Vol.2021-DPS-186 No.51
Vol.2021-CSEC-92 No.51
2021/3/16

IoT マルウェア自動検知のための 悪性コマンド列の特徴に対する概念ドリフト検出

小川真聖 [1]　松井俊浩 [2]

概要：近年，IoT マルウェアの増加が深刻な脅威になっている．この脅威に対して機械学習を活用した IoT マルウェア検知の研究が多くなされている．しかし，機械学習システムには，教師データの傾向や基準が変化する概念ドリフトの問題が存在する．概念ドリフトに起因する機械学習モデルの劣化により，新種や亜種の IoT マルウェアを検知できなくなるといった問題が存在する．そこで，本研究では，IoT マルウェアの悪性コマンド列の特徴に対する doc2vec とコサイン類似度による概念ドリフト検出手法を提案した．そして，実際に IoT マルウェアの実行する悪性コマンド列に対して本手法を適用し，概念ドリフトを検出できることを確認した．

キーワード：概念ドリフト，IoT マルウェア，悪性コマンド，機械学習

Concept drift detection for malicious command sequences for automatic identification of IoT malware

MASATOSHI OGAWA[1]　TOSHIHIRO MATSUI[2]

Abstract: In recent years, the increase in IoT malware has become a serious threat. Many studies have been conducted on IoT malware detection using machine learning against this threat. However, machine learning systems have the problem of conceptual drift in which the trends and criteria of teacher data change. Due to the deterioration of the machine learning model due to conceptual drift, there is a problem that new types and variants of IoT malware cannot be detected. Therefore, in this study, we proposed a conceptual drift detection method based on doc2vec and cosine similarity for the characteristics of malicious command sequences of IoT malware. Then, we applied this method to the malicious command sequence actually executed by IoT malware and confirmed that conceptual drift can be detected.

Keywords:Concept Drift, IoT malware, Malicious commands, Machine Learning

1. はじめに

近年，IoT マルウェアの増加が深刻な問題となっている．AV-test 社のレポート[1]によると，2018 年から 2019 年の間で新種の IoT マルウェアの出現数は，3 倍以上に増加していることが確認されている．IoT マルウェア増加の要因の一つに IoT マルウェア Mirai のソースコードがハッカーフォーラム上に公開されたことが考えられる．これに伴い，Mirai 亜種や新種の IoT マルウェアが急増している．IoT マルウェアの急増により，従来のパターンマッチングによる検知では，新種や亜種の IoT マルウェアを検知できないという問題が発生している．そこで，近年では，IoT マルウェアの増加に対して，機械学習を活用したマルウェア検知やマルウェア解析の研究が活発に行われ，目覚ましい成果を上げている．

1 情報セキュリティ大学院大学
INSTITUTE of INFORMATION SECURITY, Yokohama, Kanagawa 221-0835, Japan

2 情報セキュリティ大学院大学
INSTITUTE of INFORMATION SECURITY, Yokohama, Kanagawa 221-0835, Japan

ⓒ2021 Information Processing Society of Japan

Vol.2021-DPS-186 No.51
Vol.2021-CSEC-92 No.51
2021/3/16

しかし，機械学習を用いたマルウェア検知には，概念ドリフトの問題が存在する．概念ドリフトとは，モデルの学習に用いる学習データの特徴や傾向が変化する現象である．例えば，新型コロナウイルスによる人々の生活様式の変化により，観測されるデータの傾向が変化したことも一種の概念ドリフトである．また，新種や亜種が出現，増加している IoT マルウェアにおいては，特に概念ドリフトの発生頻度が多いと考えられる．概念ドリフトの発生に起因する問題として教師データの規則性や基準が変化し，マルウェアの検知精度が低下してしまうことが考えられる．

また，今後は，各 IoT デバイスの特性に応じたマルウェア検知の仕組みが必要になると想定される．このことから IoT デバイス上で自動的に機械学習モデルの構築，更新，マルウェア検知を行う自動検知の仕組みが必要となると考える．したがって，機械学習モデルの劣化に対するモデルの自動更新のためにも，概念ドリフトの検出は重要である．

