Vol.2020-CSEC-91 No.18
Vol.2020-SPT-40 No.18
Vol.2020-EIP-90 No.18
2020/11/26

3D 画像識別によるマルウェア検知を目的としたプログラムの挙動の可視化に関する検討

山本真生[1,a)]　小林 良太郎[1]　加藤雅彦[2]

概要：近年の IT 技術の発達に伴い, マルウェアと呼ばれる悪意あるプログラムによって, 個人・組織問わず情報窃盗や機器の乗っ取りなどの被害が甚大化している. マルウェアを機械学習や深層学習を用いて検知する研究は行われているが, その検知率は課題が残る状況である. 一方, 画像処理における機械学習や深層学習を用いた分類・検出の技術は著しく進歩している. また, 3D 画像処理に関しても研究が盛んにおこなわれており, 医療など様々な現場で 3D 画像処理が活かされている. そこで本稿では, マルウェアの検知にそれらの画像処理技術を用いることを提案する. 画像処理技術を用いる前段階として, 通常プログラムとマルウェアの振る舞いを可視化し, それらの視覚的な差異を検証する. マルウェアの性質上, メモリ領域へのアクセスパターンは通常プログラムと異なる. 本研究では通常プログラムおよびマルウェアがアクセスしたメモリアドレスのログを取得し, 3 次元空間にプロットすることでプログラムの振る舞いを画像として出力する. その結果として, 通常プログラムとマルウェアの振る舞いを視覚的に分類可能であることを示す.

キーワード：マルウェア, 3D 画像, 深層学習, 判別, 可視化

1. はじめに

近年, 画像処理における深層学習の技術は著しく進歩している. 深層学習は人間の神経細胞（ニューロン）の仕組みを模したニューラルネットワークがベースになっており, 多層にしたニューラルネットワーク（DNN）を用いることでデータの特徴を段階的により深く学習することができる. DNN に大量の画像等のデータを入力することでデータの特徴を各層で自動的に学習し, 画像の識別などを高い精度で行うことができる. MathWorks の調査によると, 深層学習を用いた画像認識などのタスクの精度は人間の認知能力を超えるほどになっている [1]. また, 近年では 3D 画像における画像認識の研究が盛んにおこなわれており, 医療の分野においては Google が 3 次元深層学習を用いた「肺がん診断 AI」を開発している. この AI は複数枚の CT スキャン画像に基づいた肺の 3D モデルを作成し, 悪性腫瘍の有無を判別する. AI による判別は人間の放射線科医に匹敵する精度であると発表されている [2].

一方で, 情報窃盗や機器の乗っ取り, サービス不能攻撃など, マルウェアと呼ばれる悪性のプログラムによる被害が個人・組織問わず甚大化しており, 情報セキュリティにおいて大きな問題となっている. マルウェアの対策としてアンチウイルスソフトの導入が挙げられる. アンチウイルスソフトには, マルウェアに現れる共通のコードやハッシュ値などをあらかじめ解析し, 検査対象のファイルにそれらが含まれているかで安全であるかどうかを判別をするシグネチャ式と, プログラムの振る舞いに着目して判別をするビヘイビア式が存在する. シグネチャ式のアンチウイルスソフトではデータベースに登録されている情報を持つマルウェアは検知でき, 誤検知を大きく抑えることができる. しかし, 近年では既存のマルウェアを改造した亜種が多数登場しており, 従来のシグネチャ式アンチウイルスソフトで検知できない可能性がある. また, それらのマルウェアに対応するために検体を解析しなければならないが, 亜種を含めマルウェアの数は大幅に増加傾向にある. AV-TEST の調査によれば, 2019 年のマルウェアの総数は 10 億を超えており, また 2020 年は 6 月時点で 2019 年の総数を超えている [3]. このことから, 全てのマルウェアを解析することは困難である.

シグネチャ式に対し, ビヘイビア式のマルウェア検知はマルウェアの実行時に現れる特徴を収集し, 深層学習を用いた分類を行うことで, 未知のマルウェアにも対応できるようになることを目指している.

我々は, プログラムの挙動を画像として出力し, それらを

[1] 工学院大学 Kogakuin Uniersity
[2] 長崎県立大学 University of Nagasaki
[a)] j117291@ns.kogakuin.jp

