第８部

特集8 Network Muscle Learning

NMLプロジェクトチーム

第１章　はじめに

Network Mustle Learningプロジェクトは機械学習、深層学習技術(AI技術)をネットワークログデータに活用することを目的として2017年に開始した。主な応用先として、ログデータの異常検出、異常予測、またインシデント報告対応のためのスマートな支援技術を念頭に置いている。

本プロジェクトは国立研究開発法人科学技術振興機構(JST)の戦略的創造研究推進事業CREST研究領域「イノベーション創発に資する人工知能基盤技術の創出と統合化」内で「サイバー脅威ビッグデータの解析によるリアルタイム攻撃検知と予測」として採択され、およそ3年に渡る研究活動を進めてきた。活動は東京大学を中心に、東京工業大学、IIJイノベーションインスティテュート、奈良先端科学技術大学院大学の研究員で進められ、ここに外部有識者を加えた形で取り組んだ。今回、プロジェクト終了にあたりこれまでの成果を共有する。

第２章　成果の概要

本章では本プロジェクトの成果を簡単にまとめる。より詳細な内容は続く章を参照して欲しい。

本研究の目的は、AI技術を用いて個人の知識や経験に左右されないサイバーセキュリティ対策のアシストを実現することである。現在のセキュリティ対策は、セキュリティの専門家による知識と経験に依存している。すなわち、優れたセキュリティ専門家のいない組織はセキュリティ対策がおろそかになりがちであり、セキュリティ事故が発生した場合にも、対応が後手になり被害が拡大しがちである。そこで本研究では、図1に示すサイバーセキュリティ対策フローにAI技術を適用し、セキュリティ担当者のアシストを行う手法とシステムを確立することを目指した。

主な成果として、(1)サイバーセキュリティ脅威の検知に関するデータセット前処理の手法と機械学習アルゴリズムへの適用手法の開発、(2)データセットの蓄積・解析基盤Hayabusaの開発と実運用、(3)インシデント対応の自動アシストに向けたインシデントレスポンス情報の正規化と自然言語処理の適用、があげられる。これらの成果に関して、論文としての発表、オープンソースとしての公開を行なった。

2.1　各組織ごとの成果概要

東京大学サイバー脅威を検知するためのデータセットの収集、およびそのシステムの設計と開発を行った。また、収集したデータセットに機械学習を適用することで、複数種類の攻撃を検知するための手法を開発した。これにより、機械学習を用いたサイバー脅威の検知が可能であることを実証した。

さらに、インシデントレスポンスの対策アシストに関し

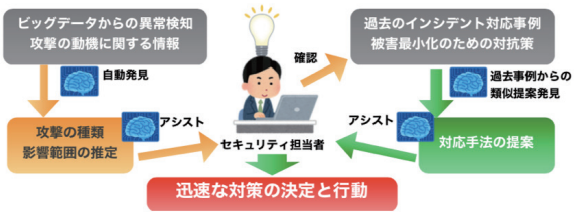

図1　AI技術を利用したサイバーセキュリティ対策のアシスト

て、自然言語処理を用いた類似事例の抽出に取り組んだ。これらの成果は、本研究成果の中核をなすものである。

東京工業大学機械学習・深層学習技術を用いてインシデントレスポンスのアシストを行うにあたり必要となる、インシデントレスポンスデータセットの調査、及びデータフォーマットの正規化、また自然言語処理を利用した類似事例の抽出に関して取り組んだ。

IIJイノベーションインスティテュートサイバー脅威検知に必要となるデータセットの調査と定義、およびそれらデータセットのリアルタイム収集と分析に必要となる解析基盤Hayabusaの開発を行った。

奈良先端科学技術大学院大学既存のデータセットや本プロジェクトで収集したネットワークデータセットを用いたサイバー脅威検知手法の開発に取り組んだ。特に、データセットの画像化に関する基本アイディアと、提案手法の評価に関して取り組んだ。

2.2 成果の要約

本研究成果は以下の4つの点で今後のセキュリティ対応技術分野に貢献したと考える。

サイバーセキュリティデータセットのAI技術への適用手法　まだデータセットの処理方法とアルゴリズムへの適用手法が確立していないサイバーセキュリティ脅威検知へのAI技術適用手法を提案した。データセットの前処理の手法や既存アルゴリズムへの適用手法を発案し、実データを用いた検証を行った結果を論文として公開した。

インシデントレスポンスのデータセットおよびフローの正規化　今までは属人的な経験と知識に基づいて行われてきたインシデントレスポンスの処理フローを、AIによるアシストに適応すべくデータセットとフローの正規化を行った。また、正規化したデータセットを用いて自然言語処理を行うための基礎段階の手法を提案した。

インシデントレスポンスにおける「対策アシスト」を目指した手法　サイバーセキュリティ脅威に対抗するにあたり、「対策のアシスト」を前提としたサポートシステムを構築した。これはサイバーセキュリティ分野において、個々の脅威の検知ではなく対応策のアシストを行うという意味において独自性のある研究であり、製品化の可能性を持つ研究である。

実データを用いたサイバーセキュリティ脅威の検知
公開されているテストデータのみならず、大学の環境やInterop Tokyo 2018, 2019における実ネットワークにおけるデータを用いてAI技術を用いたリアルタイム検知の実証実験、及び評価を行っているため、実環境に適用しやすい技術開発となっている。

以後の章では、これまでの研究成果のより詳細な内容を報告する。

第３章　サイバー脅威ビッグデータのストリーミングデータ解析基盤

本章では、大量に発生するセキュリティログデータのストリーミングデータを取り扱うための研究開発成果について述べる。本項目ではサイバーセキュリティの分析に必要となるデータセットを高速に収集かつ蓄積し、検索することで分析のパイプラインに流すことのできるソフトウェアを設計、実装に取り組んだ。また、本ソフトウェアを用いて長期的なデータセット収集と蓄積を行った。

データセットの収集と蓄積は、サイバー脅威の検知に不可欠である。脅威検知と分析を的確に行うために収集すべきデータセットを検討すると、通信ログ、DNSの名前解決記録、アクセスしたウェブサイトのログ、IDSからの検知通知、DHCPの認証ログ等が考えられる。例えば、東京大学のネットワークを例にとって考えた場合、通信ログだけでも平常時の日中で秒間約20,000メッセージ程度の流量となる。それに加え、ウェブサイトへのアクセス記録やセキュリティ機器によって判定されたアプリケーション記録などを入れると、平常時においても秒間約50,000メッセージをリアルタイムに蓄積し、かつ解析が行えるビッグデータ解析基盤が必要となる。そこで本研究では、日々大量に発生するネットワークデータを適切

に処理することを目的とした、ストリーミングデータ処理による即応性の高い応答システムを設計した。

3.1 Hayabusaの設計と実装

データ蓄積と解析のために、本研究では独自の解析基盤を設計し構築した。このリアルタイムデータ蓄積・解析システムをHayabusaのと名付けた。Hayabusaの設計要件は次の通りである。

- 非構造化データを蓄積できること
- データのストリーミング処理(リアルタイム処理)ができること
- CPUのマルチコアを利用して検索・解析処理を並列に行えること
- 蓄積されたデータをファイルとして操作しやすいこと
- オンプレミスのシステムで簡易に動かせること
- 複雑な処理系を利用せずシンプルなアーキテクチャであること

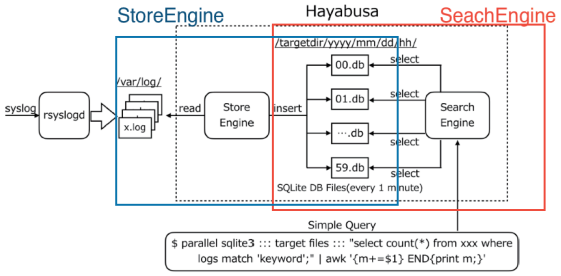

図2 Hayabusaのシステムアーキテクチャ

これら要件を満たすために、図2に示すシステムアーキテクチャを設計、実装した[67][68][69]。

3.2 分散Hayabusaによるクラウド展開

前述のHayabusaは、主にスタンドアロンシステムとしての設計と実装であった。しかし、多種多量のデータを長期間蓄積するにあたっては、Hayabusaのアーキテクチャを利用したとしても、スタンドアロンでは検索・解析性能が頭打つ可能性がある。そこで、Hayabusaを分散処理システムへと進化させスケールアウトさせる環境を作成し、分散処理基盤として性能を高める設計を行った。一方で、分散システム的な煩雑さではない方法を考え、シンプルでかつ性能がでる設計と実装を考慮した。実行する検索処理をRemote Proceduer Call (RPC)として

