1

特徴量写像関数の学習による類似インシデント検索

大下　裕一
大阪大学大学院情報科学研究科

2

IoT機器におけるセキュリティインシデント

- IoT機器の普及
  - ネットワーク接続のカメラ
  - 工場自動化

  →ネットワーク接続機器数の増加

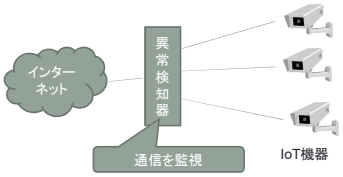
- IoT機器を対象とした攻撃が発生
  - Mirai
    - IoT機器に感染し、Botnetを構築
  - 監視カメラの情報漏洩
    - デフォルトパスワードで運用されていた監視カメラが世界中から閲覧可能な状態に

3

IoT機器におけるセキュリティ対応

- IoT機器は、CPU、メモリともに資源が限られている
  →IoT機器内でアンチウィルスソフトウェア等を動かすのは困難

ネットワーク側での異常検知→インシデント対応



4

異常検知後の対応は？

- 発生しているインシデントにより異なる
  - 機器が侵入されている場合　→　当該機器の取り換え
  - 外部から侵入を試みているものの失敗　→　FWの設定強化

異常検知されただけだと、どのように対処していいかは不明
- 異常検知は、インシデント対応のトリガでしかない

オペレータが発生した通信や機器の状況を解析し、対応を決める

5

研究の目標

オペレータがインシデント対応をする際に、有用な情報を提示することができるシステムの提案
- 過去に発生し、対応が完了したインシデントに関する情報を蓄積
- 蓄積された情報を検索することにより、現在のインシデントと類似の事象に関する情報を得る

↓

インシデントの分析の高速化、対応の高速化

6

インシデント検索システムの要件

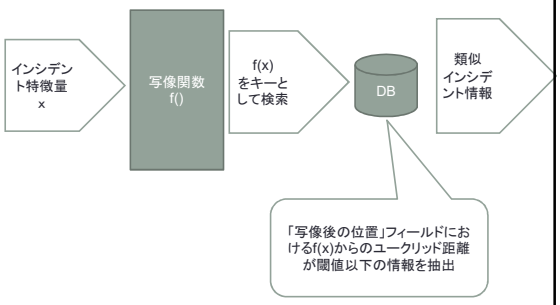
以下のような検索が可能であること
- 検索クエリ:
  現在発生したインシデントの特徴量
  （インシデント発生時に通信が行われていた宛先ポート番号や通信量等）
- 検索結果:
  現在発生したインシデントに類似したインシデントの詳細情報

留意事項
- インシデントの特徴量が類似＝同種のインシデントではない
- どのような種類のインシデントが発生するかは、事前には分からない
  - 新種のインシデントの発生の可能性もあり

本研究のアプローチ
- インシデントの特徴量を写像する写像関数を導入
- 類似のインシデントを近い位置に写像することにより、過去の類似のインシデントの情報を取得

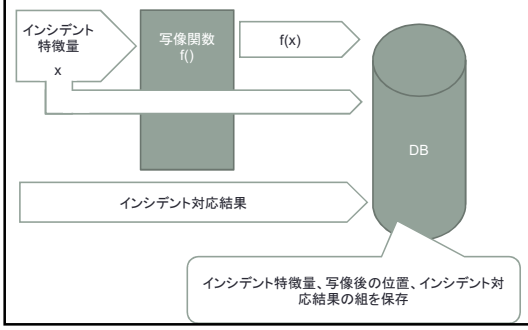
7

提案手法の概要　～検索～



8

提案手法の概要　～情報の蓄積～



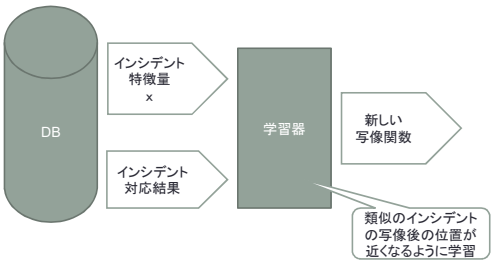
9

写像関数の学習

- どのインシデント同士が類似であるかは、インシデント対応後に判明
  類似のインシデントと考えられる例
  - インシデント対応情報に含まれる、インシデントの種類に関する情報が同一
  - インシデント対応の手順が同じ
- インシデント情報を蓄積後に、その情報をもとに写像関数を逐次学習することが望ましい