Vol.2020-CSEC-91 No.18
Vol.2020-SPT-40 No.18
Vol.2020-EIP-90 No.18
2020/11/26

図 1 提案手法の概略図

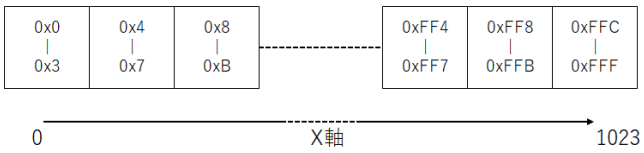

図 2 メモリアドレスに対応する x 軸座標

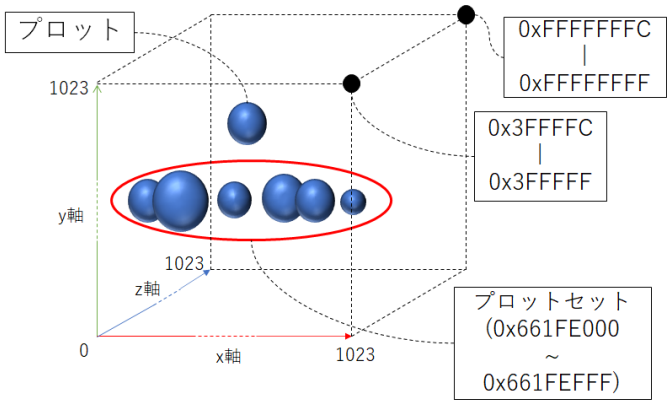

図 3 メモリアドレスに対応する x-y-z 立方体

特徴量とした深層学習を用いることでマルウェアを判別する手法を提案する．この提案手法の概略図を図 1 に示す．

本稿では図 1 のうち，通常プログラムとマルウェアのメモリアクセスログを可視化し，振る舞いの差異を視覚的に分類することに焦点を当てる．

2 章では提案手法について述べる．3 章では実装方法について述べ，4 章で評価環境と結果を示す．5 章で考察を述べ，6 章で本論文をまとめる．

2. 提案手法

本章では，プログラムのメモリアクセスのログを可視化する空間と，プロットの色や大きさの変化のプロセス，可視化プロセスの詳細を示す．

2.1 メモリアクセスを可視化する 3 次元空間の構成法

提案手法は，メモリへのアクセス頻度を描画する．プログラムが実行する際にアクセスするメモリ空間は広大であるため，各辺を 1024 個に区切った 3 次元空間を用意し，この空間内にプロットする．また，本稿では暫定的に 1 座標あたり 4Byte を割り当てる．x 軸，x-y 平面，x-y-z 立体がそれぞれ，4KB 分の軸，4MB 分の平面，4GB 分の立体に相当している．メモリアドレスに対応する x 軸方向の座標を図 2，立体における座標を図 3 に示す．また，3 次元空間において，1 つの球体をプロット，図 3 内で，赤い円で示されているような，x 軸に平行になるように一直線に置かれているプロットの集合体をプロットセットと呼称する．メモリはページ単位で管理されており，メモリ上の 1 つのプロットセットに対応する．描画対象のプログラムが使用しない空間も情報に含めることで，空間上の距離や位置も情報量として維持できる．メモリアドレスと用意したメモリ空間の対応は，メモリアドレスを $addr$，3 次元空間座標を (x, y, z) として式 (1)(2)(3) により求められる．また，メモリ空間にはアプリケーションを稼働するために用いられるユーザ空間とカーネルが稼働しているカーネル空間がある．それぞれは独立しており互いが互いのメモリ空間に直接アクセスできない．ユーザ空間は 0x00000000〜0xbfffffff のアドレスに相当しており，カーネル空間は 0xc0000000〜0xffffffff のメモリアドレスに相当している．

$$x = \left(\frac{addr}{4} \ \% \ 1024^2\right) \ \% \ 1024 \tag{1}$$

$$y = \left(\frac{addr}{4} \ \% \ 1024^2\right) \frac{1}{1024} \tag{2}$$

$$z = \frac{addr}{4} \cdot \frac{1}{1024^2} \tag{3}$$

2.2 アクセス頻度を基にしたプロットの変化

提案手法では，アクセス頻度を基にしてメモリアクセスを可視化する．図 4 にメモリアクセスの可視化例を示す．図 4 では z 軸方向の奥行きを省略している．図 4 のように空間のプロットとして，色や大きさをヒートマップのように変化させ，メモリアクセスを可視化する．

色の変化はヒートマップに相当するものを用いて，アクセス頻度が低い場合は青になり，徐々に頻度が上がるに連れて赤に変化していく．ある頻度 f の色は，f_{max} を頻度の最大値，f_{min} を頻度の最小値として式 (4) により正規化し，R, G, B を色の成分として式 (5)(6)(7) により求める．式 (5)(6)(7) と色の成分の関係を図 5 に示す．f_{max} と f_{min} は全プロットのうちの頻度の最大値と最小値の場合，1 度しかアクセスされない巨大なメモリ空間がある場合に全体的に青に寄ることや，ループカウンタ変数の膨大なアクセスがある場合に全体的に赤に寄ることが考えられる．色が赤や青に寄る場合，プロット同士の色差が小さくなり，アクセス頻度差が視覚的に認識しづらくなることが考えられる．そのため本研究では，色がアクセス頻度の偏りに影響しにくくなるように全プロットの頻度の第 1 四分位数を f_{min}，第 3 四分位数を f_{max} とした．

