Computer Security Symposium 2019
21 - 24 October 2019

分散表現を用いたアラートログにおけるアノマリ検知

江田 智尊[1,a)]　及川 孝徳[1]　古川 和快[1]　村上 雅彦[1]

概要：セキュリティ機器のあげるアラートは増加の一途をたどり，サイバー攻撃対策を行う SOC 業務従事者のログ分析負担は増加している．それに伴い，膨大なログの中から潜在的脅威であるアノマリを効率的に検知する機械学習分析技術が注目されている．しかしアラートログは，IP アドレスやポート番号といったパターン数が膨大なカテゴリカル変数を含むため，計算コスト・精度の面で分析が困難となる．そこで本稿では，アラートログを自然言語とみなし，同分野の分散表現技術 [word2vec] に基づくアノマリ検知技術を提案する．提案技術により，検知対象の特徴量を，低次元かつ検知に適した形で分散表現することを可能にし，アノマリ検知の精度改善を実現する．数値実験では，ネットワーク型 IDS アラートから攻撃者 IP アドレスを検知するタスクに取り組み，従来手法を上回る検知精度を得た．

キーワード：アノマリ検知，機械学習，自然言語処理，IDS

Anomaly detection for alert logs with word embeddings

Satoru Koda[1,a)]　Takanori Oikawa[1]　Kazuyoshi Furukawa[1]　Masahiko Murakami[1]

Abstract: The number of alerts raised by security protection systems has been constantly increasing, and the burden of security operators has also been increasing accordingly. At the same time, machine learning algorithms, which can detect potential threat [anomaly] among a large amount of alerts, have been paid attention. However, it is pretty hard to analyze alert logs efficiently because they typically contain categorical variables with enormous patterns such as IP addresses and port numbers. The paper proposes an algorithm which can detect anomaly on such logs on the basis of the embedding approach in the natural language processing field, word2vec. The proposed algorithm extracts low-dimensional, feasible features of objects for anomaly detection from logs. In our experiments, the proposed approach outperformed a baseline approach on the task of detecting attackers' IP addresses from network IDS logs.

Keywords: Anomaly detection，Machine learning，Natural language processing，IDS

1. はじめに

サイバー攻撃の巧妙化・増加に伴い，セキュリティ検知機器があげるアラートは増加の一途を辿っている．アラートの分析・対応は，組織のセキュリティ管理部門である Security Operation Center (SOC) のオペレータが担当する．しかし Imperva 社の調査 (2018 年) によると，SOC の 27% が 1 日に 100 万件以上のアラートを検知しているとの報告がある [1]．このような状況下では SOC は優先度が最高レベルのアラートに限ってしか対応できず，優先度低・中レベルのアラートを分析するには困難な状況である．しかしサイバー攻撃は往々にして，高レベルのアラートの前に予兆的な低・中レベルのアラートをあげるため，そのようなアラートから潜在的脅威を検知できれば，被害が深刻化する前に脅威を捉えることが可能になる．

この状況を打開するため，機械学習的アノマリ検知技術のアラート分析への活用が広まっている．アノマリ検知では通信のアノマリ的振る舞いを，セキュリティ検知機器があげるアラートログを分析して検出する．例えば組織の通信を監視する場合，従業員の業務通信は多数が同じような通信の振る舞いをする一方で，攻撃や攻撃前の探索のよう

[1] 株式会社富士通研究所 セキュリティ研究所
FUJITSU LABORATORIES LTD.
[a)] koda.satoru@fujitsu.com

な通信はアノマリな動作になると考えられる．アラートログを機械学習で分析してアノマリな通信を行う端末の IP アドレス [アノマリ IP アドレス] を検知することで，脅威を早期に検知することが期待される．

機械学習のアノマリ検知手法は多く開発されているが，その多くは連続値データを扱う [2], [3], [4]．従ってセキュリティログのようなカテゴリカル変数を含むデータを扱う場合，検知の前処理で数値データの特徴量を抽出する必要性が生じる．しかしセキュリティログの場合，単純な変換による抽出 (one-hot encoding など) では，変換後データの次元が膨大となり，“次元の呪い” と呼ばれる精度が向上しない問題が生じる．また特徴量を自前で設計する場合は，その特徴量が本当に分析に有効であるかを十分検証する必要がある [5]．そこで近年では，離散的なテキストデータを解析する自然言語処理分野の手法がセキュリティログ分析及び特徴量抽出に応用されている (Recurennt Neural Network [6], entity embedding [7], word2vec [8], [9])．特に word2vec をログ分析へ応用した IP2Vec (Ring *et al.*, 2017 [9]) は，IP アドレスの分散表現（特徴量）をアラートから学習する技術として有効であり，抽出した特徴量にアノマリ検知アルゴリズムを適用することで，膨大なログに潜むアノマリ IP アドレスの検知を実現する．しかし IP2Vec は IP アドレスとそれらの関係性をベクトル表現することが目的であるため，アノマリ検知という特定のタスクにおいてはより適した特徴量があると考えられる．

そこで本論文では，より検知に適した IP アドレス特徴量の抽出とアノマリ検知精度の改善を実現する手法を提案する．本手法は，IP2Vec に基づく IP アドレスの特徴量抽出時にアノマリ検知の評価を織り込むために，特徴抽出とアノマリ検知の損失関数を同時に最適化する．具体的には後述するが，IP2Vec の損失関数 L_{i2v} に，検知の評価に関する正則化項 R を加えた目的関数 $L = L_{i2v} + \lambda R$ (式 10) を新たに定義し，この目的関数の最小化を実行する．これによりカテゴリカル変数からアノマリ検知に適した IP アドレス特徴量を抽出することを可能にし，結果的にアノマリ検知精度の改善を実現する．