本研究では，IoT マルウェアにより実行される悪性コマンド列の特徴に着目し，概念ドリフトの検出を行う．悪性コマンド列に着目した理由は，多くの IoT マルウェアに共通する特徴だからである．例えば，Mirai，Hajime，BrickerBot，VPNFilter といった IoT マルウェアは，共通して悪性コマンド列を実行する．

本研究では，二つの概念ドリフト検出手法を提案する．一つ目の手法は，自然言語処理手法の一つである doc2vec[2]と特異値分解(SVD)，K 近傍法を用いた手法である．二つ目の手法として，doc2vec とコサイン類似度による手法を提案する．

2. 関連研究

Bin Wu ら[3]の研究では，IDS(侵入検知システム)のために概念ドリフト検出技術を応用した研究を行った．この概念ドリフト検出の手法では，自然言語処理手法の一つである，LDA(Latent Dirichlet Allocation)用いている．LDA とは，文書解析にて，しばしば用いられるトピックモデルの一手法であり，対象の文書のトピックを推定する手法である．

この研究にて使用されているデータセットは，IDS の研究において，しばしば使用される KDD データセットを用いている．このデータセットには，偵察行為や DoS 攻撃，権限昇格攻撃等の攻撃パケットが含まれており，このデータセットを用いて，概念ドリフト検出による，攻撃パケットの検知が可能となっている．評価実験では，概念ドリフト検出手法を用いない IDS と筆者が提案する概念ドリフト検出に基づく IDS とで攻撃検知性能の比較を行い，筆者の提案する IDS は，検知精度が 96.9%で高い検知精度を維持しつつ，検知速度を改善している．

Anshuman Singh らの研究[4]では，マルウェアファミリーにおいて発生する概念ドリフトの追跡がなされている．この研究では， Agent，Hupigon，Pcclient の三種類のマルウェアファミリーを対象としている．また，マルウェアに含まれるユニークなオペコードの数を特徴量として扱っており，特徴量の抽出には，N-Gram の 2-Gram が用いられた．この研究では，3 種類のマルウェアファミリーのオペコード数に基づくコサイン類似度の変化により，3 種類のマルウェアファミリーにおける概念ドリフトの発生を追跡できることが確認されている．

3. 提案手法

本研究では，二つの概念ドリフト検出手法の提案を行った．

3.1 悪性コマンド列の特徴に対する K 近傍法を用いた概念ドリフト検出

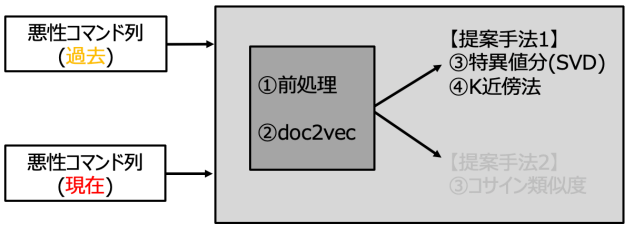

図 1 “悪性コマンド列の特徴に対する K 近傍法を用いた概念ドリフト検出手法”の概観

Figure 1 Overview of "conceptual drift detection method using K-nearest neighbor method for characteristics of malignant command sequence".

図 1 に一つ目の概念ドリフト検出手法の概観を示す．本手法では，IoT マルウェアによって実行された悪性コマンド列データの特徴を対象に概念ドリフト検出を行う．まず，過去のある一定期間に実行された悪性コマンド列と現在のある一定期間に実行された悪性コマンド列に対して，doc2vec を用いることで固定長の特徴ベクトルを抽出している．次に，抽出した特徴ベクトルに対して，特異値分解を用いた次元削減を行うことによって悪性コマンド列の多次元特徴ベクトルから 2 次元特徴ベクトルを抽出する．悪性コマンド列の特徴ベクトルから成る，過去データと現在データをそれぞれの時系列データとして扱う．最後に，過去の時系列データと現在の時系列データに対して，K 近傍法を用いた時系列データ間の比較をする．これによって，データの傾向に大きな変化が発生している箇所を検出する変化点検出を行う．この変化点検出がなされた場合，概念ドリフトが発生しているとみなす．