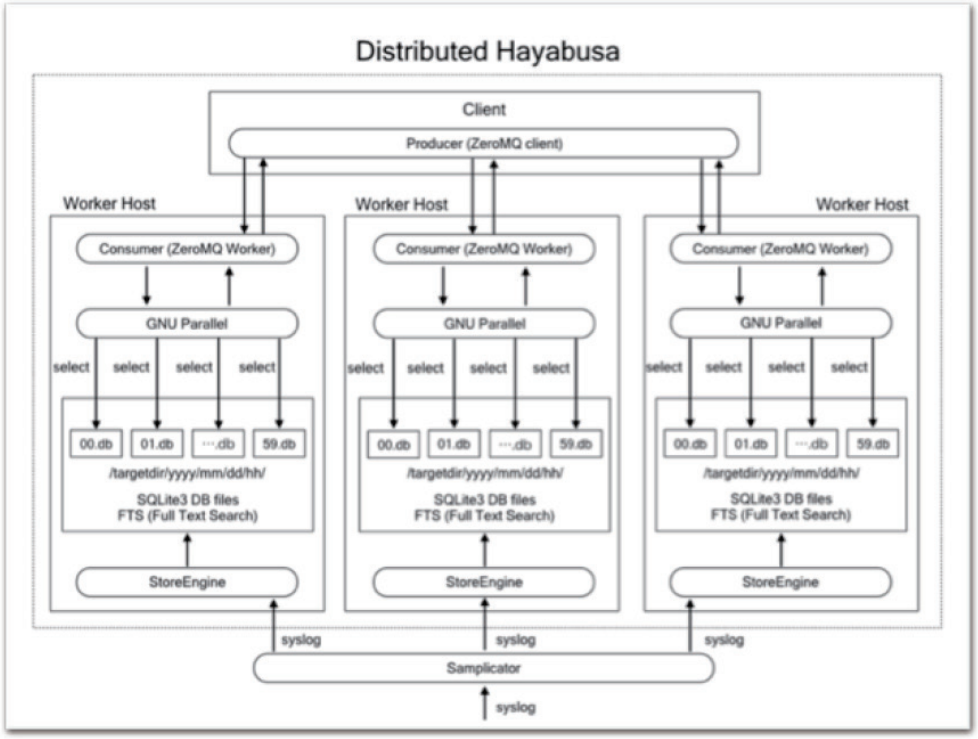

図3 分散Hayabusaのシステムアーキテクチャ

Hayabusaに送る手法を考案し、評価した。分散Hayabusaのアーキテクチャを図3に示す。

実装についてはZeroMQのPush/Pullパターンにより検索処理を行った。リクエストを投入する部分、結果を集約する部分をZero MQクライアントとワーカに担わすことにより、Hayabusaと同等の処理が可能となる。これにより、オンプレミスの資源とクラウド上に展開した資源を組み合わせてシステムを構築することが可能となった。

3.3 Hayabusaの性能評価

分散Hayabusaの性能を評価するにあたり、1日分のデータ(1440ファイル、1ファイルあたりのレコード数は10万件を想定)に対し、キーワード検索を実行した。検索は全レコードに対する文字列検索であり、マッチしたものをカウントするという単純な処理を行った結果をクライアントで集計した。ワーカを1台から10台にスケールアウトを行うと、249秒かかっていた処理が6.8秒となった。この計測結果を図4に示す。Amazon Elastic MapReduceと結果を比べたところ、分散Hayabusaの方が17倍高速に動作する結果となった。

3.4 成果の展開

本実装は、OSS (Open Source Software) としてGitHubのリポジトリにて公開した*1。公開したOSS実装では、Ansibleを用いたプロビジョニング、なGentelellaを用いたWebUIの提供、さらにはZab-bixを用いたシステム全体の監視と、システムログの収集機構を備えたパッケージとして分散Hayabusaを手軽に利用することができる。また、さらなるユーザビリティの向上のための改良を予定している。これにより多数のユーザに利用してもらうことが可能となり、かつ不具合の修正や機能向上のための改修などはコミュニティベースで行うことができ、必然的にソフトウェアとして成熟していくことが期待できる。

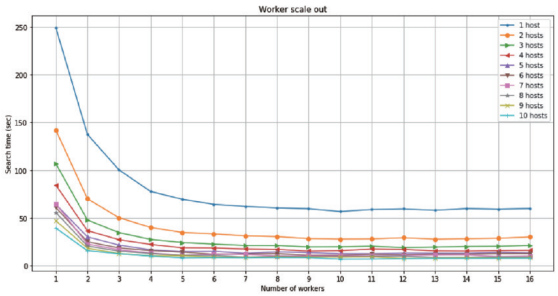

図4 分散Hayabusaの性能評価

第4章 知識ベースを用いたサイバー脅威予測手法の確立

本課題では、収集ならびに蓄積した多種多量のデータセットから単一のデータセットおよび複数のデータセットの組み合わせを用いて、AI技術の学習アルゴリズムを適用することで自動的に特徴を抽出し、攻撃者の攻撃挙動を検知することが可能であるかを検証した。また、機械学習に適用するまえのデータ前処理をエッジ側のスイッチにて行う機構の設計と実装を行った。

4.1 SYN Packetの画像化によるCNNを用いた悪性ホスト検知

現在の機械学習や深層学習のアルゴリズムは、その多くが画像認識や音声認識に利用されるために考案されている。すなわち、ネットワークトラフィックの代表的な特徴量である、5タプルなどの意味を持った情報を扱うためにどのようなアルゴリズムを用いればよいか、その知見は多く存在しない。そこで、ネットワークトラフィックの情報を従来のような5タプルの数値情報として扱うのではなく、通信状態を画像化することによって解析すれば従来のアルゴリズムがうまく適用できるのではと考えた。そこで、SYNパケットを画像化することに取り組んだ。本提案では、本手法を「Picturization」と呼称する。提案手法の概要は次の通りである。ネットワーク中からSYNパケットを採取し、送信元IPアドレスごとに保存する。送信元IPアドレスごとに、一定数のSYNパケットの各フィールドの値をつなげて正規化し、二次元の画像(SYN画像)を生成する。この画像をCNNアルゴリズムを用いて学習することで、未知のホストが送信した一連のSYNパケットからそのホストが良性であるか悪性であるかを判定する。図5に、学習に用いたSYN画像の一部を示す。

各SYN画像は、一台のホストから送信された100個のSYN

*1 https://github.com/hirolovesbeer/hayabusa2

パケットから生成される。各列がひとつのSYNパケットを示しており、各行は、上からタイムスタンプ、送信元ポート番号、宛先ポート番号、シーケンス番号、ウィンドウサイズ、となっている。図5から、例えば悪性ホストはSYNパケットを一定の時間間隔で送信していること、送信元ポート番号が一定の周期で同じ値を繰り返していることが見て取れる。一方良性ホストのSYNパケットから生成した画像では、ユーザの挙動に応じてTCPコネクションを確立しようとするため、SYNパケットの送信間隔にばらつきが見られる。また送信元ポート番号にも悪性ホストのような規則性は見られない。これらの画像を生成するためのデータセットとして、/24のダークネットに3-way handshakeのみを行うハニーポットを設置し、24時間で62,659個のIPアドレスから1,971,972個のSYNパケットを収集した。これを悪性のSYN画像として、悪性のホストのいないユーザネットワークで収集したSYNパケットから良性のSYN画像を生成、畳み込みニューラルネットワークを用いてSYN画像の分類を行なった。図6に、学習結果を示す。