本技術の貢献は以下の通りである．

(1) アノマリ検知に適したカテゴリカル変数からの特徴抽出を実現する．連続値データで似た手法は存在するが ([2], [3], [4])，提案手法ではカテゴリカルデータでこれを実現する．
(2) セキュリティログのような膨大なカテゴリカル変数を含むデータの中から，潜在的な脅威であるアノマリ IP アドレスを高精度に検知することが可能になる．これにより，アラート内の潜在的脅威の早期発見と，加えて SOC オペレータのアラート分析業務の負担軽減を実現する．

数値実験では，ネットワーク型 IDS [Intrusion Detection Systems] アラートから攻撃者 IP アドレスを検知するタスクにおいて，従来手法を上回る検知精度を得た．あるデータセットにおいては，約 1400 IP アドレス を含む 1 万行のログを分析し，攻撃者端末の IP アドレス 1 つを誤検知なしで検知することを実現した．より大規模な 10 万ログのデータ (約 4000 IP アドレスを含む) においても，ほぼ完全な検知を実現した．

本論文の構成は以下の通りである．2 章では基本的な概念についての準備と，提案手法の関連手法を詳細に説明する．3 章では提案手法について，定式化と最適化に関して述べる．4 章ではオープンデータを用いた評価実験の結果を紹介し，5 章でまとめと今後の課題について述べる．

2. 準備

本章では後述する提案技術のための準備として，アラート分析の概要説明を行う．それに続き，関連技術として word2vec, IP2Vec の詳細説明を加える．

2.1 アラート分析概要

まず始めに本稿で行うアラート分析の対象と目的を述べる．分析対象として一般的なネットワーク型セキュリティ検知のアラートを想定する（例：IDS アラート）．アラートは，図 1 上段に示されているような行列形式のデータで与えられているとする；各行が 1 アラートに対応し，各アラートには送信元 IP アドレスと，その他の通信情報（宛先 IP アドレス・ポート，プロトコル等）が記録されているとする．本アラート分析の目的は，アラートに含まれる全送信元 IP アドレスの中から，他とは異なる振る舞いをしたアノマリ IP アドレスを通信情報を基に検知をすることとする．例えば，多くが業務通信を行う IP アドレスの中から，攻撃もしくは攻撃準備行動を行う端末の IP アドレスを検知する状況を想定する．全体的な分析フローを，図 1，及び以下に記す．

(1) アラートから，アラートに含まれる全送信元 IP アドレスの特徴抽出を行う．この特徴抽出は，送信元 IP アドレスが行った通信情報を基に，IP アドレスをベクトル表現することである．
(2) IP アドレスの特徴量を基に，IP アドレスのアノマリスコアを算出する．一般には，One-Class SVM [Support Vector Machine] 等の検知アルゴリズムを適用することで，アノマリスコアを算出する．
(3) アノマリスコアを基に，アノマリ IP アドレスを決定する．一般に，アノマリスコアが高い IP アドレスをアノマリとして検知する．