3.2 悪性コマンド列ベクトルのコサイン類似度による概念

ⓒ2021 Information Processing Society of Japan

Vol.2021-DPS-186 No.51
Vol.2021-CSEC-92 No.51
2021/3/16

ドリフト検出

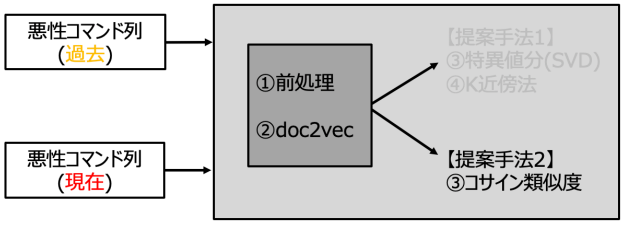

図 2 “悪性コマンド列の特徴に基づくコサイン類似度による概念ドリフト検出手法”の概観

Figure 2 Overview of "conceptual drift detection method based on cosine similarity based on the characteristics of malignant command sequences".

図 2 に本手法の概観を示す．本手法では，3.1 の手法と同じく，時系列順に実行された悪性コマンド列に対して，doc2vec を用いることで，悪性コマンド列をベクトルに変換する．過去の悪性コマンド列から抽出した特徴ベクトルと現在の悪性コマンド列から抽出した特徴ベクトルを比較し，コサイン類似度を算出する．そして，算出されたコサイン類似度よって概念ドリフトが発生したかを判断する．なお，3.1 の手法は，特徴空間におけるベクトル間のユークリッド距離に着目した概念ドリフト検出手法であるが，本手法は，特徴空間におけるベクトル間のなす角の大きさ，すなわち，コサイン類似度に着目したものである．つまり，過去データと現在データの特徴ベクトル間の類似度の高さによって，過去から現在にかけて，データの特徴が変化したかを判別できると考える．過去データと現在データの特徴ベクトル間のコサイン類似度が高い場合は，過去から現在にかけて，データの傾向に大きな変化は見られないとして，概念ドリフトは発生しないと判断する．コサイン類似度が低い場合は，過去から現在にかけて，データの傾向に大きな変化が見られたとして，概念ドリフトが発生したとみなす．

4. 実験

4.1 データセット

過去に収集された悪性コマンド列と現在収集された悪性コマンド列を用いた．過去の悪性コマンド列データは，2018 年 6 月にハニーポットを用いて収集されたものである．また，現在の悪性コマンド列データは，2020 年 1 月にハニーポットを用いて収集されたものである．データ収集に使用したハニーポットは Cowrie を用いた．Cowrie の構築と悪性コマンド列データの収集は，表 1 に示す環境で行った．また，2018 年 6 月に収集した悪性コマンド列 100 個と 2020 年 1 月に収集した悪性コマンド列 100 個の計 200 個を使用した．

表 1 ハニーポット Cowrie の構築環境

Table 1 Honeypot Cowrie construction environment

種別	環境名
プラットフォーム	AWS EC2
インスタンスタイプ	T2.Micro
CPU	Intel Xeon E5-2670 v2 2.50GHz
OS	Debian
ハニーポット	Cowrie

4.2 前処理

ハニーポットで収集した悪性コマンド列データに対して次の流れで前処理を行った．前処理を行うことで重要度の低い不要な情報を削減することが可能となる．前処理は，自本実験では，主にステミング処理，ノイズの除去，置換処理を行なった．

まず，ステミング処理をコマンド列に適用した例を次に示す．cat etc/passwd | grep user というコマンド列の場合は，cat，etc/passwd，|，grep，user という形にスペースで区切ることができる．