学習では、良性と悪性の両方のSYN画像から、ランダムに選択した半分のSYN画像を学習に、学習に用いなかった残りの半分を用いて精度の判定に用いた。グラフ中trainingは学習時の精度を、testingは学習に用いなかったSYN画像を判定させた際の精度である。本グラフの示すように、畳み込みニューラルネットワークによるSYN画像の分類は98%を越える高い精度で実現できた。また、汎用化性能を調べるため、本実験環境で作成したダークネットではなく、NICTの用意しているダークネットデータセットを用いて、79,498個のIPアドレスか1,000,000パケットを利用し、7698個のSYN画像を作成した。ダークネットにSYNパケットを送信してくるホストはほぼ悪性ホストであるため、このデータセットを用いて作成したSYN画像は悪性と判定されるべきである。先ほどのデータセットから学習したニューラルネットに、このNICTのデータセットから生成したSYN画像を判定させた結果を図7に示す。この判定の結果、全体の86%のSYN画像について、50%以上の確率で悪性と判断できた。また、全体の51%のSYN画像は、99%の確率で悪性と判断できた[70]。

図5 SYN画像のサンプル

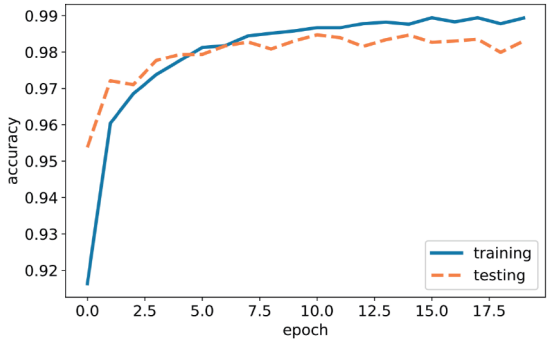

図6 SYN画像の学習結果

4.2 Bag of Bytesを用いたURL分類の試み

この研究では、フィッシングサイトというサイバー脅威の識別技術を考える。APWGによれば、2016年には120万以上の攻撃が観測されている。フィッシング対策技術は様々な方式があるが、ホワイトリスト・ブラックリストをすべてを網羅することは難しい。そこで、URL文字列だけによる解析を試みた。URLのホスト名、パス名からそのURLが悪性かどうかの判定を試みる。機械学習や深層学習を用いるには何らかのベクトル化作業が必要だが、これをヘキサコードに対する変換を試みる。この際

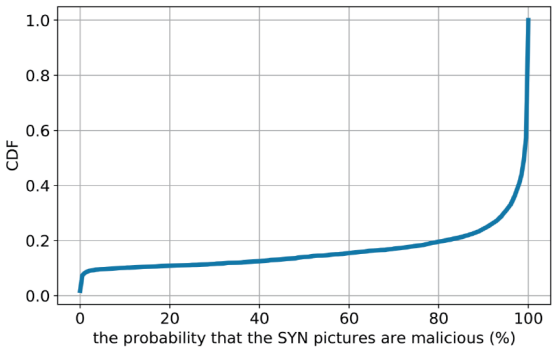

図7 NICTのダークネットで観測されたSYNパケットによるSYN画像の分類結果

に用いた技術がBag of Bytesである。具体的には、図8に示す手法においてURLのベクトル化を行った。

データセットにはPhishTank.comの提供するフィッシングサイト、正規サイトからデータを収集し、この内容をディープニューラルネットを使って学習を試みた。先行研究にはeXpose[71]という論文があり、この論文に説明される方式を実装し、提案方式との比較評価を行った。

なお、eXposeの研究論文では検知制度は99%となっているが、データセットの違いから、我々のデータセットでは90.52%の検知率であった。一方で、我々の検知精度は94%であった。また、4月に学習したニューラルネットの学習結果を、5月に採取したデータを用いた場合も、95%の精度で学習することができた[72]。

4.3 DNSプロトコルのリアルタイム解析

本研究では、近年高速化、高度化するネットワークデバイスが有する、DPDKという機能を用いて、リアルタイム判定とその対応を実現することを試みた。具体的には図9に示す通り、DPDKで単純な転送を行うプログラムに機械学習処理(scikit-learn)を追加した。受信はDPDK、パース及び特徴量抽出は自作、分類はscikit-learnという組み立てである。

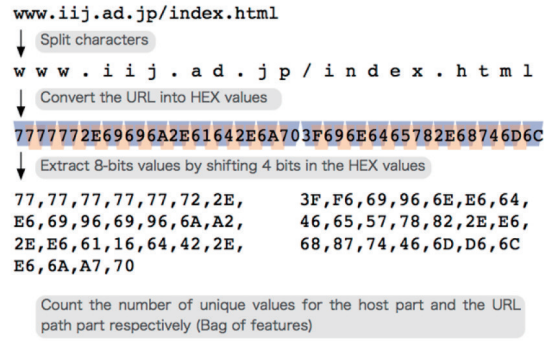

図8 Bag of Bytesを用いたURLのベクトル化

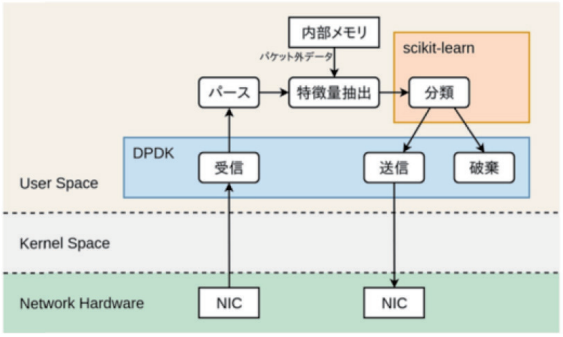

図9 DPDKへの機械学習処理の組み込み

プロトタイプとして、DNSのクエリパケットから問い合わせドメインを抽出し、DGAによって生成されたものかどうかを判定する実装を作成した。なお、DGAによって生成されたドメイン名と正当なドメイン名は、あらかじめラベル付けした学習データを与えた学習済みの判定機を利用する。リアルタイムにて行うのは、DNSの問い合わせパケットからドメイン名を抽出し、判定機を用いて判定した結果を用いて送信か破棄かのアクションを決定する部分となる。時間の計測は、プログラム中においてclock get time関数を用いた。

このプロトタイプ環境においてdigコマンドを100回発行し、分類、パース、受信、特徴量抽出、送信もしくは破棄に必要となる時間を計測した。結果として、図10に示す通り、本プロトタイプ実装においてDNSの問い合わせ1パケットの処理にかかる時間は平均して50μ秒であり、その中でも判定機を用いた分類にほとんどの時間が取られていることがわかった。処理性能としては、本プロトタイプ実装において機械学習の判定機を利用しない場合には2.3Mppsの速度が出るが、判定機を利用した場合には、20Kppsになることがわかった。このままではリアルタイム性に影響を及ぼすため、さらなる高速化を目指した改良を続ける[73]。

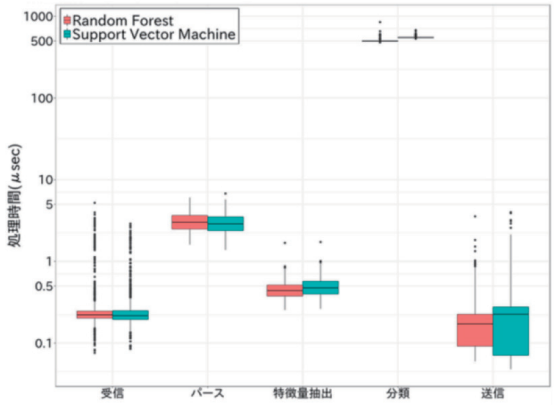

図10 DNS問い合わせパケットの判定にかかる処理時間内訳

4.4 Robust Autoencoderを用いた教師なし学習による検出手法

ネットワーク通信のデータを用いて深層学習を行うにあたっては、大きな問題点が存在する。それは、良性通信と悪性通信を明確に区別した教師データセットが十分に存在しないことである。前述のCIC-IDS2017といったデータセットは存在するが、単一の環境において短時間に観測されたデータであり普遍性に乏しく、含まれている攻撃のパターンも限られたものになっている。一方で、長期間のトラフィックデータを提供しているMAWIのようなプロジェクトも存在するが、ユーザが利用している通信のデータセットであり、良性と悪性のラベル付は行われていない。一般的に、現在公開されている通信トラフィックのデータセットは、ラベル付が行われていないデータセットがほとんどである。より多くの環境や期間、複数種類の攻撃を含んだデータセットを利用しようとすると、そのデータセットの分量は膨大なものとなるため、人間の手によるラベル付けは現実的ではない。そこで、ネットワークトラフィックに基づく侵入検知に深層学習を適用する際には、教師付き手法の代わりに教師なし手法を採用することが必要となる。しかし、サイバー脅威の検知に機械学習もしくは深層学習を適用したこれまでの研究では、教師なし学習手法を適用している研究はほとんど存在しない。これは、画像処理などの分野と異なり、トラフィックデータを用いたサイバー脅威の検知に教師なし学習を適用するための知見がほとんど存在していないため、その手法が不明確であるためである。本研究開発では、レプリケータニューラルネットワーク、すなわちオートエンコーダを用いたフローベースの検出方法をサイバー脅威検知に適用することを提案した。具体的には、画像検知に対して用いられるRobustAutoencoderという手法を用いて、悪性通信と良性通信が混ざるデータセットから良性の通信と悪性の通信を分離することを試みた。Robust Autoencoderを用いたトラフィックデータ学習の概要を図11に示す。