10

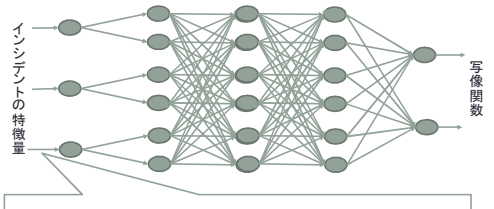
提案手法の概要　～写像関数の学習～



※　写像関数の更新後は、必要に応じてDBの写像後の位置も更新

11

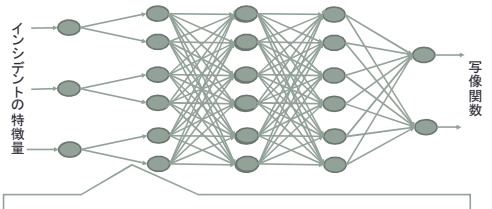
写像関数の具体例：ニューラルネットワーク



インシデントの特徴量ベクトルを入力

12

写像関数の具体例：ニューラルネットワーク



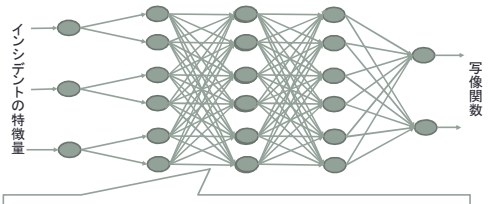
第一層：特徴(0,1)の抽出

$$y^{(1)}_{in+j} = \sigma\left(a^{(1)}_{in+j} v_i + b^{(1)}_{in+j}\right)$$

閾値 $b^{(1)}_{in+j}$ / $a^{(1)}_{in+j}$ を超えると1に近く、それ以外は0に近い値を出力

13

写像関数の具体例：ニューラルネットワーク

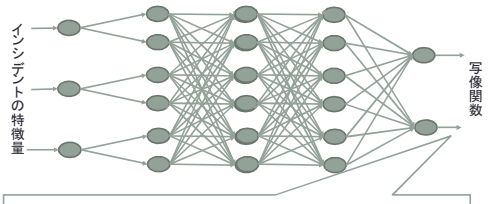


第二層～：全結合層

$$y_j^{(n)} = \max\left(0, \sum_i \left(w_j^{(n)} y_j^{(n-1)} + b_j^{(n)}\right)\right)$$

14

写像関数の具体例：ニューラルネットワーク



最終層の出力を、写像結果とする

15

写像関数の学習方法の具体例

ミニバッチを構成し、ニューラルネットワークを誤差逆伝搬により学習

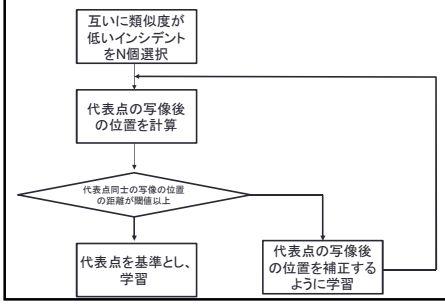
ロス関数：$\sum_{f_1,f_2 \in F} \left(S(f_1,f_2) \left(\frac{f(v_{f_1}) - f(v_{f_2})}{\alpha} \right)^2 \right) + \sum_{f_1,f_2 \in F} \left((1 - S(f_1,f_2)) \frac{1}{\left(\frac{f(v_{f_1}) - f(v_{f_2})}{\alpha} \right)^2} \right)$