2.2 word2vec

本小節では，上記フローのステップ (1) の IP アドレス特徴抽出に関し，後述の提案技術に関連する特徴抽出技術と

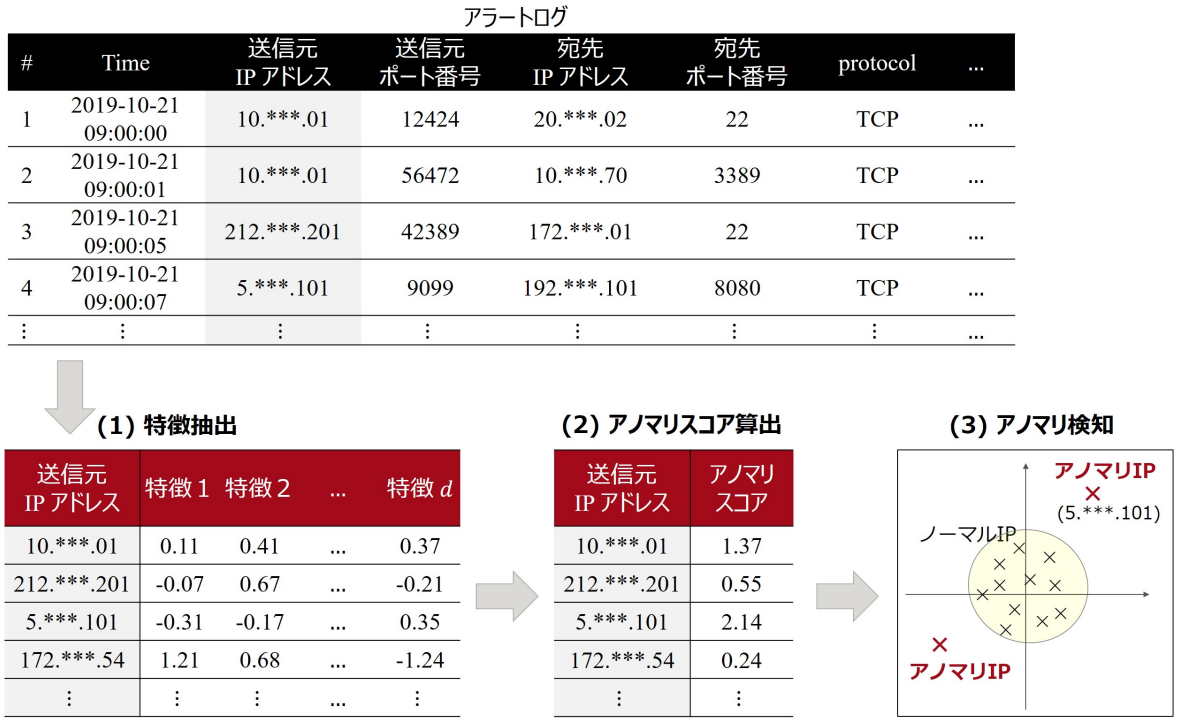


図 1　アラート分析フロー

して word2vec アルゴリズム [10] を説明する．word2vec は自然言語処理分野で開発された手法で，元々は単語の分散表現 (特徴量) を学習する手法として開発された．以下に記号を導入する．

いま，p 個の単語を含む文章が与えられているとする．各単語を 1 対 1 に自然数へ対応することで，単語を w_i ($\in \{1, \cdots, p\}$) と表現する．簡単のため，$w_i = i$ とする．各単語の周辺語 (コンテクスト) の集合を $C(w_i) \subset \{1, \cdots, p\}$，$C(w_i)$ の要素を $w_{c,i} \in C(w_i)$ と記述する．w_i に対応する one-hot ベクトルを $\boldsymbol{x}_i \in \{0,1\}^p$ とする；i 番目の要素 x_i が 1，それ以外の要素が 0 の p 次元ベクトルである．

次にネットワークに関する記号を導入する．word2vec は，隠れ層 1 層のニューラルネットワークとして表現される．そのネットワーク構造には，Skip-gram と Continuous Bag-of-Words と呼ばれる構造が存在する．今回は Skip-gram 形式のネットワーク構造に倣いネットワークを定義する．ネットワークの入力層 $\boldsymbol{x} \in \mathbb{R}^p$ には，単語の one-hot ベクトルが入力される．隠れ層，出力層（最終層）の出力は，重み行列 $U = (\boldsymbol{u}_1, \cdots, \boldsymbol{u}_p) \in \mathbb{R}^{d \times p}$，$U' = (\boldsymbol{u}'_1, \cdots, \boldsymbol{u}'_p) \in \mathbb{R}^{d \times p}$ を用いてそれぞれ，

$$\boldsymbol{h} = U\boldsymbol{x} \in \mathbb{R}^d, \tag{1}$$

$$\boldsymbol{y} = \mathrm{softmax}(U'^T \boldsymbol{h}) = \frac{\exp(U'^T \boldsymbol{h})}{\sum_k \exp(\boldsymbol{u}'^T_k \boldsymbol{h})} \in \mathbb{R}^p, \tag{2}$$

と表現される．各単語 w_i に対し，隠れ層の出力 $\boldsymbol{h}_i = U\boldsymbol{x}_i = \boldsymbol{u}_i$ が，ベクトル表現された単語の特徴量となる．ベクトルの次元数 d は，数十〜数百で十分とされている．

最後にネットワークの学習について議論する．skip-gram 形式の word2vec では，単語が与えられたときの，その前後に周辺語が共起する尤度に基づいて損失関数を定義し，その損失関数の最小化によってパラメータ (U, U') を学習する．単語 w_i と，その周辺語が共起する確率は，以下で与えられる：

$$\prod_{w_{c,i} \in C(w_i)} P(w_{c,i}|w_i), \tag{3}$$

$$P(w_{c,i}|w_i) = \frac{\exp(\boldsymbol{u}'^T_{c,i} \boldsymbol{u}_i)}{\sum_k \exp(\boldsymbol{u}'^T_k \boldsymbol{u}_i)}. \tag{4}$$

ここで，$\boldsymbol{u}'_{c,i} := U'_{w_{c,i}}$，即ち行列 U' の第 $w_{c,i}$ 列と表記する．$P(w_{c,i}|w_i)$ は，ネットワークに $\boldsymbol{x}_i$ を入力したときの，出力層 $\boldsymbol{y}$ における第 $w_{c,i}$ 番目のノードに対応する．word2vec の最適化は，以下の負の対数尤度 L_{w2v} の最小化によって実現される：

$$L_{w2v}(U, U') = -\sum_i \sum_{w_{c,i} \in C(w_i)} \log P(w_{c,i}|w_i). \tag{5}$$

最適化の結果，最適な埋め込み行列 $\hat{U} = (\hat{\boldsymbol{u}}_1, \cdots, \hat{\boldsymbol{u}}_p)$ が得られる．これを用いて，単語 w_i の特徴量が $\boldsymbol{h}_i = \hat{U}\boldsymbol{x}_i = \hat{\boldsymbol{u}}_i$ と表現される．これにより，単語の特徴抽出が実現される．

2.3　IP2vec

前述の word2vec を，単語の特徴抽出ではなく IP アドレスの特徴抽出に用いた IP2Vec [9] と呼ばれるアルゴリズムが 2017 年に提案されている．IP2Vec では word2vec に

