潜在的な重要アラートの推定によるインシデント特定の効率化

芝原 俊樹[1,3,a)]　小寺 博和[1]　千葉 大紀[1]　秋山 満昭[1]　波戸 邦夫[1]　Ola Söderström[2]
Daniel Dalek[2]　村田 正幸[3]

概要：実際に成功したマルウェア感染等のインシデントを特定するために，マネージドセキュリティサービスでは，悪性な挙動（イベント）が検知された際に送信されるアラートをセキュリティオペレーションセンタ（SOC）でリアルタイムに解析している．SOC では，多くの顧客の大量のアラートからインシデントを迅速に特定するために，過去に発生したインシデントとの関係に基づいてイベントをレベル分けし，インシデントとの関連の強いイベントのアラートを優先的に解析している．しかし，この方法では，イベントに関連するインシデントが発生するまで，イベントのレベルを正確に決定できない．そこで，失敗した攻撃や誤検知に関するアラートはほとんどのイベントで観測できることに着目し，これらのアラートから抽出された攻撃の種類や誤検知のしやすさに関する特徴量に基づいて，レベル推定を実施するシステムを提案する．大規模な SOC において 1 ヶ月間で収集された約 500 万件のアラートを用いて，提案システムの有効性を評価し，提案システムで推定されたレベルが高いイベントから解析することで，インシデントの迅速な特定に貢献できることを示した．

Efficient Incident Detection by Predicting Potential Important Alerts

Toshiki Shibahara[1,3,a)]　Hirokazu Kodera[1]　Daiki Chiba[1]　Mitsuaki Akiyama[1]　Kunio Hato[1]
Ola Söderström[2]　Daniel Dalek[2]　Masayuki Murata[3]

Abstract: To detect incidents, i.e., successful attacks such as malware infection, managed security services analyze alerts regarding malicious behavior (events) in real time in a security operation center (SOC). To immediately find incidents among a large number of alerts from many customers, professional analysts in a SOC divide levels of events based on their correlation with incidents and prioritize alerts whose events are highly correlated with incidents. However, levels of events cannot be accurately determined in this way unless their related incidents occur. Therefore, by focusing on the fact that alerts of failed attacks and false positives can be observed without depending on events, we propose a system for predicting levels of events based on features regarding types of attacks and tendency of false positives. We evaluate the effectiveness of the proposed system using about 5 million alerts collected from a large-scale SOC for one month. The evaluation result shows that our system can contribute to efficient incident detection by prioritizing events whose predicted levels are high.

1. はじめに

マネージドセキュリティサービス（MSS）は，顧客の通信をセキュリティオペレーションセンタ（SOC）でリアルタイムに監視することを通じて，実際に発生したマルウェア感染やサーバの侵害をインシデントとして特定し対処することで，サイバー攻撃の被害を最小限に抑えることに貢献している [1]．MSS では，セキュリティアプライアンスや SIEM（security information and event management）に加えて専門的な知識をもつアナリストの分析によって，多くの顧客の大量の通信から少数のインシデントを効率的に特定している．まず，顧客の通信は，図 1 の左端に示すように顧客環境に置かれた UTM（unified threat management）

[1] NTT セキュアプラットフォーム研究所
NTT Secure Platform Laboratories
[2] NTT セキュリティ
NTT Security
[3] 大阪大学
Osaka University
[a)] toshiki.shibahara.de@hco.ntt.co.jp