次に，ノイズ除去では，コマンド列内に出現する，乱数や欠損データ，文字数が多い文字列をノイズとみなして除去を行なった．

次に，置換処理を実施した．置換処理では，コマンド列内の引数や文字列等の置き換えを行う．例えば，http や https からはじまる引数は URL であると判断して，特定の文字列 “URL” に置き換えた．また，ポート番号や回数を指定するような数値として表される引数についても特定の文字列”Numerical value”に置換を行った．これらのデータを置換した理由としては，URL や数値引数のパターンは，再現なく存在するためである．特に URL は，攻撃者が使い捨てる前提で用意することが多いため，重要度が低く，不要な情報と考えられる．

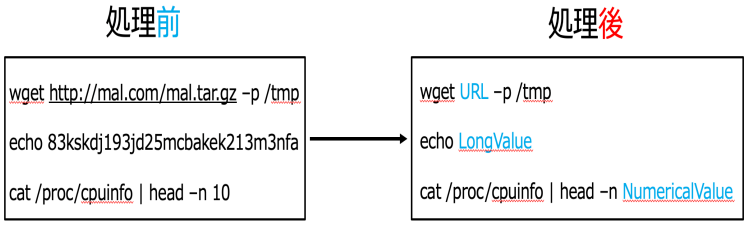

図 3 悪性コマンド列に対する前処理の例

Figure 3 Example of preprocessing for malicious command sequences.

なお，図 3 には，前処理を実施する前の悪性コマンド列の例と前処理を実施した後の悪性コマンド列の例を示している．

4.3 doc2vec を用いた悪性コマンド列のベクトル化

ⓒ2021 Information Processing Society of Japan

Vol.2021-DPS-186 No.51
Vol.2021-CSEC-92 No.51
2021/3/16

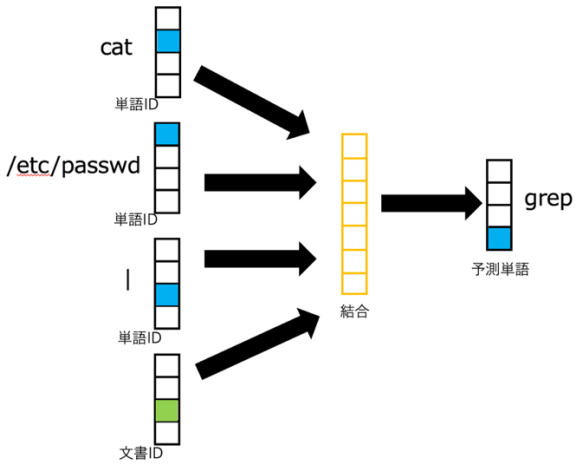

図 4　doc2vec によるコマンド列のベクトル化

Figure 4　Vectorization of command sequence with doc2vec

4.2 にて前処理を行なった悪性コマンド列に対して，doc2vec を用いることで悪性コマンド列の特徴ベクトルを抽出する．doc2vec には，PV-DM と PV-DBOW の二つのアルゴリズムが存在するが，本実験では，一般に精度が良いとされる PV-DM を用いた．図 4 には，doc2vec の PV-DM によるコマンド列の特徴ベクトルを獲得する際のニューラルネットワークを示している．図 4 に示す通り，入力層には，ベクトル化対象のコマンド列 ID とコマンド列に含まれる，それぞれのコマンドを入力している．そして，入力された各コマンドの次に出現するコマンドを予測する．このタスクを一定のコンテキストサイズごとに繰り返すことでコマンドの出現順序を考慮したコマンド列の特徴ベクトルを獲得することができる．例えば，図 4 では，cat etc/passwd | grep user というコマンド列があった場合，コマンド列に含まれる各コマンド cat，etc/passwd，| とコマンド列 ID をニューラルネットワークの入力層に入力し，|の次に出現するコマンドを予測している．