プロットの大きさはヒートマップを模して，頻度が低い場合は小さく，徐々に頻度が上がるにつれて大きく変化していく．ループカウンタ変数などによる局所的な頻度の爆発的な上昇がある場合，他のプロットに対して巨大化することが考えられる．そのため，頻度が上がるにつれて非線

Vol.2020-CSEC-91 No.18
Vol.2020-SPT-40 No.18
Vol.2020-EIP-90 No.18
2020/11/26

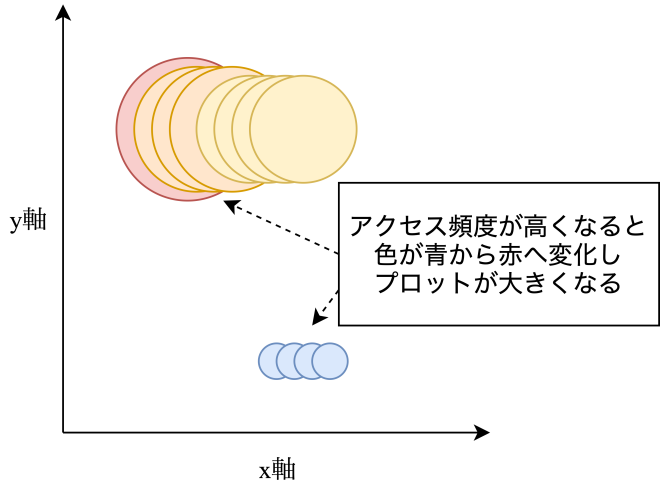

図 4 メモリアクセスの可視化例

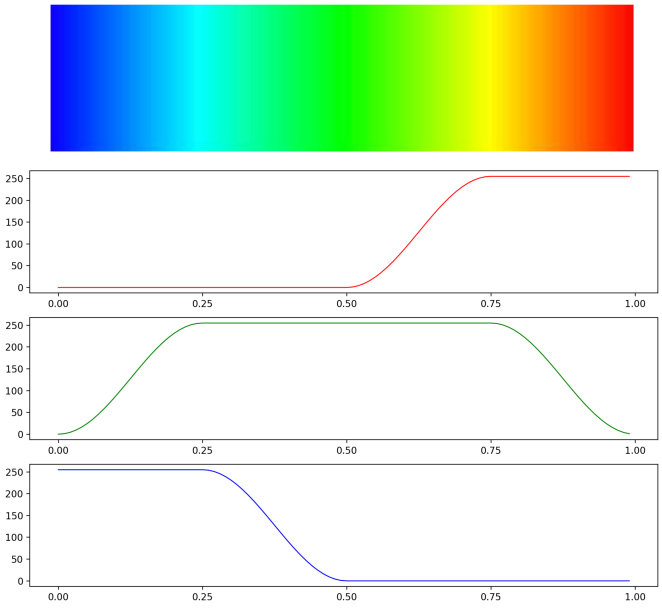

図 5 アクセス頻度と色の関係

型[*1]に大きさを変化させる．また，プロットの大きさに比例して透過度を上げることで，プロットが大きくなった際にも周りのプロットが隠れない．

$$f' = \frac{f - f_{min}}{f_{max} - f_{min}} \tag{4}$$

$$R = \begin{cases} 255 & (f' \geq \frac{3}{4}) \\ 255 \sin^2 2\pi f' & (\frac{3}{4} > f' \geq \frac{1}{2}) \\ 0 & (f' < \frac{1}{2}) \end{cases} \tag{5}$$

$$G = \begin{cases} 255 & (\frac{3}{4} > f' \geq \frac{1}{4}) \\ 255 \sin^2 2\pi f' & (1 > f' \geq \frac{3}{4},\ \frac{1}{4} > f' \geq 0) \\ 0 & (f' \geq 1,\ f' < 0) \end{cases} \tag{6}$$

[*1] 大きさは 3^{-1} 乗で減衰する

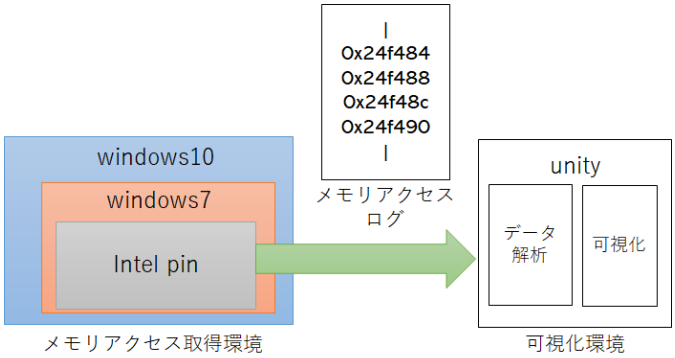

図 6 メモリアクセス可視化システム

$$B = \begin{cases} 255 & (f' < \frac{1}{4}) \\ 255 \sin^2 2\pi f' & (\frac{1}{2} > f' \geq \frac{1}{4}) \\ 0 & (f' \geq \frac{1}{2}) \end{cases} \tag{7}$$