- $S(f_1,f_2)$：f1とf2の類似度（0以上1以下）
- 類似度が低いフロー同士は斥力、類似度が高いフロー同士は引力が働くロス関数

16

写像関数の学習の高速化

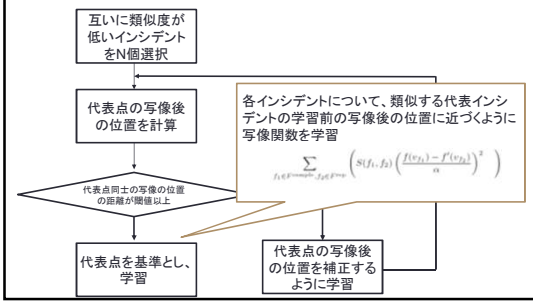
方針： 代表点を選択し、代表点を基準に学習を行う



17

写像関数の学習の高速化

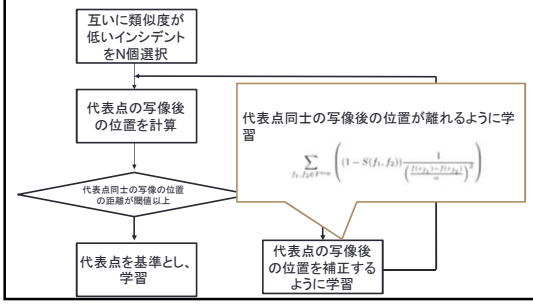
方針： 代表点を選択し、代表点を基準に学習を行う



18

写像関数の学習の高速化

方針： 代表点を選択し、代表点を基準に学習を行う



19

インシデント発生時のフロー分類への適用

以下の特徴量を生成

説明	値
同時刻の同一 IP アドレス宛のフロー数	整数値
同時刻の同一 IP アドレス発のフロー数	整数値
同時刻の同一ポート番号宛のフロー数	整数値
同時刻の同一ポート番号発のフロー数	整数値
フロー中の監視対象機器への総パケット数	整数値
フロー中の監視対象機器への総トラヒック量（Byte）	整数値
フロー中の監視対象機器からの総パケット数	整数値
フロー中の監視対象機器からの総トラヒック量 (Byte)	整数値
フロー中の監視対象機器への平均パケットレート（packets/sec）	実数値
フロー中の監視対象機器からの平均パケットレート（packets/sec）	実数値
フロー中の監視対象機器への平均トラヒックレート（Byte/sec）	実数値
フロー中の監視対象機器からの平均トラヒックレート（Byte/sec）	実数値
フロー中の監視対象機器の送信トラヒック量と対象機器の受信トラヒック量の比	実数値
監視対象機器側のポート番号	代表的なポート番号について One Hot 化した値
監視対象機器の通信相手側のポート番号	代表的なポート番号について One Hot 化した値
プロトコル番号	代表的なプロトコルについて One Hot 化した値
開始 TCP フラグ	各フラグについて ON なら 1，OFF なら 0
逆方向開始 TCP フラグ	各フラグについて ON なら 1，OFF なら 0

20

動作実験（フロー分類への適用）

実装環境
- OS: CentOS
- 言語：Python
- ニューラルネットワークフレームワーク： Chainer
- データベース：PostgreSQL

評価目標
- 新種のフローが登場し始めたのち、当該種類のフローについて、正しい情報が検索できるようになるまでに、必要な学習サンプル数が少ないことを示す

評価指標
- 精度：
  システムから検索された最も類似するフローが、検索したフローと同種のものである割合

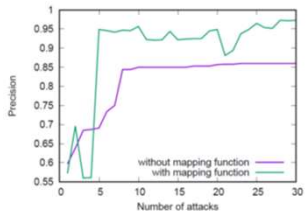
評価手順
1. 評価対象の種類以外のフローデータを学習
2. 評価対象の種類のフローデータを一つ学習
3. 学習したもの以外の評価対象フローを検索キーとして用いた場合の精度の評価
4. 2に戻る

比較対象
- 特徴量ベクトル同士を比較し、DBからユークリッド距離が最小のフローを抽出する手法

21

評価結果

- 写像関数を通すことにより、一定数の経験が蓄積されると、高い精度で分類が可能



22

まとめ

- 写像関数の学習を通して、類似のインシデントを検索することができるシステムを提案
- インシデント発生時のフローの検索に適用し、評価。

今後の予定
- インシデント全体の特徴から、類似インシデントを検索する場合の評価