4.4 特異値分解による次元削減

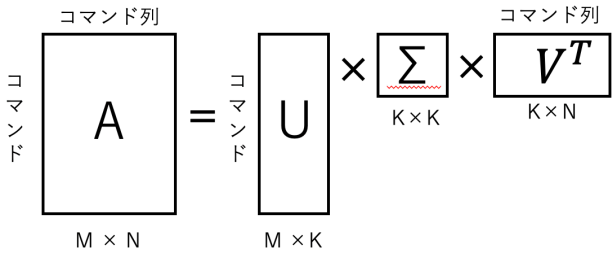

図 5　特異値分解による次元削減

Figure 5　Dimensionality reduction by singular value decomposition

次に 3.1 で述べた，一つ目の概念ドリフト検出手法についての実験を行なった．本実験では，特異値分解を用いて悪性コマンド列特徴ベクトルの次元削減を行なった．今回は，100 次元の特徴ベクトルから 2 次元の特徴ベクトルへの次元削減を行った．特異値分解を行うことで悪性コマンド列がもつ重要な情報を保持しつつ，情報量の集約をすることができる．これにより， 悪性コマンド列の 2 次元特徴ベクトルが抽出される．2 次元のベクトルに削減した理由としては，図 5 の対角行列 Σ に含まれる特異値の出力結果が，4.85，1.80，0.10，0.07(後略)であり，特異値 4.85 と 1.80 以外の軸は，情報として重要度が低い．そのため，本実験では，2 次元のみに着目し，次元削減を行なった．

4.5 K 近傍法による変化点検出

最後に，2018 年 6 月の時系列データと 2020 年 1 月の時系列データに概念ドリフトが生じているか，4.3.2 で述べた，K 近傍法による変化点検出で判断する．まず，2018 年 6 月の時系列データを K 近傍法により学習し，学習した時系列データと 2020 年 1 月の時系列データとのユークリッド距離を算出する．また，K の値は，2018 年 6 月の時系列データと 2020 年 1 月の時系列データの二つの時系列データ間のユークリッド距離を算出するため，K=2 を設定した．そして，算出されたユークリッド距離の大きさに応じて，変化点検出を行う．すなわち，ユークリッド距離が大きければ，概念ドリフトが発生しているとみなす．

4.6 コサイン類似度による概念ドリフト検出

次に 3.2 で述べた，悪性コマンド列の特徴ベクトルのコサイン類似度による概念ドリフト検出手法について実験を行なった．コサイン類似度は，2 つのベクトル間の成す角からの大きさから，ベクトル間の類似度として算出される．

本実験で用いたデータセットは，2018 年 5 月，2018 年 6 月，2020 年 1 月，2020 年 2 月の期間に，それぞれ収集した悪性コマンド列を使用した．また，各期間で収集した悪性コマンド列に対して前述した前処理と doc2vec を用いた特徴ベクトルの抽出を行なった．なお，獲得したベクトルの次元数は，100 次元である．