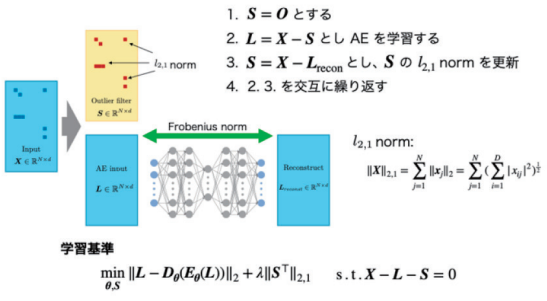

図11 Robust Autoencoderを用いたトラフィックデータの学習

Autoencoderにoutlier filter Sを適用することで、教師なしでトレーニングデータを良性の特徴と異常値のある特徴に分割し、良性のデータのみから低次元表現を学習させた。これにより、良性通信と悪性通信が混在したデータから良性通信のみを分離して学習させることを可能とした。

本提案手法を用いて、実際のMAWIデータを用いて学習を行った。さらに、MAWIデータに対してポートスキャニングという疑似攻撃を挿入したトラフィックデータを生成し、評価に用いて検証を行った。その評価結果を図12に示す。

MAWIデータセットに対して挿入したポートスキャニング疑似攻撃を、Robust Autoencoderを用いて学習させたニューラルネットワークによって判定できていることが確認できた。明確に悪性通信と良性通信がラベル付けされたデータセットがほとんど存在していないため、教師なし学習によるサイバー攻撃の検知はサイバー脅威の検知分野において非常に重要な手法となる。

本手法の研究はまだ初期段階であり、引き続き研究開発を継続する[74]。

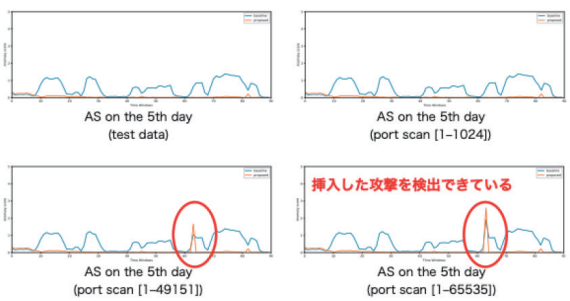

図12 評価データセットによる提案手法の評価

4.5 DNSサーバーマトリクスによる踏み台サーバーの分類

DNS（ドメインネームシステム）は、インターネットの最も重要な技術の1つである。このサービスなしには、インターネットは現在のように広く普及しなかったとさえ言える。DNSメッセージは通常、UDPパケットの上に構築される。TCPと異なり、UDPパケットの送信元アドレスは簡単に偽造できる。そのため、偽の送信元アドレスを使用して、偽のDNS要求メッセージをDNSサーバーに簡単に送信できる。プロトコル上、DNSサーバーはあらゆるドメイン名解決要求に応答できる。クライアントノードからの要求メッセージを制限またはフィルタリングするためのプロトコル上の制限はない。　本提案は、DNSメッセージを監視することにより、DNSサーバーがリフレクターとして使用されているかどうかを分類する方法を提案する。具体的には、SYNパケットの画像化で用いられた「Picturization」を利用している。DNSサーバーから送信された一連のDNSパケットを収集し、DNSサーバーの特徴マトリクスを作成する。リフレクターとして悪用されているDNSサーバーには通常のDNSサーバーとは異なるマトリクスパターンがある可能性がある。予備実験においては、テストデータとトレーニングデータが同じ日内にある場合、0.9を超えるF1スコアでリフレクターを分類できるこが判明した。図13に、正しく運用されているDNSサーバーと、リフレクターとして利用されている可能性のあるDNSサーバーの特徴マトリクスの図を示す[75]。

4.6 通信フロー Picturizationによる類似ホストの分類

本提案手法では、Picturizationを通信フローに適用し、通信を構成するフローの挙動に基づいてホストを分類する。本手法は3つの段階に別れる。まずは、ホスト（IPアドレス）ごとに送受信された一連のフローを時系列に基づいて画像化する。続いてこの画像を分割して小さな画像を生成しする。この小さな画像を一連のフローを構成する「単語」と解釈することで、文書解析に用いられるアルゴリズムを応用し、最初の一連のフローによる画像を分類する。フローの画像化は、SYN画像と同様に行われる。図14に、あるユーザネットワークで観測された一日分のフローから生成したフロー画像の一部を示す。図中7つのフロー画像は、それぞれ異なるホスト（IPアドレス）から送受信された一連のフローを示している。図14ではフロー画像は各行が1つのフローを示し、各列は左からタイムスタンプ、宛先IPアドレス、宛先ポート番号、送信元ポート番号、プロトコル、そしてフローが始まってから終了するまでの時間、の各値を0から1で正規化したものである。本手法は、このフロー画像を解析・分類することで、ネットワーク上のふるまいが類似したホストを発見する。

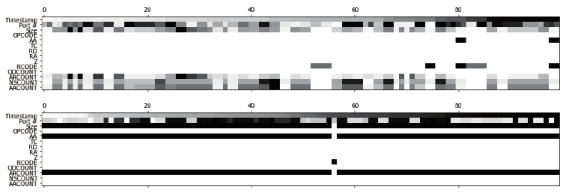

図13 正しく運用されているDNSサーバー（上）とリフレクターの疑いのあるDNSサーバー（下）の特徴マトリクス

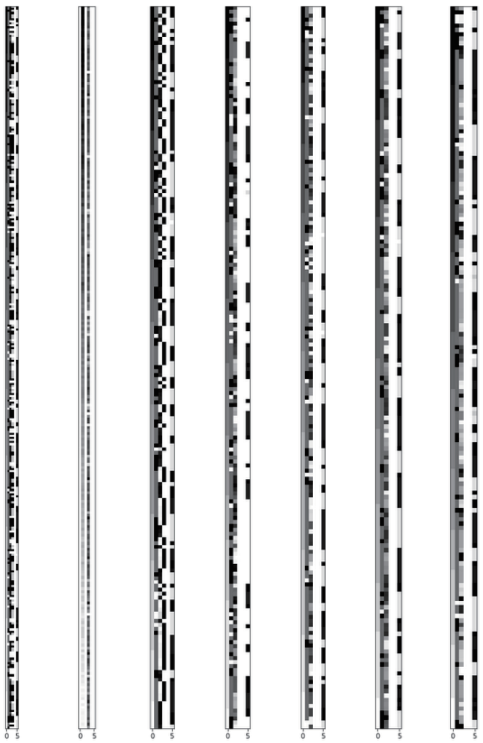

図14 通信フロー画像化の例

ネットワーク上のふるまいが類似したホストを発見するために、フロー画像を類似したもの同士に分類する。そのために本手法では、まずフロー画像を細かい画像に分類し、それらの画像を「単語」として扱うことで文書解析のアルゴリズムをフロー画像に適用する。図15にフロー画像から10個のフローごとに取り出した小さなフロー画像を示す。この小さなフロー画像を「フローワード」と呼ぶ。フローワードを単語として扱うなら、一連のフローから生成されたフロー画像とは、連続する複数のフローワードから成る文章として捉えることができる。