© 2019 Information Processing Society of Japan

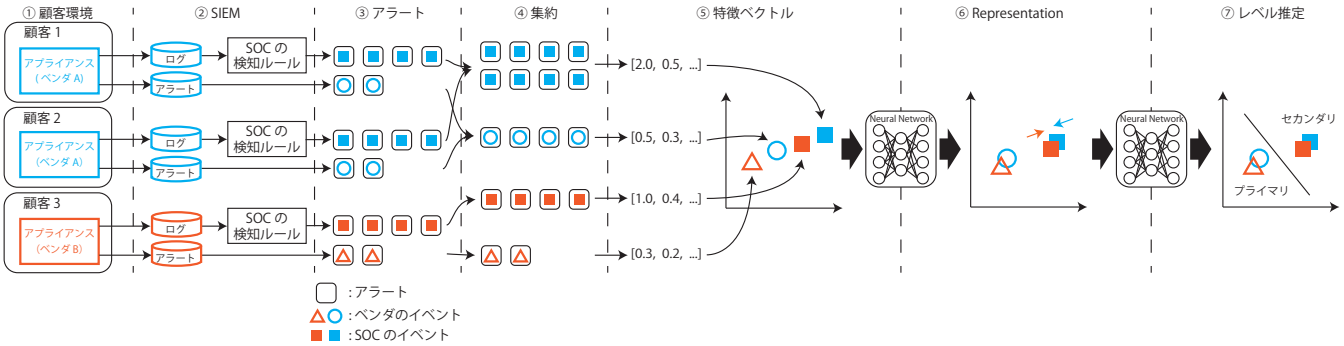

図 1 提案システムの処理の流れ

に代表されるセキュリティアプライアンスで観測され，HTTP リクエスト等の通信ログが SIEM に送信される．さらに，観測された通信は，アプライアンスのベンダが開発した検知ルールで解析され，Command and Control（C&C）サーバへの通信等の悪性な挙動（以降，イベント）が検知されると SIEM にアラートが送信される．これらの通信ログとアラートは，様々な形式のログやアラートを一元的に扱える SIEM により管理および分析が行われる．具体的には，SIEM に組み込まれた SOC で開発された検知ルールにより，すべての顧客の通信ログが解析され，イベントが検知されると SOC にアラートが送信される．すなわち，SOC に送信されるアラートには，前述のセキュリティアプライアンスが検知したアラートと SIEM が検知したアラートの 2 種類が含まれる．最後に，SOC に送信されたアラートをもとに，アナリストがインシデントが発生したかどうかを特定する．アラートには，アラートを引き起こしたイベントの種別，通信の送信元・宛先，時刻が含まれている．しかし，攻撃の初期段階のイベントや誤検知が多いイベントも存在するため，アラートを引き起こしたイベントを起点として，システムの侵害まで到達しかたや正しく検知されたイベントかどうかを手動で確認する必要がある．

インシデントへの対応および復旧を迅速に行うためには，大量のアラートからインシデントを特定する時間を可能な限り短縮する必要がある [2]．そのために，アラートを引き起こしたイベントの種別に従って，アラート解析の優先度合いを決定することで，インシデント特定までの時間を短縮している [3] ．アラートの優先度合いは，イベントごとに事前に付与されたインシデントとの関連に基づくレベルによって決定される．イベントのレベルは，複数の顧客において検知されたイベントのアラートを統合し，インシデントと特定されたアラートの割合を参照して算出される．この結果，顧客に依存しないイベントのレベルが推定され，効率化のために用いられている．本稿では，一般的なレベル分けとして，イベントの重要度が高いプライマリ，重要度が低いセカンダリに分類することを想定する．なお，イベントは攻撃固有の情報なのに対し，アラートはイベントが特定された時刻・送信元・宛先を含むものである．

新たな脆弱性，マルウェア，エクスプロイトキットが発見されると，それらを用いた攻撃を検知するための検知ルールがセキュリティアプライアンスのベンダによって開発される．その結果，アプライアンスで検知されるイベントの種類は日々増加している [4]．インシデント特定の効率化のためには，新たなイベントのレベルについても正確に分類する必要がある．しかし，従来のシステム [3] では過去に発生したインシデントとの関連に基づくため，正確に分類できない場合がある．例えば，一般的にインシデントは頻繁に発生しないため，イベントが観測され始めてから数週間または数ヶ月後に関連するインシデントが発生する場合もある．この場合，インシデントが発生した後にプライマリと分類されるイベントであっても，インシデントが発生する前には関連する過去のインシデントが存在しないため，セカンダリと分類されてしまう．本稿では，このようなイベントを潜在的なプライマリイベントと呼ぶ．SOC でのアラート解析で，潜在的なプライマリイベントが優先されないと，インシデントを特定するまでに時間がかかり，サイバー攻撃による被害が増大してしまう可能性がある．

そこで，我々は，失敗した攻撃やイベントの誤検知に関するアラートはほとんどのイベントで観測できることに着目し，潜在的なプライマリイベントを特定するシステムを提案する．プライマリイベントと攻撃の種類や誤検知が発生する確率が類似するイベントのレベルは，プライマリであると考えられる．そこで，同一イベントによって引き起こされたアラートごとの集合を作成し，攻撃の種類や誤検知のしやすさを推定するために使用する．具体的には，イベントの名称，イベントが検知された通信の送信元・宛先，アラートの時刻等から，攻撃の種類や誤検知のしやすさに関する特徴量を抽出し，教師あり機械学習でレベルを推定する．そして，セカンダリイベントの中で推定されたレベルが高いイベントを，潜在的なプライマリイベントとして特定する．この提案システムでは，過去に発生したインシデントの情報をレベルの推定に用いていないため，インシデントが発生する前であっても，レベルを正確に推定可能である．

