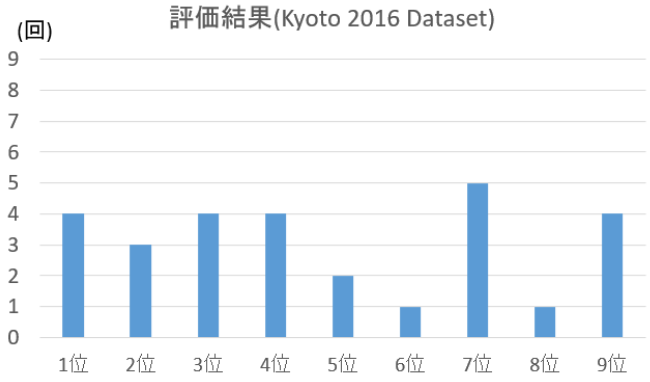

図 7　評価結果 (Kyoto 2016 Dataset)
Fig. 7　Evaluation results(Kyoto 2016 Dataset)

Vol.2020-CSEC-91 No.21
Vol.2020-SPT-40 No.21
Vol.2020-EIP-90 No.21
2020/11/26

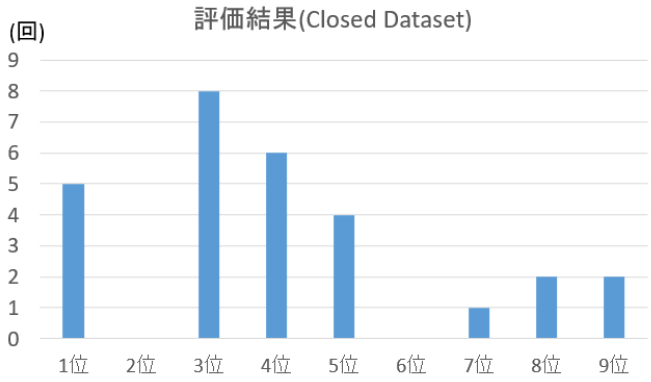

図 8 評価結果 (Closed Dataset)
Fig. 8 Evaluation results(Closed Dataset)

物と固定し評価を行った．しかし Kyoto 2016 Dataset はデータでの違いが確認され，図 7 での結果では順位による観測回数に大きな差が無かった為，Kyoto 2016 Dataset では，判別器生成条件の 1 つである学習データも判定データに合わせ，最適なものを選択する必要があると考えられる．

他にも，今回は事前検証として行った総当たりでの検証結果から使用頻度の高い特徴量やアルゴリズムの内容や個数を割り出し，提案手法で使用する判別器を作成した．しかし，Closed Dataset を使用した際に全体正解率や偽陽性率の精度が悪い結果も確認されたことから，精度の良かった特徴量やアルゴリズムだけでなく悪かったものも入れた判別器も作成しておくことで，より様々なデータに対応出来るようになることが期待される．

9. まとめ

本研究では，従来の機械学習型 NIDS において標的型攻撃によって様々な手法で攻撃された際に，判別器生成条件が固定された機械学習プログラムでは高い精度が維持できないという問題があることを提示した．

この問題を解決するため，条件の異なる判別器を複数作成し，データによって使い分ける機械学習プログラムを提案し，評価を行った．事前検証として特徴量やアルゴリズムに対する総当たりを行うことで検知精度の良いものを調べ，判別器を作成する際の基準とした．

事前検証の結果を基に判別器を作成し，本提案手法を評価することで，精度の良い判別器を選択できた割合が約 40%程度，精度の悪い判別器を選択してしまった割合が約 30%，それ以外が約 30%という結果を観測できた．この結果から，判定データに合った判別器を動的に選択することで高い検知精度を維持することが可能であると考えられるが，学習データの選択やより様々なデータに対応した判別器の作成が課題として残った．

今後は検知精度の向上と維持に加え，検証時間についても考慮しながら今述べた課題について検討していく．

謝辞 本研究の一部は，JSPS 科研費 20K11818，19K11968，19H04108 の支援により行った．

参考文献

[1] 情報セキュリティ 10 大脅威 2020：IPA 独立行政法人 情報処理推進機構，https://www.ipa.go.jp/security/vuln/10threats2020.html (参照 2020-09-04).
[2] 多田 竜之介，小林 良太朗，嶋田 創，高倉 弘喜：新 Kyoto 2006+ データセットの作成に関する検討と評価，コンピュータセキュリティシンポジウム 2016（CSS2016），2016 年 11 月.
[3] 高瀬 誉，小林良太郎，加藤雅彦，大村 廉：プロセッサ情報を用いたマルウェア検知機構における分類器のサイズ削減手法の検討，研究報告コンピュータセキュリティ，Vol.2018-CSEC-83，No.9，pp.1–8 (2018).
[4] 平野 誠， 八槇 博史： 機械学習を用いた攻撃検知に関する学習手法の精度評価，第 81 回全国大会講演論文，pp.461-462， 2019.
[5] 佐藤秀哉，林はるか，小林良太郎：組織内ネットワークにおけるハニーポットを備えた動的な機械学習ベースの NIDS の作成と予備的評価，コンピュータセキュリティシンポジウム 2020（CSS2020），2020 年 10 月.
[6] 機械学習で使う指標総まとめ (教師あり学習編)，https://www.procrasist.com/entry/ml-metrics (参照 2020-10-28).
[7] Description of Kyoto University Benchmark Data，http://www.takakura.com/Kyoto_data/BenchmarkData-Description-v5.pdf (参照 2020-10-20).
[8] tcpdump2gureKDDCup99，https://github.com/inigoperona/tcpdump2gureKDDCup99 (参照 2020-10-20).
[9] Choosing the right estimator，https://scikit-learn.org/stable/tutorial/machine_learning_map/ (参照 2020-10-05).
[10] Ensemble methods，https://scikit-learn.org/stable/modules/ensemble.html (参照 2020-10-28).