フローワードは、そのホストのネットワーク上のふるまいのごく一部を表現しており、あるフローワードと似たフローワードは、類似した他のホストでも観測できると考えられる。また、似たようなフローワードを多く含むホスト同士は、ネットワーク上で似たようなふるまいをするホストである言える。この仮説に基づいて、フローワードに基づくフロー画像の分類を試みた。まず全てのフロー画像から切り出したフローワードをK-meansを用いて分類し、分類したクラスタを単語として、文章の特徴を単語の頻出頻度から評価するTF-IDF（Term Frequency, InverseDocument Frequency）を用いて各フロー画像の特徴量を算出する。そして、この特徴量からフロー画像を類似したもの同士に分類する。図16に、上記の手法で類似していると判断されたフロー画像の例を示す。

図15 フローワードの生成例

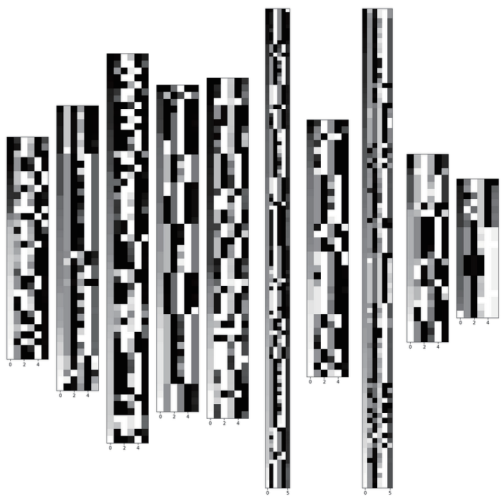

図16 類似しているフロー画像クラスタの例A（左）およびB（右）

各図から、フロワードとTF-IDFによる分類手法は、概ね類似したフロー画像をクラスタリングでクラスタリングできていると言える。Aのクラスタに属すホストを調べたところ、10個のフロー画像のIPアドレスは全てエンドユーザのものであった。また、Bのクラスタに示す6個のフロー画像は、全て日本国外のIPアドレスのものであった。これらの結果から、本手法は、ネットワークから得られたフローの情報だけを用いて、ネットワーク上で類似したふるまいをするホストをある程度分類できていると言える。本手法は、現在の段階ではまだフローの画像化と、フローワードを用いて文書解析のアルゴリズムをネットワークトラフィックの解析に応用することの可能性を試行している段階である。今後、各種パラメータの調整や様々なデータセットでの解析等を行い、本手法の有効性をより深く検証していく予定である。

4.7 早期警戒システムの設計

インターネット上で行われるサイバー攻撃のもっとも初歩的な攻撃のひとつは、脆弱性を持つホストを発見し、その脆弱性をついて制御を奪うことである。そのため攻撃者は、インターネット上のアドレス空間を定常的にスキャンしている。また攻撃を受け制御を奪われたホストやマルウェアに感染したホストは、さらに感染を広めるため、同一LAN内や、インターネットに向けてスキャンを開始する。初歩的なセキュリティ対策のひとつは、ネットワークを流れる大量のトラフィックからこうした攻撃を識別し、その攻撃を行っているホストを特定することである。これは、外部からの攻撃を防ぐとともに、自組織のネットワーク内部の感染したホストを発見し、迅速に隔離するためにも重要である。

本提案手法では、こうした一般的なスキャンのような攻撃と、攻撃を実行しているホストを特定することを目的とした。本手法では、インターネット上のホストが通信

する上でほぼ確実に送信するSYNパケットの送信の仕方に着目し、マルウェアなどに感染していないホスト（良性ホスト）と、マルウェアなどに感染したホスト（悪性ホスト）を識別した。現在のインターネットにおける通信の多くは、TransmissionControl Protocol（TCP）である。TCPは、通信を行うホスト間のコネクションの制御やパケットの再送などを担うプロトコルであり、様々なアプリケーションがTCPを用いて通信している。これはスキャンなどを行う悪性ホストも同様である。悪性ホストはTCP通信を対象のホストに試みる。我々はこの、悪性ホストもTCPを用いるという点に着目し、悪性ホストと良性のホストでTCP通信の行い方に差があるのではないか、という仮説を立てた。この仮説に基づいて、TCPで通信を行う際に必ずはじめに送信するTCP SYNパケットを集めることによって、そのSYNパケットを送信しているホストが良性か悪性かを判断する手法を開発した。

さらに、基礎となる手法の開発に引き続き、提案手法を実際に利用可能なシステムとするための研究開発も行った。本手法には、システムとして運用する上で下記にあげる3つの利点がある。(i)データ収集の容易さ:本手法で解析の対象とするのはTCPのSYNパケットのみである。ひとつのSYNパケットのサイズはTCPヘッダ全体で20バイト、ホストを識別するためのIPアドレスを加えても、ひとつひとつのデータサイズは極めて小さい。そのため、実際に取得すべきデータ量を小さく抑えることができる。またデータの保存にも大規模なストレージを必要としない。(ii)プライバシーへの配慮:本手法は、パケットのうち、IPヘッダとTCPヘッダ部のみを用いる。そのため、ユーザデータを含むペイロードまで解析する必要がない。(iii)アプリケーション非依存:本手法はホストのSYNパケットの送信の仕方にのみ着目している。どのようなマルウェアやアプリケーションがSYNパケットを送信しているかには関知しない。そのため、今後未知のマルウェアが出現しても、そのマルウェアのSYNパケット送信の振る舞いを学習することによって、同じ手法で悪性ホストの検知が可能である。

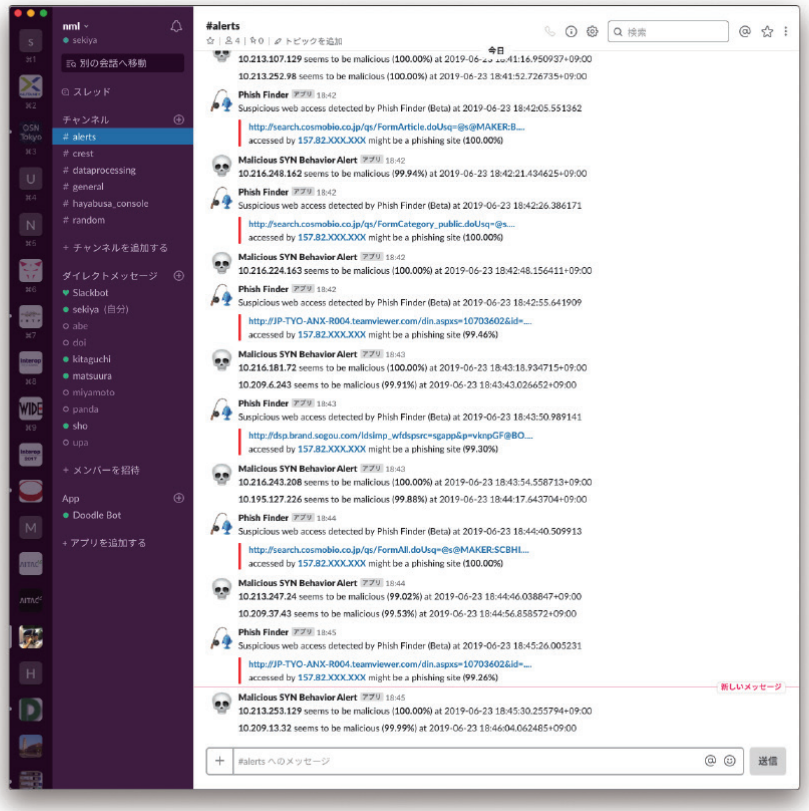

図17 Slackへのアラート通知による早期警戒の例

本提案システムでは、研究開発したサイバー脅威検知手法やサイバー脅威対策のアシスト手法等に関して、図17に示すようにSlackを利用したユーザインタフェースを開発している。