2.3 可視化プロセス

メモリアドレスに対応する 3 次元空間の各座標の頻度は 0 から始まり，0 の時にはプロットは表示せず，0 以上の値をとる．メモリアクセスのログを 1 アクセス分読み込み，対応する座標の要素を増加させる．座標ごとの要素を座標ごとの頻度として 2.2 節の変化方法に従い，対応するプロットの色や大きさを変化させる．最後に空間の全要素を減少させ，次の 1 アクセス分を読み込み同じ動作を繰り返す．増加や減少させる大きさを調整することにより，任意の時間間隔においての頻度を求めることができる．

3. 実装

本章では，実装について述べる．本システムはメモリアクセス取得環境と可視化環境の 2 つからなっている．概要を図 6 に示す．メモリアクセス取得環境では，Windows10 上にゲスト OS として Windows 7 Professional を解析環境として用意し，解析環境上においてプログラム検体を実行し，Intel Pin[4] を用いて解析することによってメモリアクセスログを取得する．可視化環境では，Unity を用いてプロット，頻度計算による色や形の変形を行い描画する．

4. 評価

本章では評価を行う環境，その可視化した結果について述べる．

4.1 評価環境

評価は行う環境別に 3 種類の評価を行う．本稿では通常プログラムとマルウェアのメモリアクセスを可視化し，それらの差異を視覚的に認識できることを根拠に，提案手法の有用性を評価する．ただし，解析者が設定できるパラメータが多く，全ての組み合わせは膨大であるため，注目すべき結果のみ示す．また，今後深層学習を用いることを想定し

ているため, 可視化結果では, 任意の時間や角度で切り出した静止画像を示す.

可視化するメモリアクセスログはプログラム実行開始から 5 万回目のメモリアクセスまでとする. 描画を実行する際のパラメータは heat 値が 20, cool 値が 0.005 とする. これはメモリにアクセスされるごとに頻度が 20 上昇して 20 に相当するだけプロットが大きくなり色が赤に近づき, メモリにアクセスされないごとに頻度が 0.005 減少し, 0.005 に相当するだけプロットが小さくなり色が青に近づくということである.

4.2 可視化プログラム

本研究では通常プログラム 5 種類とマルウェア 6 種類を用いる. 本研究で用いたマルウェアはマルウェアデータベースである theZoo[5] から入手したものである. 通常プログラムの 1 つ目は Microsoft 社の web ブラウザである Microsoft Edge である. 2 つ目は Google 社の web ブラウザである Google Chrome である. 3 つ目は Microsoft 社が提供している Word である. 4 つ目は同じく Microsoft 社が提供している Excel である. 5 つ目は同じく Microsoft 社が提供している PowerPoint である.

マルウェアの 1 つ目は BigBang というマルウェアである. BigBang に感染すると .ppt や .pdf 等の拡張子を持つドキュメントを抜き取られるといった被害を及ぼす [6]. 2 つ目は DarkTequila というマルウェアである. DarkTequila に感染するとユーザのキー入力操作情報が記録されてしまい, パスワードなどの情報漏洩に繋がる [7]. 3 つ目は GrayFish というマルウェアである. GrayFish に感染するとルートキットが自動でコンピュータにインストールされてしまい, 感染したコンピュータへの侵入が容易になってしまう [8]. 4 つ目は PetrWarp というマルウェアである. Petrwarp に感染するとマスターブートレコードを書き換えられてしまい, コンピュータを起動できなくなってしまう [9]. 起動できなくなったコンピュータを人質にすることで身代金を要求するというランサムウェアと呼ばれるマルウェアの一種である. 5 つ目は Wannacry である. Wannacry は Petrwarp と同じランサムウェアの一種であるが, マスターブートレコードを書き換えるのではなく, コンピュータ内のデータを暗号化することで身代金を要求する. 6 つ目は KeyPass である. KeyPass はランサムウェアの一種であり, 感染すると Wannacry と同じくデータの暗号化を行い身代金を要求する. Wannacry と異なる点は攻撃者が手動でコントロールできる点であり, 暗号化対象とするファイルの指定や暗号キーの編集などができる [10].

4.3 評価項目

4.3.1 評価 1：通常プログラムとマルウェアの比較

評価 1 では, 通常プログラムとマルウェアではメモリアクセスの可視化にどのような差異が出るのかを比較するため, 複数の通常プログラムとマルウェアを可視化する. 評価 1 では 5 種類の通常プログラムと 5 種類のマルウェアを用いる. 用いる通常プログラムの情報を表 1, マルウェアの情報を表 2 に示す.

評価 1 の結果として Edge, Chrome, Excel, Word 及び PowerPoint の可視化結果を図 7 に示し, BigBang, DarkTequila, GrayFish, Petrwrap 及び Wannacry の可視化結果を図 8 に示す. 図 7 は左上から (a) Edge, (b) Chrome, (c) Excel, (d) Word, (e) PowerPoint の可視化画像である. 図 8 は左上から (a) BigBang, (b) DarkTequila, (c) GrayFish, (d) Petrwarp, (e) Wannacry の可視化画像である.