次に，ある 2 つの期間に収集された悪性コマンド列間のコサイン類似度を算出する．本実験では，下記の期間における悪性コマンド列間でコサイン類似度の算出を行う．

・2018 年 5 月に収集された悪性コマンド列と 2018 年 6 月に収集された悪性コマンド列

・2020 年 1 月に収集された悪性コマンド列と 2020 年 2 月に収集された悪性コマンド列

・2018 年 6 月に収集された悪性コマンド列と 2020 年 1 月に収集された悪性コマンド列

前述した通り，悪性コマンド列間の類似度が高いものについては，概念ドリフトが発生していないとし，類似度が低いものについては，概念ドリフトが発生しているとする．

5. 実験結果

5.1 “悪性コマンド列の特徴に対する K 近傍法を用いた概念ドリフト検出”の結果

ⓒ2021 Information Processing Society of Japan

Vol.2021-DPS-186 No.51
Vol.2021-CSEC-92 No.51
2021/3/16

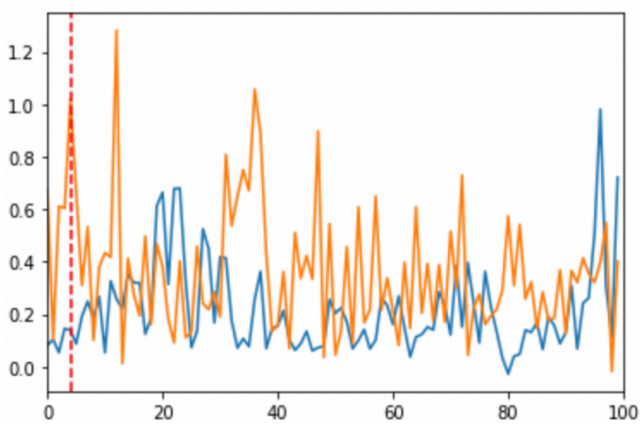

図 6　K 近傍法による変化点検出

Figure 6　Change point detection by K-nearest neighbor method

図 6 には，K 近傍法により，変化点検出された箇所を示している．グラフの横軸は，各悪性コマンド列が実行された時系列であり，縦軸は，各悪性コマンド列の特徴ベクトルである．青色の線が 2020 年 1 月のデータであり，橙色の線が 2018 年 6 月のデータである．検出された箇所では，2018 年 6 月のデータと 2020 年 1 月のデータ間でユークリッド距離が大きいことわかる．

5.2 “悪性コマンド列ベクトルのコサイン類似度による概念ドリフト検出”の実験結果

表 2　コサイン類似度の算出結果

Table 2　Cosine similarity calculation result

比較	コサイン類似度
2018年5月と2018年6月の比較	0.89
2020年1月と2020年2月の比較	0.87
2018年6月と2020年1月の比較	0.48

表 2 は，2018 年 5 月，2018 年 6 月，2020 年 1 月，2020 年 2 月に収集された，悪性コマンド列間のコサイン類似度の算出結果である．2018 年 5 月と 2018 年 6 月に収集された，悪性コマンド列間のコサイン類似度は 0.89 である．この値は，コサイン類似度としては高いと考えられるため，2018 年 5 月と 2018 年 6 月の悪性コマンド列間に大きな概念ドリフトは発生していないと判断できる．類似度が高い理由として，5 月と 6 月では，期間が短いため，データの傾向にあまり変化がなかったためと考えられる．

また，2020 年 1 月と 2020 年 2 月に収集された，悪性コマンド列間のコサイン類似度は 0.87 である．この値も同様に，コサイン類似度としては高いと考えられるため，2020 年 1 月と 2020 年 2 月の悪性コマンド列間に大きな概念ドリフトは発生していないと判断できる．類似度が高い理由として，2018 年 5 月，6 月と同様に，1 月と 2 月では，期間が短いため，悪性コマンド列に出現するコマンドやオプション，引数の種類や出現順序といった，データの傾向にあまり変化がなかったためと考えられる．

そして，2018 年 6 月と 2020 年 1 月に収集された，悪性コマンド列間のコサイン類似度は 0.48 である．この値は，コサイン類似度としては低いと考えられるため，2018 年 6 月と 2020 年 1 月の悪性コマンド列間に概念ドリフトが発生している可能性があると判断できる．類似度が低い理由として，2018 年 6 月と 2020 年 1 月とでは，期間に大きな開きがあるため，悪性コマンド列に出現するコマンドやオプション，引数の種類や出現順序といったデータの傾向が 1 年以上の間に変化したためと考えられる．

6. 評価

次に，本研究における概念ドリフト検出手法の有効性を評価するため，実際に悪性コマンド列検知器の構築を行い，検知器を用いた評価を行った．一般的に，概念ドリフトが発生した場合，モデル(検知器)の性能が劣化する．そのため，提案手法を用いて概念ドリフトが発生していると判断された，悪性コマンド列データに対する分類において，検知器の分類精度が低下した場合，実際に概念ドリフトが発生していると考えられる．本研究では，悪性コマンド列と良性コマンド列の二値分類を行う分類器を悪性コマンド列検知器とする．なお，悪性コマンド列検知器の構築にあたって， Pierre Dumont らの研究[5]を参考にした．Pierre Dumont らの研究は，ハニーポットで収集した悪性コマンド列に対して，N-Gram と K 近傍法を用いた機械学習モデルを構築し，悪性コマンド列と良性コマンド列の分類を行った研究である．