今後は、実装した解析と通知システムの性能評価と、セキュリティ対策のアシストを行うシステムへの統合を目指した開発を進める。SYN画像は、5x100ピクセルの比較的小さいデータを機械学習による分類の対象としている。しかし、ネットワークは100Gbps、400Gbpsと広帯域化しており、そうした高速なネットワークにおいてどれほどの頻度でSYN画像が生成されるかは、実際に計測を行わなければわからない。そうした計測から、実装したアプリケーションが現実的な速度で解析と通知を行えるか、評価が必要である。また、これらの事前処理を前述のDPDK手法を用いてエッジスイッチにて分散処理することも検討している。

第５章　インシデントレスポンスの自動化基盤とその要素技術

本章では、セキュリティ担当者がインシデント対応を行うにあたって、人工知能技術を用いて適切なアドバイスを与えるために必要となる、インシデントレスポンスのワークフロー定義と分類、またそのデータセット作成に関する調査と検討結果について述べる。

5.1　インシデントレスポンス自動化基盤の設計・実装

本研究では、CSIRT（Computer Security Incident Response Team）におけるインシデント対応の自動化を目指している。2018年度は特に東工大CERT（Computer Emergency Response Team）におけるインシデント対応の現状の解析と、対応において収集、管理が必要な情報の正規化を行った。以下にインシデント対応における情報の正規化手順や情報の正規化と自動化の関係について記す。更に、インシデントの管理、対応の自動化基盤の設計、実装の取り組みについて述べる。

近年の高度化された標的型メール攻撃やシステムの脆弱性を突いたサイバー攻撃等に対し、組織においてはリスクのより高いセキュリティインシデントが発生し、CSIRTによるインシデント対応に掛かるコストは増大の一途をたどっている。現在ではCSIRTを持つ大学は珍しくないが、一般企業におけるCSIRTと比較すると、大学においてはその体制や人員構成、規定やポリシーに大きな差がある。従って、インシデント対応の手順・構成要素においても大きな違いが見られる。大学におけるCSIRT、或いは実施するインシデント対応を成熟させるためには、他組織のノウハウを直接的に取り入れることは出来ず、自組織における対応の高度化等を図る必要がある。

上記の状況を鑑み、まず始めに東工大CERTにおけるインシデント対応フローを整理し、一般的な対応モデルとの比較を行った。図18において、典型的な対応事例であるマルウェア感染によるインシデントを基に、その対応フローを示す。

インシデント対応の自動化とは、対応の成熟された手順書、或いは様々なイベントに対する行動・判断のルール集といったものを活用し、対応フローを正規化して自動的・機械的に対応出来ることを目指すものとして捉えることができる。このためには、自組織におけるインシデント対応の特性を把握し、対応フローの基本構成要素を整理する必要がある。東工大CERTにおいて対応を行った事例を分析し、対応中の各イベント・行動に対して、何がトリガーに、或いは、判断基準になっていたか、また、その際に分析・収集すべき必要な情報はどの様なものであるか

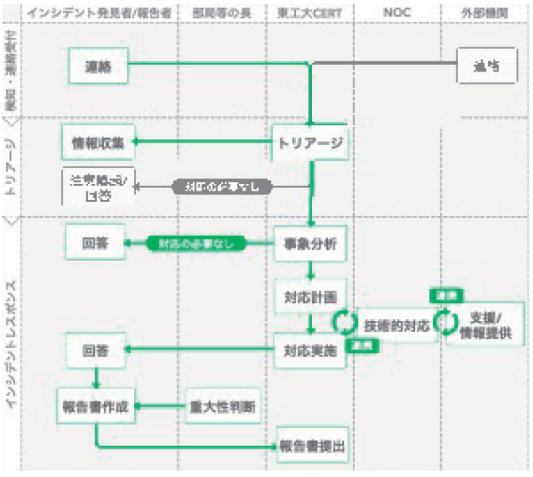

図18　Slackへのアラート通知による早期警戒の例

を対応フローの各フェーズにおいて整理した[76]。

5.2　自然言語処理によるインシデント対応の自動化

今日では、組織内にインシデント対応チーム(CERT)を置き、外部および内部からのセキュリティインシデントの対処を行うことが一般的になっている。しかしながら、一般的に経験のあるセキュリティ技術者の確保が難しいことと、人員に余裕がある組織が少ないことから他業務との兼任となる場合が多く、中小企業や、非IT系企業でのCERTによるインシデント対応は不十分になりやすい傾向がある。また、組織の規模が大きい場合においては、下記のような問題が発生してくる。

・セキュリティインシデント件数の増加
・部署特有の事情による単一的なセキュリティポリシー適用の困難

これらの問題から、インシデント対応において少数の担当者で組織全体を対応することが難しくなるため、大きな組織では部署ごとにCERTを設け、部署内で発生したインシデントをそれぞれの部署CERTが対応する、という運用をおこなう場合が多い。しかし、部署によってはセキュリティに明るい人員を確保できなかったり、部署レベルでは人員に余裕がない場合があるため、部署CERTは先述の中小企業ないし非IT系企業におけるCERTと同じような問題が発生しうる。本研究では、以上のような現状を踏まえて、人員的・スキル的に不十分なCERTであっても、発生したセキュリティインシデントに対して適切な対応が実施できるようにアシストをおこなうシステムを構築することを最終的な目的とし、その要素技術の研究・開発をおこなった。

一般的なインシデント処理のフローは、

1. 内部調査による検出や外部からの通報によるインシデントの検知
2. 検知内容の検討
3. 対象機器の隔離や解析などの処置
4. 報告といった形になる。

以上のうち、2)では1)により内部・外部から寄せられた通知について、その内容を把握しリスク・脅威の判断をおこない、3)でそれらの対象に、隔離や防護策、分析などをおこなわなければならない。そのため、特に2)および3)の部分において、セキュリティの知識が必要とされる。外部からの通知はメールによるものがほとんどであり、特に定まった形式に従っているものではない。また、通報する相手は国内組織に限ったものではなく、場合によっては日本語以外の言語であったり、通報がセキュリティ情報を扱う組織を経由した場合には複数の言語を含む場合がある。加えて、通報メールに報告対象となるスパムメールの本文が引用や添付という形で添えられたり、マルウェアの検体などがバイナリで添付されることもあり、経験や知識の少ない担当者でこれらの情報をハンドリングすることは非常に困難である。本研究では、以上の問題を解決するため、メールでのインシデント通報を自然言語処理技術により内容の解析をおこない、メールに含まれている情報を適切の構造化するとともに、通報内容から考えられる脅威および推奨する処置について担当者に提示するシステムを構築した。

本システムの概観は図19の通りであり、メール構造解析、インシデント関連情報取得、インシデント情報解析の３つのモジュールからなる。

寄せられたインシデントメールはメール構造解析モジュールに送られる。メール構造解析モジュールは、まずメールのヘッダ部分やmultipartの展開、MIMEによる添付ファイルの分離をおこなったのち、本文を文字コードや、引用符などの情報でいくつかのブロックに分割する。その後、分割した各ブロックを言語コーパスを利用したクラスタリングアルゴリズムで分類をおこなう。このとき、前後に連続して同じ分類のブロックが現れた場

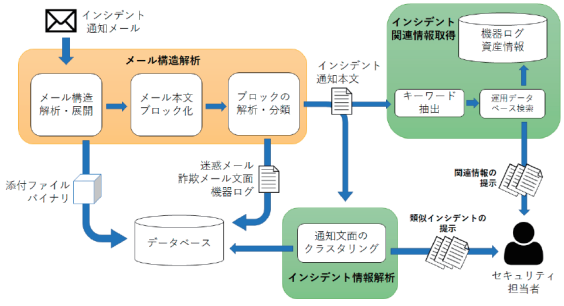

図19　インシデントレスポンス自然言語解析システム

合、それらを結合して新たなブロックとして再定義をおこなう。それぞれのブロックについて、インシデント通知本文、引用文、インシデントとして報告される対象の迷惑メールやフィッシングメールの本文、通信ログなどの機器ログなどに分類し、それぞれのブロックにタグを付与する。インシデント通知本文以外のブロックについては、あとから参照するため、それぞれ付与されたタグとともにデータベースに保存する。