#	送信元 IP アドレス	宛先 IP アドレス	宛先 ポート番号
1	10.***.01	20.***.02	22
2	10.***.01	10.***.70	3389
3	212.***.201	20.***.02	22
4	212.***.201	10.***.70	3389
5	5.***.101	20.***.01	22
6	5.***.101	20.***.02	22
7	5.***.101	20.***.03	22
8	5.***.101	20.***.04	22

#	送信元 IP アドレス	宛先 IP アドレス	宛先 ポート番号
1	1	4	9
2	1	5	10
3	2	4	9
4	2	5	10
5	3	6	9
6	3	4	9
7	3	7	9
8	3	8	9

w_i	$C(w_i)$
1	{4,5,9,10}
2	{4,5,9,10}
3	{4,6,7,8,9,10}

図 2 IP2Vec の記号例 ($q = 3$, $p = 10$)
通信情報として宛先 IP アドレス・ポート番号を含むと仮定

おける単語を IP アドレス，コンテクストとなる周辺語を各 IP アドレスが発したアラートの通信情報とみなす．つまり，word2vec では各単語を文脈の前後の単語で特徴付けるのに対し，IP2Vec では各 IP アドレスをアラート情報を基に特徴付ける．word2vec と IP2Vec の対応を，以下の表 1 に示す．

表 1 word2vec と IP2Vec の比較

	w_i	$C(w_i)$
word2vec	単語	前後の数単語
IP2Vec	IP アドレス	アラートの通信情報

IP2Vec のための記号を導入する．いま，q 個の送信元 IP アドレスが存在するとする．また，送信元 IP アドレス以外の通信情報が $p - q$ 通りあるとする．従って，各送信元 IP アドレスは，w_i ($\in \{1, \cdots, q\}$)，コンテクストは $C(w_i) \subset \{q+1, \cdots, p\}$ と表記される．図 2 に，$q = 3$, $p = 10$ の例を示す．簡単のため，通信情報は宛先 IP アドレス・ポート番号のみとし，送信元・宛先 IP アドレスの集合は素であると仮定している．

ネットワーク構造と学習については，word2vec のそれらをそのまま適用することができる．つまり，各 IP アドレス w_i と，それが発したアラート情報である $C(w_i)$ が共起する確率に基づく尤度を損失関数として用いることができる：

$$L_{i2v}(U, U') = L_{w2v}(U, U') = -\sum_i \sum_{w_{c,i} \in C(w_i)} \log P(w_{c,i}|w_i). \quad (6)$$

損失関数の最小化によって最適な埋め込み行列 $\hat{U}$ が得られ，

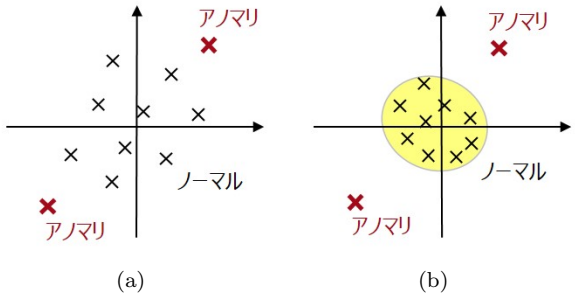


図 3 従来手法課題の概念図：従来手法による特徴抽出 (a) は，ノーマルとアノマリ IP アドレスの境界が曖昧となる．アノマリ検知では (b) のように，境界が顕著になる特徴量が望ましい．

IP アドレス w_i の特徴量を $\boldsymbol{h}_i = \hat{U}\boldsymbol{x}_i = \hat{\boldsymbol{u}}_i$ ($i \in \{1, \cdots, q\}$) と表現することが可能になる．各 $\boldsymbol{h}_i$ は，図 1 下段左の行列の各行に対応する．IP2Vec によりサイズ $q \times d$ のデザイン行列が得られ，一般には続いてこの行列にアノマリ検知アルゴリズムを適用し，アノマリスコアを算出する．

3. 提案手法

本章では，前章で議論した IP アドレスの特徴抽出技術を受け，効率的なアノマリ検知を実現する新たな特徴抽出手法を提案する．

3.1 従来手法の課題

従来手法の IP2Vec は，アラートログのような膨大なカテゴリカル変数から成るデータから，分析対象である IP アドレスの特徴抽出を実現する技術として有効である．一方で IP2Vec はあくまで特徴抽出手法であるため，アノマリ検知には IP2Vec を適用した後に機械学習のアノマリ検

知アルゴリズム等を適用する必要がある．つまり，特徴抽出とアノマリ検知を独立したステップで実行することになる．このように検知と独立して特徴抽出した場合，特徴抽出段階で検知に適した特徴量を抽出できないという課題が生じる．この課題を図 3 を用いて視覚的に示す．図は特徴量空間における IP アドレスの分布を表す．従来手法により抽出した特徴量 (図 3(a)) は，IP アドレス間の関係を適切に記述するものの，ノーマル IP アドレスとアノマリ IP アドレスの境界が曖昧になる問題が生じる．アノマリ検知の文脈では，特徴量空間でノーマルとアノマリの境界が顕著であることが望ましく，このような特徴量が検知に適していると考えられる (図 3(b))．これを実現するには，特徴抽出と検知を独立して実行するのではなく，特徴抽出段階で検知に関する評価を織り込むことが有効であると考えられる．

3.2 定式化

前小節で議論した従来技術の課題解決策として，特徴抽出の損失関数とアノマリ検知の損失関数を同時に最適化する手法を提案する．つまり，以下のような損失関数の最適化を実行する：

$$L(U, U', \boldsymbol{\theta}) = L_{i2v}(U, U') + \lambda R(U, \boldsymbol{\theta}). \tag{7}$$