ただし，顧客ごとに導入しているアプライアンスのベン

表 1　イベントとアラートの例

イベント名	ベンダ	送信元	宛先
Malicious communication	A	192.168.0.1	192.0.2.1
		192.168.0.2	192.0.2.2
		192.168.0.3	192.0.2.3
C2 outbound communication	A	192.168.0.4	192.0.2.2
		192.168.0.5	192.0.2.3
		192.168.0.6	192.0.2.4
Malware-related communication	B	192.168.0.7	198.51.100.1
		192.168.0.8	198.51.100.2
C2 outbound communication	B	192.168.0.9	198.51.100.2
		192.168.0.10	198.51.100.3
		192.168.0.11	198.51.100.4

ダは異なっているため，ベンダのルールによって検知されるイベントの特徴量は，ベンダごとに異なる顧客からのアラートを用いて抽出されている．このため，攻撃の種類や誤検知のしやすさが類似のイベントであっても，イベントが検知されるベンダが異なっている場合，特徴量が類似しているとは限らない．そこで，SOC のルールで検知されるイベントはすべての顧客の通信ログに適用されていることに着目し，イベントが検知されるベンダごとの特徴量の差異が，レベル推定に悪影響を与えないような最適化手法を提案する．その結果，異なるベンダのイベントもレベル推定に有効に活用されるようになり，レベル推定の精度向上が期待される．

大規模な SOC において 1 ヶ月間で収集された約 500 万件のアラートを用いて，提案システムの有効性を評価した．イベントが検知されるベンダごとの特徴量の差異が，レベル推定に悪影響を与えないように最適化することで，潜在的なプライマリイベントの特定精度を約 10%向上させられることと，提案システムで推定されたレベルが高いイベントから優先的に解析することで，インシデントの迅速な特定に貢献できることを示した．

我々の主な貢献は下記の通りである．

- 過去に発生したインシデントの情報を用いずに，イベントが検知された際のアラートを用いて潜在的なプライマリイベントを特定するシステムを提案し，大規模な SOC で収集されたアラートを用いて有効性を評価した．
- イベントが検知されるベンダごとの特徴量の差異が，レベル推定に悪影響を与えないように最適化することで，レベル推定の精度を向上させられることを示した．

2. Motivating Example

本研究で着目するイベントとアラートの例を表 1 に示す．表の各行が 1 つのアラートであり，それぞれのイベントに関して複数回アラートが検知されている．“Malicious communication” と “Malware-related communication” は，ベンダのルールで検知されたイベント（以後，ベンダのイベント）であり，それぞれベンダ A または B のみで検知されている．一方，“C2 outbound communication” は，SOC のルールで検知されたイベント（以後，SOC のイベント）であり，ベンダ A と B の両方で検知されている．

ベンダ A の “Malicious communication” が SOC のアナリストによってプライマリと分類されていたときに，“Malware-related communication” のレベルを推定する問題を考える．これらのイベントが両方とも，悪性な通信が検知された際のイベントであることはイベント名から推測することができ，イベントが検知される頻度も類似しているため，これらの観点からは，レベルはプライマリと推定される．イベントが検知された通信は，顧客のネットワークからインターネットへの通信であるため，通信の宛先に一致している IP アドレスがあれば，同じ攻撃者との通信が検知された可能性もあるため，レベルをプライマリと推定する根拠にすることができる．しかし，これらのイベントの宛先 IP アドレスには，重複しているものがないため，この観点からは，レベルはセカンダリと推定される．観点によって推定されるレベルが異なるため，異なるベンダのイベントを直接比較する方法では，確実な推定結果を出力することができない．

提案システムでは，異なるベンダのイベントの比較を行う際に，ベンダごとに異なる顧客からのアラートが収集されていることが悪影響を及ぼさないように，すべての顧客の通信ログに適用されている SOC のイベントを利用する．具体的には，SOC のイベントのアラートをベンダごとに集約し，ベンダが異なる同一イベントに関しては，レベルの推定結果が同一となるように制約を追加する．その結果，同一の SOC のイベントと類似するイベント同士は，同一のレベルが推定されるようになる．表 1 の例では，ベンダ A の “C2 outbound communication” と “Malicious communication” は，イベント名，検知頻度だけでなく，宛先の観点でも類似している．同様に，ベンダ B の “C2 outbound communication” と “Malware-related communication” もすべての観点で類似している．その結果，“Malware-related communication” のレベルが，確実に “Malicious communication” と同一のプライマリであると推定することができる．このように，レベル推定の学習時に SOC のイベントを利用することで，潜在的なプライマリイベントの特定精度を向上させることができると考えられる．