インシデント通知本文のタグが付いたブロックについては、インシデント関連情報取得モジュールと、インシデント情報解析モジュールにそれぞれ送られる。インシデント関連情報モジュールでは、インシデント通知本文からIPアドレス、ドメイン名、時刻、メールアドレス、URLなどの文字列を正規表現ベースで抽出する。その後、抜き出した情報を元に自組織の機器ログや資産情報から関連情報として取得し、セキュリティ担当者に対して提示をおこなう。インシデント情報解析モジュールでは、インシデント通知本文をGuided-LDAを用いたクラスタリングによりインシデント種別の分類をおこなう。また、インシデント通知本文に対してParagraph Vectorを用いた文章特徴ベクトル化をおこない、今回のインシデントの通知本文に類似した通知本文を持つ、過去に発生したインシデントを検索しセキュリティ担当者に提示する。過去に発生したインシデントは過去に実施した対処方法とともに提示されるため、セキュリティ担当者はそれらの過去の記録を参考にインシデント対応をおこなうことができる。以下に、それぞれのモジュールについて詳細を述べる。

メール構造解析モジュール

メール構造解析モジュールでは、外部・内部からのインシデント通知メールの自然言語処理によるクラスタリングをおこなう。まず、メールファイルをInternet Message Formatに従いメールヘッダの構造をパースし、ヘッダ情報を保存ののち分離する。その後、MIMEデコーディングを行い添付ファイルなども同様に分離する。このとき、テキストファイルないしメールファイルが添付されていた場合、その本文をテキストとして展開する。メールのフォーマットに従い本文以外の部分を除去したのち、本文を一行ずつ抽出しそれぞれの行に対して言語の分類をおこなう。言語分類は行内における文字コードの出現回数を集計し、同様に日本語例文および英例文の文字コードの出現回数を用いて学習させた単純ベイズ分類器により、日本語文および英語文の判定をおこなう。言語の分類後に、言語別に英語文はそのまま、日本語文は単語の形態素解析をおこなった後に単語の抽出をおこない行内の単語の出現頻度を計算する。日本語文の形態素解析にはMeCabソフトウェアを使い、形態素解析の辞書としてはipadic-NEologdを利用した。それぞれの行の内容によって分類をおこなうため、分類器の学習用にタグ付けをおこなったメール文面やログから単語を抽出しTFIDFコーパスを作成する。このコーパスを用いてLDAモデルを学習して行の内容種別に関する分類器を作成し、それを用いて行内容の分類をおこなった。

内容の分類は下記の通りである。分類器の学習は既存のメールデータおよびログデータのデータを用いておこなった。

- 日本語のインシデント通知本文
- 英語のインシデント通知本文
- 機器ログ
- 通信ログ
- メールヘッダ（前処理で除去できなかった引用メール・転送メールのもの）

行ごとの分類にしたがって、同じ分類の行を連結してブロックを構成する。このうち、ログについては別途タグ付けをしてデータベースに保存する。インシデント通知本文のブロックは後述するインシデント関連情報取得モジュールとインシデント情報解析モジュールに送り、処理を続行する。

インシデント関連情報取得モジュール

インシデント関連情報取得モジュールでは、インシデント通知本文からインシデント処理に利用するキーワードを抜き出し、そのキーワードをもとに機器ログや資産管理台帳から関連情報を取得する。キーワードは正規表現によって抽出を行う。抽出するキーワードの種類は下記のとおりである。

・IP アドレス(v4/v6)
・ドメイン名
・時刻情報(YY/MM/DDないしHH:MM:SS形式のみ)
・メールアドレス
・URL

これらの抽出した情報をキーとして、ログなどを蓄積したデータベースから全文検索によって関連する情報を表示する。データベースは本プロジェクトで開発したhayabusaエンジンを用いて高速な全文検索をおこなう。

インシデント情報解析モジュール

インシデント情報解析モジュールでは、インシデント通知本文をもとに、深層学習を利用した類似インシデント事例の表示をおこなう。過去に発生したインシデントについて、通知文と対応を記録し、これらの過去のインシデントをメール構造解析モジュールと同じく、日本語通知文の場合はMecabを用いて形態素解析をしたのちにSkip-Gramモデルを利用して文中で用いられている単語の特徴ベクトルを計算し、保存しておく。ここで、今回発生したインシデントについても同様の処理をおこない、それぞれの文章における単語の特徴ベクトルの平均を取り、そのCos類似度を計算する。そののち、過去のインシデントの中から類似度が近いものを3つピックアップし、その概要と過去に実施した対処方法について表示を行う。本システムは、スキル的に不十分なセキュリティ担当者であっても、発生したセキュリティインシデントに対して適切な対応が実施できるように、インシデント通知情報の構造化および保存と、インシデント通知に含まれたドメイン名やIPアドレスの情報をもとにした機器や通信ログの検索と表示、および通知されたインシデントに類似した過去のインシデント事例の表示をおこなう。インシデント通知の窓口となっているメールアドレスに寄せられるメールを本システムに供給することで、これらの分析が自動的におこなわれ、セキュリティ担当に対して適切な処置を促すことが可能となる。

本システムを効果的に運用するためには、特に過去のインシデント情報の蓄積とその学習が必要となる。また、新しいタイプの脅威は常に増加するため、それらインシデント情報の蓄積はリアルタイムにおこなわれることが望ましい。しかしながら、インシデントは学習に有意となるほどの量を1つの組織で集めることは難しく、かつインシデント情報は組織にとっての機微情報を含んでいるために、組織をまたいだ情報共有を行うことは困難である。そのため今後の課題として、インシデント情報について本システムで実施している情報抽出を応用して機微な情報の匿名化を自動的に行ったり、インシデント情報のうち、具体的な処置方法に関する情報以外は特徴のみベクトル化をしたのちに共有するなどの仕組みを検討する。

第6章　リアルデータによる実証実験

6.1　実験環境の要件定義

本章では、本課題で研究開発しているサイバー脅威ビッグデータを利用した攻撃検知、予測、対応技術を、現実のシステムに応用し、その有効性を検証するために構築する実験環境の要件定義について述べる。実験環境は研究の進捗に応じて柔軟に対応できる必要がある。本研究課題では、対象組織の規模として20,000人程度の利用者が存在する中規模組織を想定しているが、実験環境としては規模の小さい箱庭的な環境から徐々に規模を拡大し、

表1　実証実験要件項目

要件項目	内容
機能的要件	研究開発した技術を実装するための必要十分な機能を提供できるかどうか
性能的要件	実装した技術が検証活動に必要な処理速度、規模で動作するかどうか
柔軟性要件	段階的に研究開発される異常検知技術、予測技術、連携技術など、実証実験のステージに応じて基盤を拡張していくことができるかどうか
適応性要件	実証実験基盤の変更や拡張が研究開発活動に支障をきたさない範囲で迅速に対応できるかどうか

最終的には目的とする規模に近い実験環境を構築し、実証実験のための事前検証として利用できることを条件とする。

実証実験を実施するにあたり考慮すべき要件を表1に示す。

機能的要件に関しては、今回の課題に対応するための実証実験として、最終的に構築を検討しているサービスを実装、検証するための十分な機能が提供されているかどうかが重要となる。本課題では、サイバー脅威ビッグデータを解析し、深層学習などの技術を用いてその活動を検知・予測することが目標である。この目標を達成するために必要となる機能項目を以下の通り定義した。

1. サイバー脅威ビッグデータ取り扱い
   (a) 収集
   (b) 蓄積
   (c) 検索
2. ビッグデータ解析
   (a) 機械学習・深層学習のトレーニング基盤
   (b) 機械学習・深層学習の運用基盤
3. メッセージ基盤
4. ユーザーインターフェース基盤

サイバー脅威ビッグデータに関しては、その収集、蓄積、および検索の運用が想定されている。本研究開発が想定しているネットワーク規模、20,000人規模の組織を運用するためのネットワーク環境としており、実証実験環境としてその規模の組織で生成されるデータ量を取り扱える必要がある。サイバー脅威ビッグデータにはさまざまな種類が想定されているが、頻度および量ともに最大のものとして、ネットワーク機器が出力するログメッセージが挙げられる。昨年度の研究成果として、高速ログ収集・検索システムとしてHayabusaを開発した。Hayabusaでは、140,000人規模のイベントネットワーク(実際の想定イベントは、日本最大級のネットワーク機器展示イベントであるInterop Tokyo)におけるネットワーク機器からのログ出力に耐えうる設計(最大秒間20,000メッセージ)を想定しており、本課題が目指している規模を十分に網羅できる。