第一項 $L_{i2v}(U, U')$ (式 6 と同様) が特徴抽出の損失関数，第二項 $R(U, \boldsymbol{\theta})$ がアノマリ検知の損失関数に関する正則化項である．正則化項のパラメータは $\boldsymbol{\theta} \in \Theta$ とする．二項のトレードオフは，実数 λ (> 0) で調整される．この最適化により，特徴抽出段階でアノマリ検知の損失関数を小さくするような特徴量を抽出できるようになるため，より検知に適した特徴量を抽出することが期待できる．

これを実現するために具体的に正則化項を定式化する．ここでは正則化項として，Support Vector Data Description [SVDD] [11] の損失関数に基づく正則化項を用いる．SVDD は，よく用いられる機械学習的アノマリ検知手法の一つである．データ点 $\{\boldsymbol{z}_i\}_i$ が与えられているとき，SVDD の損失関数は以下で与えられる：

$$L_{SVDD}(r, \boldsymbol{c}_0) = r^2 + C \sum_i \max\{0, \|\boldsymbol{z}_i - \boldsymbol{c}_0\|_2^2 - r^2\}. \tag{8}$$

この損失関数は，おおよそ全てのデータ点を含む最小の超球面 (半径 r, 中心 $\boldsymbol{c}_0$) を求めることを要求する．超球面外のデータがアノマリと認定される仕組みである．2 次元の場合を視覚的に表現すると，図 3(b) のような円形の判別境界を引くことに当たる．提案手法においては，SVDD におけるデータ点 $\boldsymbol{z}_i$ は，IP2Vec により抽出される特徴量 $\boldsymbol{h}_i$ に相当する．これを正則化項に用いると，抽出した特徴量が超球面内に分布するように仕向けられる．更に簡単のため，文献 [3] に倣いこれを簡略化した SVDD に基づく損失関数

$$R(U, \boldsymbol{h}_0) = \sum_i \|\boldsymbol{h}_i - \boldsymbol{h}_0\|_2^2 = \sum_i \|U\boldsymbol{x}_i - \boldsymbol{h}_0\|_2^2 \tag{9}$$

を正則化項として採用する．この最小化は，ある点 $\boldsymbol{h}_0$ を中心に特徴量 $\boldsymbol{h}_i$ が分布することを要求する．

以上より，提案手法において最小化する損失関数を以下のように定める：

$$\begin{aligned} L(U, U', \boldsymbol{h}_0) &= L_{i2v}(U, U') + \lambda R(U, \boldsymbol{h}_0) \\ &= L_{i2v}(U, U') + \lambda \sum_i \|\boldsymbol{h}_i - \boldsymbol{h}_0\|_2^2. \end{aligned} \tag{10}$$

第一項で IP アドレスの特徴量を抽出しつつ，第二項の効果で特徴量が点 $\boldsymbol{h}_0$ に集まるように仕向けられる．勿論，アノマリも $\boldsymbol{h}_0$ に寄ることになる．しかし，アノマリ検知においてアノマリは一般にごく一部であること，また共起パターン（通信情報）がノーマルとは異なるため，アノマリは $\boldsymbol{h}_0$ に集まることなく，結果外れた点に分布することになる．従って，IP アドレス w_i のアノマリスコア S_i を，最適化されたパラメータ $\hat{U}, \hat{\boldsymbol{h}}_0$ を用いて

$$S_i = \|\boldsymbol{h}_i - \hat{\boldsymbol{h}}_0\|_2^2 = \|\hat{\boldsymbol{u}}_i - \hat{\boldsymbol{h}}_0\|_2^2 \tag{11}$$

と定めることができ，この値が大きい程アノマリの度合いが強いと判断できる．本手法による検知への効果は数値実験にて検証する．

Algorithm 1 IP2Vec for Anomaly Detection

Input: $\{w_i, C(w_i)\}_i$
Output: A set of anomaly scores $\{S_i\}_i$
  Initialize $U^{(0)}, U'^{(0)}$ randomly, $\boldsymbol{h}_0^{(0)} \leftarrow \frac{1}{q}\sum_{i=1}^q \boldsymbol{h}_i^{(0)}$
  for $k = 1, \cdots, maxepoch$ **do**
    Find$(U^{(k)}, U'^{(k)})$ (Optimize e.q.(10) using mini-batch optimization, while $\boldsymbol{h}_0^{(k-1)}$ is fixed.)
    $\boldsymbol{h}_0^{(k)} \leftarrow \frac{1}{q}\sum_{i=1}^q \boldsymbol{u}_i^{(k)}$
  end for
  $(\hat{U}, \hat{\boldsymbol{h}}_0) \leftarrow (U^{(maxepoch)}, \boldsymbol{h}_0^{(maxepoch)})$
  Compute anomaly score $S_i = \|\hat{\boldsymbol{u}}_i - \hat{\boldsymbol{h}}_0\|_2^2$ for each $\{w_i\}_i$
  return $\{S_i\}_i$