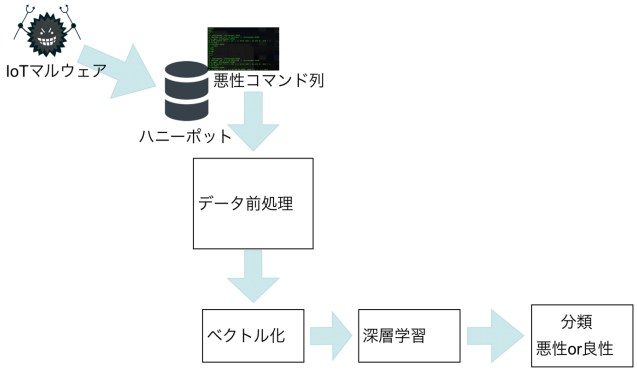

図 7　悪性コマンド列検知器構築の流れ

Figure 7　Flow of malignant command sequence detector construction

図 7 には，悪性コマンド列検知器を構築するまでのプロセスを表している．学習用データとして，Cowrie を用いて収集した，2003 個の悪性コマンド列を用いた．一方の良性コマンドは，Github にてユーザが公開しているコマンド履歴を用いた．なお，ハニーポット Cowrie を用いて収集したデータは，4 の実験で用いたデータセットと同様のものである．次に，収集した悪性コマンド列に対して，前処理を行う．前処理の手順は，4.2 で述べた前処理の手順と同様である．次に，前処理済みの悪性コマンド列をベクトルに変換する．

ⓒ2021 Information Processing Society of Japan

ベクトルへの変換には，doc2vec を用いた．最後に，深層学習では，LSTM を用いてモデルの学習を行い，悪性コマンド列と良性コマンド列の二値分類を行う検知器を構築する

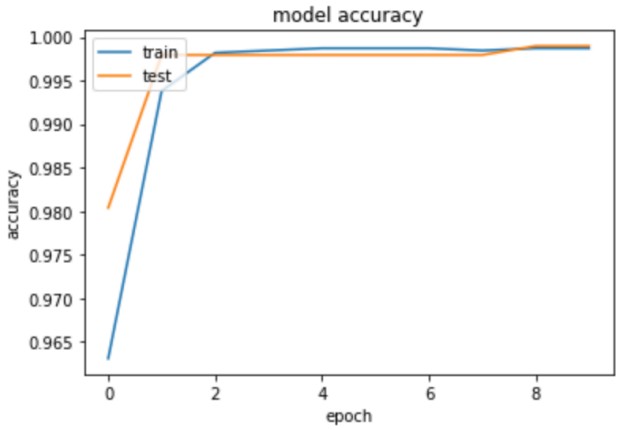

図 8　悪性コマンド列検知器の精度

Figure 8　Accuracy of malignant command sequence detector

図 8 は，悪性コマンド列と良性コマンド列に対する二値分類の精度を示すグラフである．学習用データに対する分類精度とテスト用データに対する分類精度が共に，99%以上であることがわかる．

提案手法で述べた概念ドリフト検出手法で，概念ドリフトが発生しているとされた 2020 年 1 月のデータを，2018 年 6 月のデータで学習した検知器に入力した場合，99%以上だった分類精度が 61%に低下したため，概念ドリフトによって検知器の性能が劣化したと考えられる．分類精度が低下した要因としては，2018 年 6 月のデータには出現しなかった未知のコマンドの存在やコマンド，オプション，引数の出現順序が異なる悪性コマンド列が 2020 年 1 月のデータに含まれていたからだと考えられる．したがって，実際に概念ドリフトが発生している可能性が高く，本概念ドリフト検出手法は有効であると考えられる．