なお, 各図における赤緑青の軸はそれぞれ, x 軸, y 軸, z 軸である. x, y, z 軸が交わる原点が「0x0」のメモリを表している. また, 目視における差をわかりやすくするため, 3 次元空間を斜め前方から見た図を結果として採用する. また, 図 8 において, 特徴的な箇所を赤い円で示す.

4.3.2 評価 2：ランサムウェアの比較

評価 2 では, 同じ機能を持ったランサムウェアでも描画結果が変わる可能性があるため, 異なるランサムウェアの可視化画像を比較する. 評価 2 では 2 種類のプログラムを用いる. 1 つ目は評価 1 でも用いた Wannacry である. 2 つ目は KeyPass である.

評価 2 の結果として Wannacry と KeyPass の可視化結果を図 8 の (e) と (f) に示す. (e) は Wannacry の可視化結果であり, (f) は KeyPass の可視化結果である.

4.3.3 評価 3：実行回数での比較

近年の主要な OS（Windows, MacOS, Android 等）にはアドレス空間をランダム化する ASLR（Address Space Layout Randomization）機能が備わっている [11]. ASLR は主にバッファオーバーフロー攻撃からコンピュータを保護することに用いられる. バッファオーバーフロー攻撃はコンピュータの処理能力を超えたデータを送り付けて誤作動を誘発し, 正常に動かなくなっている間にコンピュータを乗っ取るという攻撃である. バッファオーバーフローを仕掛けるためにはプログラムがアクセスするメモリを把握する必要があるが, プログラムを実行するたびに ASLR によってメモリがランダムに移動するため, メモリの場所を把握することができなくなる. また, カーネル空間版の ASLR である KASLR（Kernel ASLR）も存在する.

評価 3 では, ASLR がメモリアクセス及び可視化結果にどのような変化をもたらすかを検証する. 評価 3 では評価 1 でも用いた PowerPoint と Wannacry を用いる. それぞれ 2 回目のメモリアクセスログを取得して可視化し, 1 回目の可視化画像との差異を示す.

評価 3 の結果として PowerPoint の 1 回目と 2 回目の可視化結果を図 9 に示し, Wannacry の 1 回目と 2 回目の可視化結果を図 10 に示す. それぞれの 1 回目の可視化結果