3.3 最適化

提案手法の目的関数 (式 10) の最適化手順を Algorithm 1 に示す．最適化は，ネットワークパラメータ (U, U') と，正則化項にかかるパラメータ $\boldsymbol{h}_0$ の交互最適化によって実現する．まずネットワークパラメータ (U, U') をランダムに初期化し $(U^{(0)}, U'^{(0)})$，$\boldsymbol{h}_0$ を特徴量ベクトル $\{\boldsymbol{h}_i^{(0)}\}_{i=1}^q$ の重心として初期化する．各エポック k ではまず，$U^{(k)}$ 及び $U'^{(k)}$ の最適化を $\boldsymbol{h}_0^{(k-1)}$ を固定して行う．これは目的関数 (式 (10)) のミニバッチ勾配降下法に基づいて実行する．つまり，以下の更新をミニバッチ毎に行う：

表 2　データセット情報

データセット	ログ数	総 IP アドレス数	攻撃端末数
OpenStack week1	8,451,520	9,346	1
OpenStack week2	10,310,733	9,333	1
External Server week1-4	671,241	34,161	3

$$U^{(k)} = U^{(k-1)} - \eta \frac{\partial L}{\partial U} \tag{12}$$

$$U'^{(k)} = U'^{(k-1)} - \eta \frac{\partial L}{\partial U'} \tag{13}$$

パラメータ更新後，重心を $\boldsymbol{h}_0^{(k)} \leftarrow \frac{1}{q}\sum_{i=1}^{q} \boldsymbol{u}_i^{(k)}$ ($\boldsymbol{u}_i^{(k)}$: $U^{(k)}$ の第 i 列) と更新して次のエポックへ進む．この交互最適化を既定のエポック数回実行後に得られる $\hat{U}, \hat{\boldsymbol{h}}_0$ を用いて，IP アドレス $\{w_i\}_i$ のアノマリスコア $\{S_i\}_i$ を算出する．

ここで注意として，実際の最適化においては目的関数の第一項 $L_{i2v}(U, U')$ を，ほぼ等価な関数に置き換えて最適化を行う Negative Sampling という手法を踏襲する．これは softmax 関数に由来する計算量の増加を抑制する技術である．詳細は文献 [10] を参照されたい．

4. 評価実験

本章では，提案手法によるアノマリ検知の有効性を検証する．

4.1 実験データ

本実験では CIDDS-001 dataset [12] を用いる．本データセットは 2017 年に作成された，ラベル付きネットワーク型 IDS に関するデータである．本データはエミュレータ上で生成された人工的なデータであるが，一般的なビジネス環境におけるネットワークを模してデータが生成されている．通信データには，通常の従業員と攻撃者が発する通信の両方が含まれている．攻撃者は DoS 攻撃，brute force 攻撃，ping スキャン，ポートスキャンのいずれかを行う．本データセットにおけるアノマリは攻撃者端末の IP アドレスであり，それを検知できるかという観点で手法を評価する．本実験ではデータセット中の以下の 3 セットに対してそれぞれ実験を行う．

(1) OpenStack week1
(2) OpenStack week2
(3) External Server week1-4

OpenStack データでは通信トラフィックがファイアウォール内で収集され，External Server データではファイアウォール外のインターネット上のサーバで収集される．各セットの情報を表 2 に示す．

本データセットは，送信元 IP アドレスの他に 13 通りの通信情報を含む [12]．今回はその中から，送信元ポート番号，宛先 IP アドレス・ポート番号，プロトコルをコンテクスト $C(w_i)$ として用いる．

4.2 実験方法

次に実験方法について，実験設計，評価基準，比較手法に分けて説明する．

4.2.1 実験設計

実験設計に際し，サンプリングについて説明する．今回はアラートのサンプリングを行ってデータセットを成形し，そのデータセットに対してアノマリ検知を実行する．データ構造を変えないよう，サンプリングの際は攻撃に関するアラートの割合を保ちながらサンプリングを行う．サンプリングサイズは 1 万と 10 万の 2 通りで，これにより 6 通りのデータセット (データセット 3 種類×サンプリングサイズ 2 通り) が成形される．各セットに対しては 5 回のサンプリング試行を行い，後述する評価基準の平均を以て検知精度を評価する．

4.2.2 評価基準

評価基準として，アノマリ検知で一般的に用いられる Area Under Curve [AUC] を用いる．AUC は，アノマリスコアの閾値を変化させた際の False Positive [FP] と True Positive [TP] の割合に基づき描かれる ROC 曲線下の面積で定義され，$[0, 1]$ の範囲で値をとる．1 に近い程良い検知であることを示す．併せて以下で定義される precision [PRC] も評価に用いる：

$$PRC = \frac{TP}{TP + FP}. \tag{14}$$

これはアノマリと判定されたサンプルの内，真にアノマリであるサンプルの割合を表す．AUC と同様 $[0, 1]$ の範囲で値をとり，1 に近い程良い検知であることを示す．詳細には，TP Rate が 1 のときの PRC を用いる．5 回のサンプリング試行の平均 AUC と PRC で手法を比較する．