7. おわりに

7.1 まとめ

本研究における概念ドリフト検出は，IoT マルウェアが実行する悪性コマンド列の特徴を対象とし，概念ドリフト検出をハニーポットで収集した実際の悪性コマンド列データに対して行った．本研究では，二つの概念ドリフト検出手法を提案した．一つ目の手法では，悪性コマンド列に対して，doc2vec を用いたベクトル化とベクトルの次元削減を行った．更に，次元削減を行った悪性コマンド列を，時系列データと見なし，K 近傍法を用いた変化点検出を行うことで概念ドリフト検出を行なった．二つ目の手法では，まず，2018 年 6 月と 2020 年 1 月の悪性コマンド列に対して doc2vec を用いてベクトル化を行なった．2018 年 6 月の悪性コマンド列ベクトルと 2020 年 1 月の悪性コマンド列ベクトル間のコサイン類似度に着目し，概念ドリフト検出を行った．

概念ドリフト検出手法の有効性を評価する実験では，実際に悪性コマンド列と良性コマンド列を分類する悪性コマンド列検知器を構築した．そして，上記，二つの概念ドリフト検出手法によって概念ドリフトが確認されたデータに対して検知器の分類精度が低下したことを確認した．

7.2 今後の課題

本研究では，悪性コマンド列の特徴に対する概念ドリフト検出を行なったが，攻撃者が独自に作成したコマンドや難読化されたコマンドに対しては，本提案手法が適用できない可能性がある．その場合は，シェルコマンドレベルではなくシステムコール等の更に低いレイヤーの特徴に着目する必要がある．

また，本研究の実験で使用した悪性コマンド列のデータは，ハニーポット Cowrie を用いて収集した．しかし，このハニーポットは，低対話型ハニーポットであるため，IoT マルウェアによっては，ハニーポットであることを検知して悪性コマンド列を実行せず，ハニーポットからログアウトを行う，IoT マルウェアの存在が確認されている．そういった IoT マルウェアが実行する悪性コマンド列を収集する方法についても検討する必要がある．

また，一つ目の提案手法には，閾値の設定によって，概念ドリフト発生の基準が変化してしまう問題が存在する．その他にも複数の問題を抱えているため，一つ目の提案手法よりも二つ目の提案手法の方が優れていると考える．

今回用いたデータセットは 2018 年に観測された悪性コマンド列と 2020 年に観測された悪性コマンド列の比較を行ったが，期間が開きが大きいため，実際の環境とは異なる．そのため，今後は，更に細かい期間での概念ドリフト検出が必要となる．

参考文献

[1]“The Security Status of IoT – High malware curve versus low learning curve”, July 22(2019) https://www.iot-tests.org/2019/07/the-security-status-of-iot-high-malware-curve-versus-low-learning-curve/

[2]Quoc Le and Tomas Mikolov: “Distributed Representations of Sentences and Documents”, International Conference on Machine Learning Beijing China 2014 (2014) https://arxiv.org/pdf/1405.4053.pdf

[3]Bin Wu, Hai Zhou Lin and Lin Feng: “Concept Drift Based on Subspace Learning for Intrusion Detection”, International Conference on Computer Engineering and Information Systems (2016)

[4]Anshuman Singh, Andrew Walenstein and Arun Lakhotia: “Tracking Concept Drift in Malware Families”, AISec '12: Proceedings of the 5th ACM workshop on Security and artificial intelligenceOctober, PP.81–92 (2012)

[5]Pierre Dumont, Roland Meier, David Gugelmann and Vincent Lenders: “Detection of Malicious Remote Shell Sessions” 2019 11th International Conference on Cyber Conflict https://ccdcoe.org/uploads/2019/06/Art_26_Detection-of-Malicious-Remote-shell-Sessions.pdf

ⓒ2021 Information Processing Society of Japan