Vol.2020-CSEC-91 No.18
Vol.2020-SPT-40 No.18
Vol.2020-EIP-90 No.18
2020/11/26

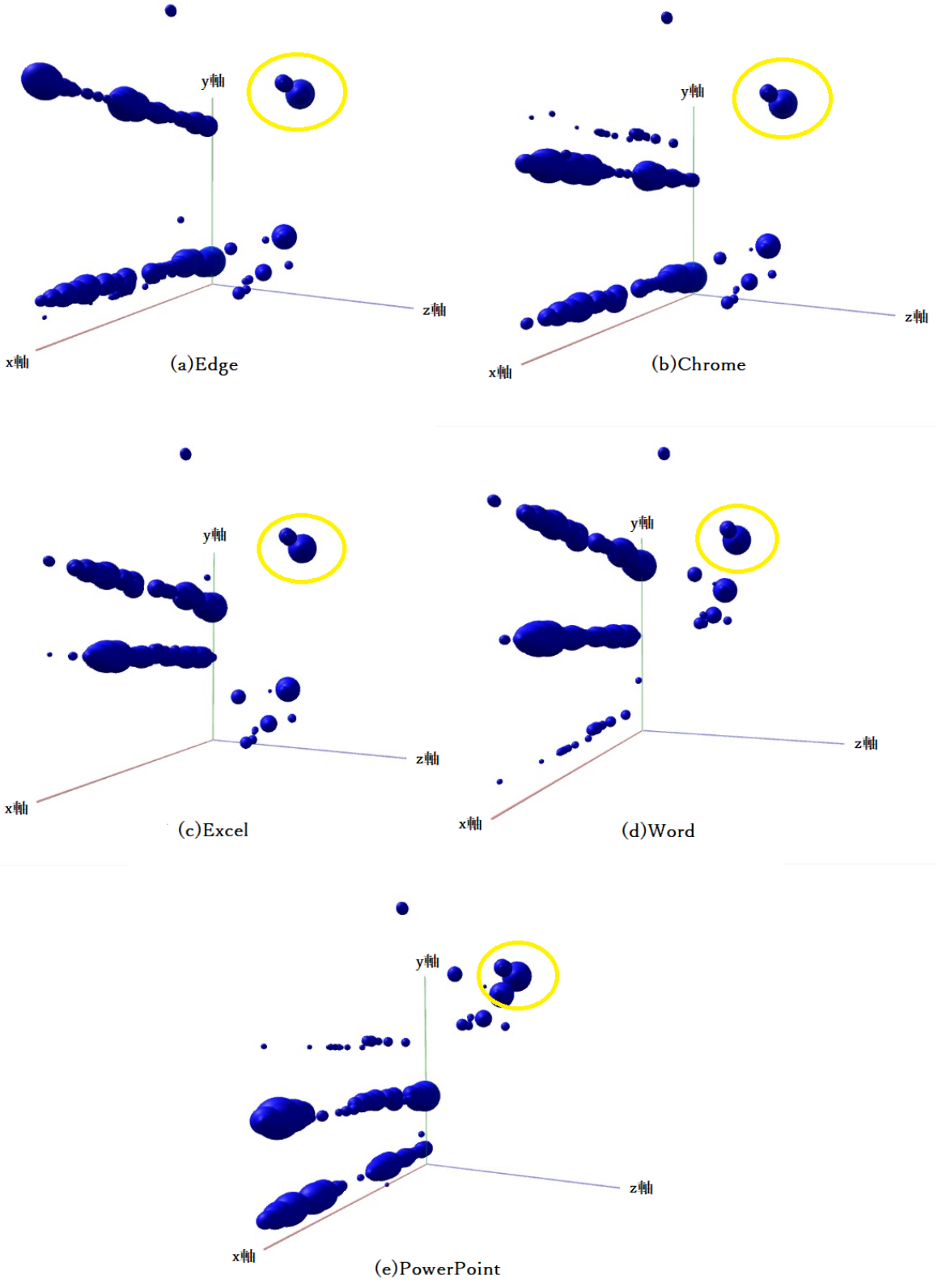

図 7 通常プログラムの可視化例

Vol.2020-CSEC-91 No.18
Vol.2020-SPT-40 No.18
Vol.2020-EIP-90 No.18
2020/11/26

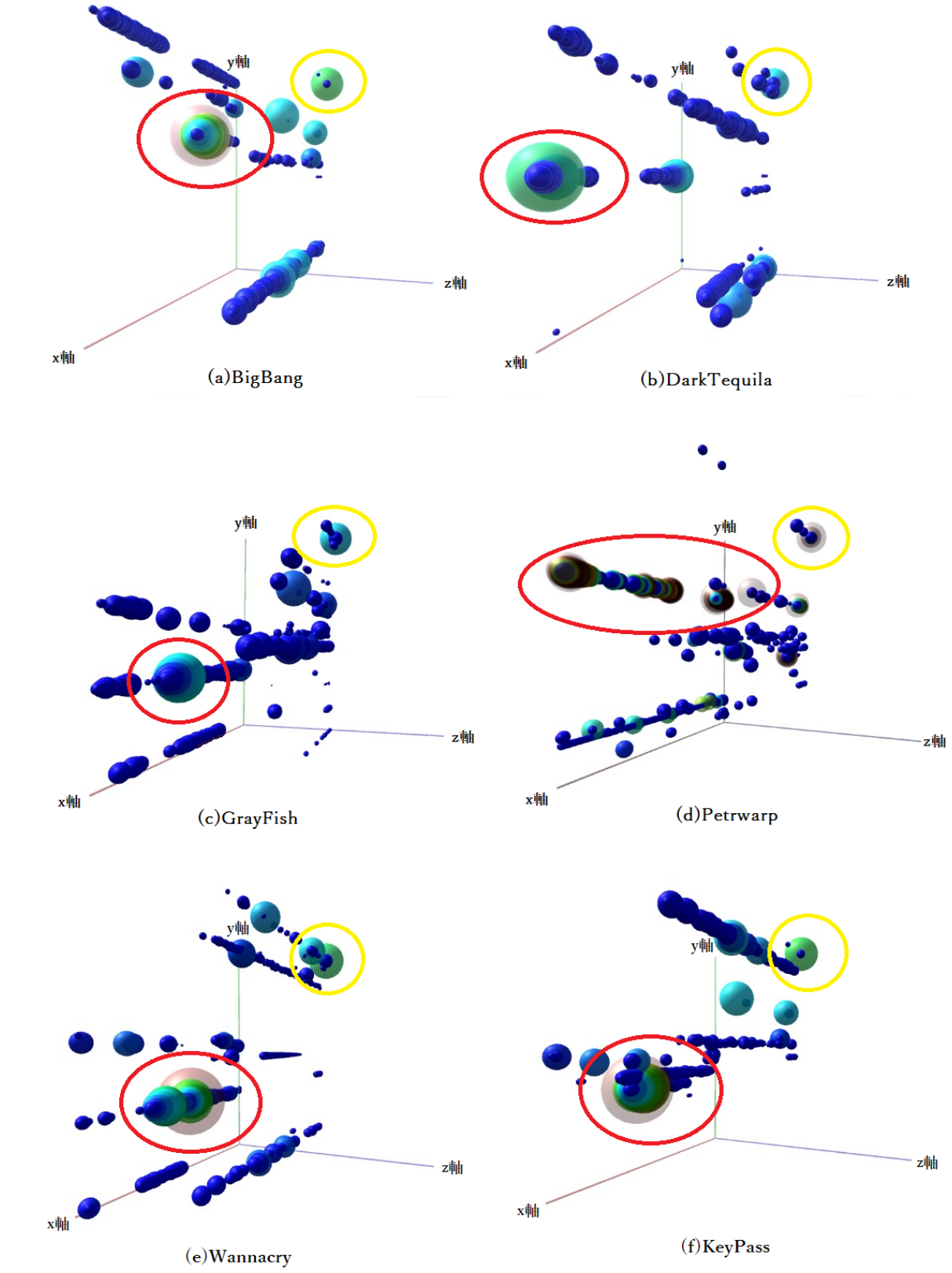