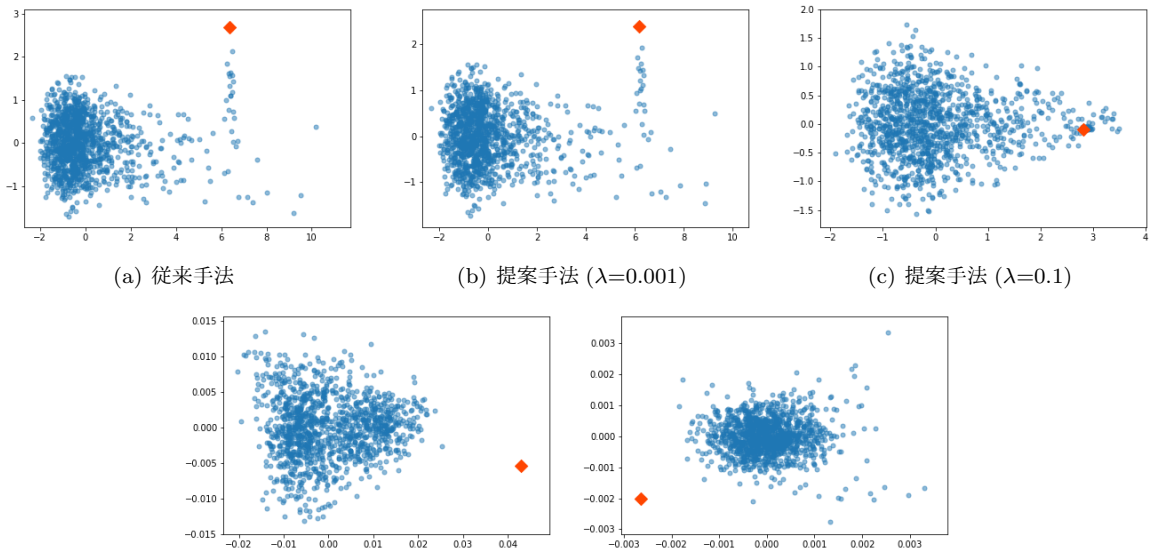
4.2.3 比較手法

比較のため，IP2Vec [9] (Ring *et al*, 2017) に，One-class SVM を合わせた従来手法を用いる．One-class SVM のカーネルにはガウシアンカーネルを用いる．カーネルのパラメータは調整の結果，Python scikit-learn パッケージのデフォルト値に従った．提案手法では正則化パラメータ λ を $\{10^{-3}, 10^{-2}, \cdots, 10^{2}, 2 \times 10^{2}\}$ と変えて実験を行う．

実装には Tensorflow を用いる．提案手法・従来手法ともに特徴量の次元は 8 次元に固定する．最適化は，モメンタム付き確率的勾配降下法 (モメンタム 0.9, 学習率 0.01, エポック数 300) を用いる．

表 3　実験結果（AUC, PRC の定義は 4.2.2 節を参照 ）

データセット	OpenStack week1				OpenStack week2				External Server week1-4			
ログ数 (IP アドレス数)	1 万 (1358.2)		10 万 (4068.4)		1 万 (1283.6)		10 万 (3855.4)		1 万 (934.0)		10 万 (6614.4)	
評価基準	AUC	PRC	AUC	PRC	AUC	PRC	AUC	PRC	AUC	PRC	AUC	PRC
IP2Vec + OCSVM	0.997	0.200	0.999	0.317	0.997	0.233	0.999	0.220	0.935	0.030	0.990	0.026
提案手法 ($\lambda = 0.001$)	0.994	0.117	0.995	0.050	0.992	0.100	0.994	0.040	**0.967**	0.049	**0.992**	**0.035**
提案手法 ($\lambda = 0.01$)	0.986	0.052	0.925	0.003	0.980	0.038	0.926	0.003	0.965	0.049	0.989	0.029
提案手法 ($\lambda = 0.1$)	0.987	0.056	0.651	0.001	0.980	0.040	0.654	0.001	0.953	0.047	0.986	0.020
提案手法 ($\lambda = 1.0$)	0.346	0.001	0.664	0.001	0.349	0.001	0.664	0.001	0.100	0.003	0.117	0.001
提案手法 ($\lambda = 10.0$)	**1.0**	**1.0**	0.930	0.003	**1.0**	**1.0**	0.853	0.002	0.956	**0.050**	0.900	0.010
提案手法 ($\lambda = 100.0$)	0.993	0.264	0.999	0.350	0.999	0.850	0.999	0.777	0.778	0.008	0.800	0.001
提案手法 ($\lambda = 200.0$)	0.996	0.327	**0.999**	**0.750**	0.999	0.900	**0.999**	**0.900**	0.588	0.005	0.789	0.002



(a) 従来手法　(b) 提案手法 (λ=0.001)　(c) 提案手法 (λ=0.1)

(d) 提案手法 (λ=10.0)　(e) 提案手法 (λ=200.0)

図 4　OpenStack week2 データセットにおける特徴量の違い．
青丸・橙色菱形がそれぞれノーマル・アノマリ IP アドレスを示す

4.3　結果

実験結果を表 3 に記す．適切な正則化パラメータ λ の調整を行うことで，全データセットにおいて提案手法が従来手法を上回る結果を得た．AUC はアノマリ IP アドレス数が少ない影響で大差はないが，PRC では大きな改善が得られた．

各データセット毎に結果を分析する．はじめにサンプリングサイズ 1 万の OpenStack データセットにおいては，week1，week2 ともに，提案手法 ($\lambda = 10.0$) が誤検知無しの完全な攻撃端末検知を実現した．1000 個以上の IP アドレス (week1: 1358.2 個, week2: 1283.6 個) の中から， 攻撃者端末の IP アドレス 1 つを検知したことは着目されたい．またサンプリングサイズが 10 万の場合でも，5 回のサンプリング試行の内 3 つないし 4 つのサンプリングセットでは，誤検知無しの検知を実現した．正則化の影響に関しては，OpenStack データセットにおいては強めの正則化が有効であることがわかる．これは本データセットが主に組織内部の通信を収集したデータであるため，攻撃者以外の通信パターンが限定的であることに起因すると考えられる．