3. 提案システム

提案システムでは，図 1 に示す流れでイベントのレベルを推定し，潜在的なプライマリイベントの特定に利用する．本章では，提案システムを 4 つの部分に分けて説明する．まず，3.1 節では，SOC に送信されているアラートの収集について説明する（図 1 の①，②，③）．3.2 節では，収集されたアラートからイベントの特徴ベクトルを抽出する方

法を説明する（④，⑤）．3.3 節では，レベル推定のためのモデルの構築について説明する（⑥，⑦）．3.4 節では，推定されたレベルに基づいて潜在的なプライマリイベントを特定する方法について説明する．

3.1 アラート収集

顧客の通信はまずセキュリティアプライアンスで解析され，ベンダのルールで検知されたイベントのアラートと通信ログが SIEM に送信される（①）．アプライアンスのベンダごとに検知ルールが異なるため，検知されるイベントも異なっている．そのため，各顧客からは，使用しているアプライアンスのベンダ特有のイベントが検知される．SIEM では，SOC が開発した検知ルールがすべての顧客の通信ログに適用される（②）．その際に検知されたイベントのアラートと，顧客から送信されたアプライアンスで検知されたイベントのアラートが SOC に送信される（③）．

アラートは，イベントの情報とイベントが検知された通信に関する情報のタプルとして表現される．イベントが検知された通信に関する情報としては，一般的にアラートに含まれているタイムスタンプ，送信元，宛先を想定する．

$$alert = (event, timestamp, source, destination) \quad (1)$$

ここで，送信元と宛先は，一般的には IP アドレスで表現される．本稿では，イベントの定義として，下記 4 つの情報を用いる．

$$event = (name, direction, action, vendor) \quad (2)$$

ここで，*name* はベンダや SOC が付与したイベントの名称，*direction* は通信の方向，*action* は通信がアプライアンスによって遮断されたかどうか，*vendor* は通信が観測されたアプライアンスのベンダ名である．通信の方向は，outbound（顧客からインターネット），inbound（インターネットから顧客），internal（顧客のネットワーク内部）に分類される．イベントの定義に通信の方向を含めたのは，同じ悪性な挙動が検知されたとしても通信の方向によって深刻度が大きく異なるからである [3]．例えば，ネットワークスキャンが検知された場合，通信の方向が outbound であった場合は，マルウェアに感染したホストが顧客ネットワーク内に存在することを示唆するため深刻である．通信が遮断されたかどうかは，遮断，許可，不明に分類される．遮断されたかどうかの情報を得ることができなかったベンダのアラートは不明に分類される．イベントの定義に通信の遮断を含めたのは，通信の遮断は攻撃の失敗とも関連し，深刻度に影響するからである．イベントの定義にベンダ名を含めたのは，次節で説明するように SOC のイベントをベンダごとに集約するためである．この定義により，以降の説明を簡潔にすることができる．ベンダのイベントは，ルールを開発したベンダのアプライアンスでしか検知されないため，このイベントの定義がベンダのイベントの処理に影響することはない．

表 2 特徴量

属性	説明	型	次元数
name	キーワードの出現	Boolean	899
direction	one-hot encoding	Boolean	3
action	one-hot encoding	Boolean	3
vendor	one-hot encoding	Boolean	8
timestamp	合計検知数，バースト性	連続値	2
source	IP アドレスの出現	Boolean	1,109
	ユニーク数，バースト性	連続値	2
destination	IP アドレスの出現	Boolean	4,232
	ユニーク数，バースト性	連続値	2

3.2 特徴抽出

まず，収集されたアラートをイベントごとに集約し（④），その後アラートの集合から特徴ベクトルを抽出する（⑤）．

イベントごとのアラートの集約では，イベント $event_i$ に関するアラートの集合 $A_i = \{alert_j \mid event_j = event_i\}$ を得る．前節の定義により，SOC のイベントに関しては，ベンダごとにアラートは集約され，次節のモデル構築に利用される．

次に，アラートの集合から，イベントの攻撃の種類と誤検知のしやすさを表現する特徴ベクトルを作成する．アラートに含まれるイベントの定義とイベントが検知された通信に関する 8 つの情報からそれぞれ特徴ベクトルを抽出し，それらを結合して 1 つのベクトルを作成してイベントの特徴ベクトルとする．各属性の特徴ベクトルの作成方法と，4 章のデータセットでの特徴ベクトルの次元数を表 2 に示す．