図 8 マルウェアの可視化例

Vol.2020-CSEC-91 No.18
Vol.2020-SPT-40 No.18
Vol.2020-EIP-90 No.18
2020/11/26

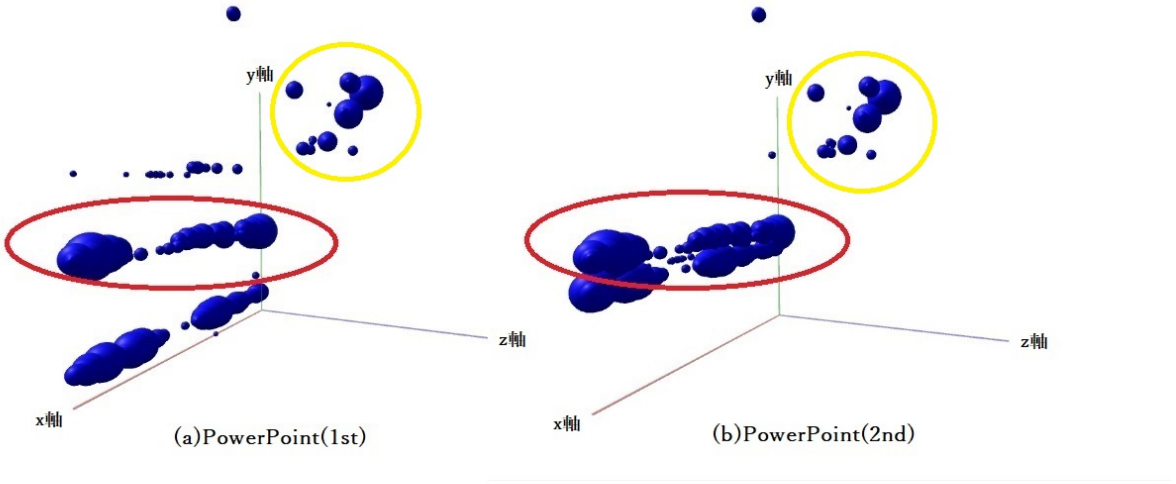

図 9 通常プログラム（PowerPoint）を 2 回実行した場合の差

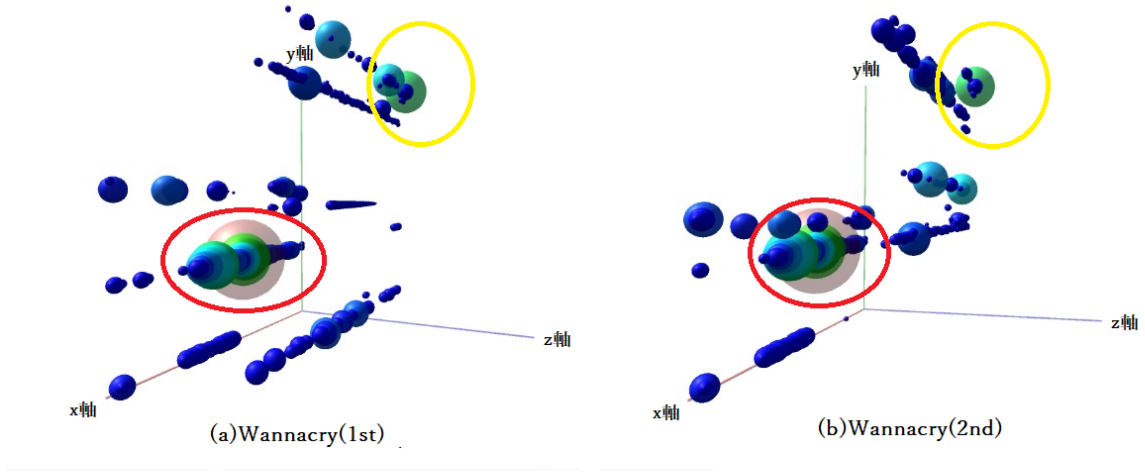

図 10 マルウェア（Wannacry）を 2 回実行した場合の差

は図 7, 図 8 のものと同一である.