一方で External Server データセットに対しては改善が確認できるものの，その効果に関しては限定的なものであり，正則化としては弱めの正則化が有効であることがわかる．これは本データセットの通信がインターネット上のサーバで収集されているため，通信パターンのバラエティが，OpenStack のそれよりも豊富，つまり特徴量のクラスタが複数存在することが要因として考えられる．この場合提案手法のように特徴量を一点に集中させてしまうと，クラスタ間構造が崩れ結果的に改善の効果が薄れてしまう．この対応は今後の課題である．

最後に視覚的に提案手法の効果を確認する．図 4 は，OpenStack week2 データセット（サンプリングサイズ 1 万）における，各手法による特徴量の分布パターンを比較する．可視化のため，8 次元の特徴量を PCA で 2 次元に圧縮して表示している．青丸がノーマル IP アドレス，橙色菱形がアノマリ IP アドレスを示す．この図を用いて提案手法の効果を 2 点確認する．1 点目として，正則化の強さを調整するパラメータ λ の値に応じて，抽出される特徴量

の分布が変化することである．パラメータ λ の値が大きくなるほど，特徴量がデータ点の中心に寄り，分布が円形に近づくことが確認できる．正則化が弱いとき ($\lambda = 0.001$) の分布 (図 4(b)) は従来手法による分布 (図 4(a)) と大差無いが，軸のスケールが多少変わっていることから正則化の効果は確認できる．2 点目に，特に $\lambda = 10.0$ (図 4(d)) において，提案手法がアノマリ IP アドレスの特徴量を，外れ値として検出できていることがわかる．従来手法 (図 4(a)) では曖昧であったノーマルとアノマリの境界が顕著となり，期待した正則化の効果が得られたことが確認できる．

5. まとめと今後の展望

本稿では，膨大なセキュリティアラートから潜在的脅威を発見する技術として，自然言語処理分野の word2vec を基にしたアルゴリズムを提案した．本手法は，アラートに含まれる膨大なパターンのカテゴリカル変数から，IP アドレスの特徴量を低次元かつ検知に適した形で分散表現することを可能にし，アノマリ IP アドレス検知の精度改善を実現する．数値実験では，ネットワーク型 IDS アラートから攻撃者 IP アドレスを検知するタスクにおいて，従来手法を上回る検知精度を得た．

今後の課題として，ノーマルの IP アドレスが複数のクラスタを持つようなケースへの対応があげられる．特にインターネットに接続した環境ではノーマル IP アドレスが多種になるため，判別境界を単一の超球面で求める手法では検知が限定的になる．他の課題として，時系列的な振る舞い変化の検知も挙げられる．更なる検知精度改善のため，今後本課題を検討していく．

参考文献

[1] “Survey: 27 percent of it professionals receive more than 1 million security alerts daily.” https://www.imperva.com/blog/27-percent-of-it-professionals-receive-more-than-1-million-security-alerts-daily/.

[2] R. Chalapathy, A. K. Menon, and S. Chawla, “Anomaly detection using one-class neural networks,” *ArXiv*, vol. abs/1802.06360, 2018.

[3] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, “Deep one-class classification,” in *Proc. of the 35th International Conference on Machine Learning*, vol. 80, pp. 4393–4402, Jul 2018.

[4] M.-N. Nguyen and N. A. Vien, “Scalable and interpretable one-class svms with deep learning and random fourier features,” in *Machine Learning and Knowledge Discovery in Databases*, vol. 11051, pp. 157–172, Springer International Publishing, 2019.

[5] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, “Kitsune: An ensemble of autoencoders for online network intrusion detection,” in *Proceedings of Network and Distributed System Security (NDSS) Symposium 2018*, Jan 2018.

[6] M. Du, F. Li, G. Zheng, and V. Srikumar, “Deeplog: Anomaly detection and diagnosis from system logs through deep learning,” pp. 1285–1298, 10 2017.

[7] T. Chen, L.-A. Tang, Y. Sun, Z. Chen, and K. Zhang, “Entity embedding-based anomaly detection for heterogeneous categorical events,” in *Proc. of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pp. 1396–1403, 2016.

[8] C. Bertero, M. Roy, C. Sauvanaud, and G. Tredan, “Experience report: Log mining using natural language processing and application to anomaly detection,” in *Proc. of 2017 IEEE 28th International Symposium on Software Reliability Engineering (ISSRE)*, pp. 351–360, Oct 2017.

[9] M. Ring, A. Dallmann, D. Landes, and A. Hotho, “Ip2vec: Learning similarities between ip addresses,” in *Proc. of 2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 657–666, Nov 2017.

[10] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Proc. of the 26th International Conference on Neural Information Processing Systems - Volume 2*, pp. 3111–3119, Dec 2013.

[11] D. M. Tax and R. P. Duin, “Support vector data description,” *Machine Learning*, vol. 54, pp. 45–66, Jan 2004.

[12] M. Ring, S. Wunderlich, D. Grudl, D. Landes, and A. Hotho, “Flow-based benchmark data sets for intrusion detection,” in *Proc. of the 16th European Conference on Cyber Warfare and Security (ECCWS)*, pp. 361–369, 2017.