また、ビッグデータを処理する場合、データ量の増加に伴う規模性の拡張が重要となる。実証実験環境においても、蓄積されるデータは日々増加していくと考えられ、増大するデータ量に応じて柔軟かつ透過的に規模拡張が実現できる必要がある。需要に応じて必要となる資源を確保するための手法として、現在ではオンプレミスクラウド環境や、インターネット上のクラウドサービスが利用されることが多い。本研究開発の成果に関しても、クラウド上にてサービスの展開が可能であり、クラウド技術を活用した柔軟な拡張性を持つことが重要である。なお、一般的に取り扱うデータ量が大きくなると、インターネット上のクラウドサービスを利用するよりも、オンプレミスでのクラウド環境の方が運用オーバヘッドを考慮したとしてもコスト的に有利になる場合が多い。実証実験では、ローカルとクラウドの環境を相互に接続できるようなハイブリッド環境も想定しつつ、いずれの構成でもサービスが成立することを示す必要がある。

続いてビッグデータ解析のための要件を定義する。機械学習・深層学習においては、判断モデルを調整・学習する段階と、それを運用する段階に分けて考える必要がある。調整および学習には、一般的には大量のデータと、計算機資源が必要であるが、一方で運用する場合は学習段階ほどの計算機資源は必要のないことが多い。今回の場合も、学習モデルを作成する環境とそれを運用する環境は異なると想定される。実証実験においては、運用の実証が主目的と考えられるため、ビッグデータ解析のための要件としては運用に必要な機能、性能などが満たされていることが必要となる。本課題では、機械学習・深層学習を活用したデータ解析により、セキュリティインシデント対応者に的確な情報を与えることが目標である。よって、蓄積されたサイバー脅威ビッグデータを必要に応じて取り出し、事前に学習されたモデルを適用してその結果を示すことが求められる。ビッグデータの取り出しに関しては、昨年度の研究開発の過程で開発してきたHayabusaの性能が基準となる。Hayabusaはログ解析活動時に頻繁に利用される、時間軸を限定した検索に特化したデータ構造を持つことで、一般的なデータベース検索で負荷の高い期間指定のデータ検索を高速化している。Hayabusaでは、144億メッセージの検索を約7秒で実行できることが実証されている。

コストを許容できるのであれば、すべての機能をクラウド環境に含めることも可能である。ただし、現時点において全てのデータをクラウド環境にて運用することは、各組織におけるデータ取扱のコンプライアンス(秘匿性等)の問題が生じる可能性もあるため、パブリッククラウド環境のみでサービスを構築する手法については、今回は検討の範囲外とした。

6.2 Interop TokyoにおけるPoC展示と技術アピール

本研究開発の技術と開発成果を一般にアピールするために展示とデモンストレーションを行った、InteropTokyo 2018でのPoC展示について報告する。本展示会への参加目的は次の2つである。

1. 研究開発した技術を広く一般に周知させる
2. 研究開発した技術をInterop Tokyoの大規模ネットワークの一部に組み込み、実際のネットワーク運用に利用できることを示す

2018年のInterop Tokyoでは、500を超える組織が1700個近い展示ブースを設置して各社の製品を紹介した。来場者数は公式発表で14万3千人となっている。本プロジェクトは、プロジェクト代表組織の東京大学として参加した我々の技術を展示会運用ネットワークに組み込むにあたり、展示会ネットワークの性質になじみやすいものとして(a)Syn-Picture (Picturization)、(b)Phish Finder、(c)Deep DNS Filterの検知を導入した。なお、バックエンドにはHayabusaを用いている。図20に、Syn-Picture、Phish Finder、Deep DNS Filterの機材および展示ブースのデモ画面イメージを示す。

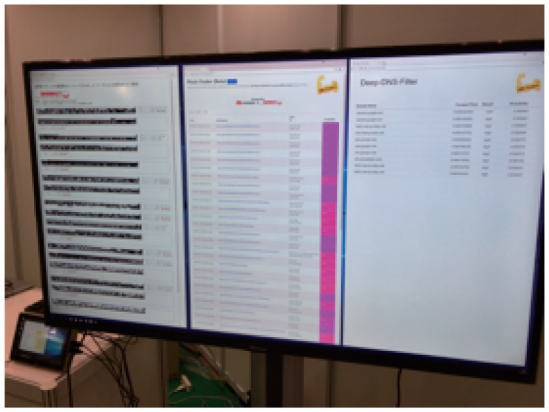

図20 Syn-Picture、Phish Finder、Deep DNS Filterのデモ画面

展示ブースでは、展示会ネットワークに組み込んだ研究開発技術の紹介、デモ画面の説明に加え、本プロジェクトの全体説明なども実施した。展示会が開催された3日間で230名を超える訪問を受け、我々の研究開発課題をネットワークサービスに興味のある一般の方々へ広く周知・広報することができた。

6.3 サイバー脅威オープンデータの公開

サイバー脅威オープンデータの公開に関しては、東京大学内においてデータ公開のポリシー制定に関する文章作成を開始し、法的なチェックを開始した。法的なチェックの後に、学内の承認権限を持った会議にかけられ、議論と承認を経てオープンデータの公開を開始する。当初はテストケースとして、学内の研究者に限定し、NDAを結んでもらうことにより公開する。この場合、個人情報の特定につながるような情報は匿名化して公開する。匿名化の基準は、個人の特定ができないレベルへの匿名化である。通信におけるIPアドレスとポート番号、およびどのデバイスがどのIPアドレスを利用したかといったDHCPの記録をつきあわせることにより個人の通信先が特定でき、さらにポート番号によって利用しているアプリケーションが特定される。これら情報をつきあわせた場合にも、個人が特定できないようにする匿名化手法を検討している。

テストケースによる公開を経て、国内外の研究者に対して、NDAを結ぶことにより一部データの公開を行うことを最終的な目標とする。また、この際にIDS等の既存セキュリティ機器、および本研究にて開発した攻撃検知手法により攻撃と検知された通信に対してラベルを提供することも検討している。

さらに、ソーシャルデータセットに関しては、サイバーセキュリティ脅威の検知に特化した辞書セットの公開を目指す。また、インシデントレスポンス事例やWeb記事の事例、対応策の事例などにより学習した判定機を共有することにより、より適切なアシストを行う判定機の作成を促す。

第７章　本プロジェクトの成果の公開

本プロジェクトの成果はhttps://www.nml.ai/にて広く公開している。出版論文リストなどはウェブページを参照してほしい。

第８章　今後の展開

本研究開発成果は、これからの産業システムの自動化、ならびに社会システムのIoT化にともなう重要インフラへのサイバー脅威に対して、その対策レベルを向上させることに役立つ。様々なシステムのIT化が進行する一方で、そのセキュリティに関しては十分な配慮が取られておらず、また十分な理解によるコストも費やされていない。そこで本システムを、特に中小規模の組織で専任のセキュリティ管理者を置くことが難しいような組織に対して提供すること目指している。そのためには、まず導入するためのコストを削減する必要がある。

本提案ではこの一案としてクラウド活用を推進した。パブリッククラウド上にてパッケージング化されたシステムとして導入でき、自社の機器から分析に必要となる情報をクラウドに転送するだけで、一定レベルのセキュリティアシストを受けることが可能となる。また日々の運用に関しても、専任セキュリティ管理者がいない環境を想定し、確度の高い情報提供と対策の提示ができ、いちいち必要な機器を操作することなく簡易な手順で必要な調査が行える環境を提供することを目指している。

さらに、本研究の目的のひとつであるサイバーセキュリティ研究に役立つオープンデータの提供に関しては、関連機関にて取得するデータを中心とした匿名化データセットを提供するための制度的ならびに技術的な研究開発および準備を進めている。

研究開発した技術については、まず大学内部による運用によってその有用性の確認と利便性の向上を目指したい。並行して、クラウド上における試験サービスとして展開し、中小規模の組織に対して試験サービスを展開することを目指す。セキュリティサービスの運用実験に興味のある大学、スポンサー組織の方々との連携を期待する。