5. 考察

図 7 と図 8 の可視化画像を見比べると, 通常プログラムは全体的に青いプロットが多いのに対し, マルウェアは緑や赤のプロットが所々に見られる. このことから, 通常プログラムはメモリアクセスが分散する傾向があり, 本論文で取り上げたマルウェアはメモリアクセスが一か所に集中する傾向があることがわかる.

図 8 の Wannacry と KeyPass の可視化画像を見比べると, 赤い丸で示したプロットセットは類似しているが, 全体として異なる特徴を示す. このことから, Wannacry と KeyPass は異なるプログラムとして判別されるが, 類似した特徴量にのみ注目すればランサムウェアかどうかを検出できる可能性がある.

図 7, 図 8 共に, プロット箇所が一致している部分を黄色の円で示す. この箇所は通常プログラムやマルウェアを問わずにアクセスされるメモリ領域であると考えられる. この箇所があることで通常プログラムとマルウェアの判別機が誤検知をする可能性がある.

図 9 の（a）と（b）の画像を比べると, 赤い円で示した箇所のプロットが一致している. また, 図 10 の (a) と (b) の画像を比べると, 赤い円で示した箇所に特徴的な部分が見られ, プロット箇所も一致している. 赤い円で示した箇所以外にも x 軸に平行なプロットセットの形は一致しており, プロットセット全体が ASLR によって平行に移動したこと

表 1 評価に用いた通常プログラムの情報

プログラム名	バージョン	ハッシュ値 (MD5)
Microsoft Edge	85.0.564.68	85e98a336aa437d912437149017a9c90
Google Chrome	85.0.4183.121	139654b5c1438a95b321bb01ad63ef6
Office Word	2016	44035f53bf0af73d13d88a7eec9843f1
Office Excel	2016	7ee616787f9bd52bc83445ed980bd52b
Office PowerPoint	2016	adcd84cd5e41ed5a812cee5282502ad2

表 2 評価に用いたマルウェアの情報

マルウェア名	ハッシュ値 (MD5)
BigBang	a233d90b8e5c19c4b3373bb76eb11428
DarkTequial	9fbdc5eca123e81571e8966b9b4e4a1
GrayFish	9b1ca66aab784dc5f1dfe635d8f8a904
PetrWarp	d2ec63b63e88ece47fbaab1ca22da1ef
Wannacry	84c82835a5d21bbcf75a61706d8ab549
KeyPass	6999c944d1c98b2739d015448c99a291

がわかる. このことから, 画像処理で判別を行う際には, プロットの場所ではなく形に注視すべきであるといえる.

6. まとめ

画像処理ベースの深層学習によるマルウェアの検知を行うため, その前段階として, プログラムの動作を可視化する新しい手法を提案した. 本稿では, 通常プログラムとマルウェアのメモリアクセスの差異を示した. 複数の評価環境で可視化を行い, その有用性の評価を行った. 今後は, 実際に本提案手法を用いた深層学習による検知を行うことを想定し, 各種パラメータの調整や, 画像判別器の作成などの検討を行う必要がある.

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] MathWorks, ディープラーニング これだけは知っておきたい3つのこと, https://jp.mathworks.com/discovery/deep-learning.html (参照 2020-9-7)
[2] 日経クロステック, グーグルが「肺がん診断 AI」、3D 画像認識の威力を知らしめた, https://xtech.nikkei.com/atcl/nxt/column/18/00692/052300004/ (参照 2020-8-4)
[3] AV.TEST, Malware, https://www.av-test.org/en/statistics/malware/ (参照 2020-9-7)
[4] Intel, Pin - A Dynamic Binary Instrumentation Tool, https://software.intel.com/en-us/articles/pin-a-dynamic-binary-instrumentation-tool (参照 2020-9-28)
[5] theZoo, theZoo aka Malware DB, https://thezoo.morirt.com/ (参照 2020-10-13)
[6] CHECKPOINT.COM, APT Attack In the Middle East: The Big Bang, https://research.checkpoint.com/2018/apt-attack-middle-east-big-bang/ (参照 2020-10-8)
[7] TREND MICRO, https://www.trendmicro.com/vinfo/jp/threat-encyclopedia/print/malware/tspy_darktequila.a (参照 2020-10-8)
[8] HACKERS CS FIRST, Grayfish…. rootkits, https://csfirst.wordpress.com/2017/10/26/grayfish-rootkits/ (参照 2020-10-8)
[9] WebTitan, PetrWrap Ransomware: An Old Threat Has Been Hijacked by a Rival Gang, https://www.webtitan.com/blog/petrwrap-ransomware/ (参照 2020-10-8)
[10] kaspersky daily, KeyPass: 無差別的なランサムウェア, https://blog.kaspersky.co.jp/keypass-ransomware/21251/ (参照 2020-10-27)
[11] Microsoft, セキュリティ対策の要点解説 第 13 回 Windows Vista のセキュリティ機能 〜 Address Space Layout Randomization 〜, https://docs.microsoft.com/ja-jp/security-updates/planningandimplementationguide/19871867 (参照 2020-10-14)