イベント名 *name* の特徴ベクトルは，攻撃の種類を表すキーワードを使用して作成する．その際に使用するキーワードは，次の方法で決定した．まず，イベント名を記号「:()._」およびスペースで分割し，その際の部分文字列を小文字に変換する．攻撃の種類を表すキーワードは，ベンダにかかわらずイベント名に含まれていると考えられるため，複数のベンダのイベント名に含まれている部分文字列を選択する．さらに，意味のない一般的な単語である stop words および数字を除外し，残りをキーワードとする．このキーワードを用いて，ベクトルの各次元が 1 つのキーワードに対応し，イベント名に含まれているキーワードは 1，それ以外は 0 とする特徴ベクトルを作成する．

カテゴリ変数である *direction*，*action*，*vendor* は one-hot encoding でベクトル化する．具体的には，各カテゴリが各次元に対応したベクトルであり，表現したいカテゴリに対応する次元のみを 1，それ以外を 0 とするベクトルを作成する．

時系列的な特徴として，*timestamp* から合計検知数とバースト性を特徴として抽出する．攻撃の種類から想定さ

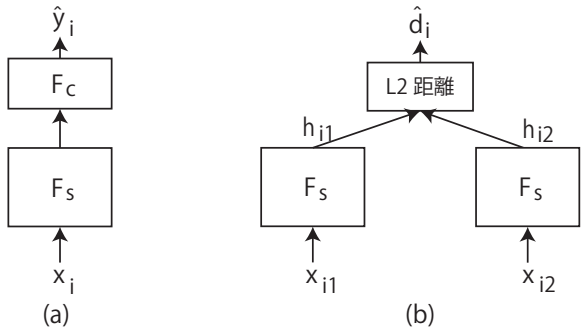

図 2 提案システムで用いられているニューラルネットワーク

れる時系列的な特徴より，合計検知数が多い場合や，バースト性が異なる場合は，誤検知がしやすいイベントと判断することができる．バースト性 [5] の算出では，アラートの収集期間を時間幅 w で区切り，その中に含まれるアラート数を計算する．バースト性 b_w を，アラート数の平均 μ_w と分散 σ_w^2 を用いて，$b_w = \frac{\sigma_w^2}{\mu_w}$ で定義する．時間幅 w としては，1 日を用いた．これは，各期間に平均して数個のアラートが含まれるようにするためである．時間幅 w を小さくしすぎてアラートが含まれない期間が多くなると，意味のある値が算出できなくなってしまう．

source と *destination* からは，IP アドレスを直接ベクトルとして表現する方法と，*timestamp* の情報も活用して，時系列的な特徴を抽出する方法で，特徴ベクトルを作成する．IP アドレスを直接ベクトルとして表現する方法では，ベクトルの各次元が 1 つの IP アドレスに対応し，*source* または *destination* に含まれている場合は 1，それ以外は 0 とするベクトルを作成する．このとき，プライベート IP アドレスは，任意に設定可能なため，グローバル IP アドレスのみを用いた．さらに，ベクトルの次元数を削減するために，レベルの推定により重要と考えられる既知のプライマリイベントの送信元および宛先のみを用いた．既知のプライマリイベントと同じ宛先または送信元が存在している場合，同じ攻撃者の通信を検知していると考えられ，レベルをプライマリと予測する根拠に使用できる．時系列的な特徴としては，ユニークな IP アドレス数と 1 日ごとのユニークな IP アドレス数のバースト性を特徴量として用いる．感染端末からのスキャンは，多くの外部の IP アドレスへの通信が発生するが，C&C サーバへの通信は少数の IP アドレスへの通信が発生するため，時系列的な特徴はレベル推定に有効であると考えられる．

上記のように各属性から抽出されたベクトルを結合して，イベントの特徴を表現する 6,260 次元のベクトルを作成する．このとき，連続値として抽出された合計回数，ユニーク数，バースト性の特徴量は，各値を log スケールに変換した．これは非常に大きな値をとるイベントが存在したためである．

3.3 モデル構築

抽出された特徴ベクトルと SOC のアナリストによって付与されたレベルを用いて，レベルを推定するモデルを構築する．提案システムでは，通常の教師あり学習でレベル推定のためのモデルを構築し（ステップ 1），その後，SOC のイベントを用いて，ベンダごとの特徴ベクトルの差異がレベル推定に影響しにくくなるようにモデルを調整する（ステップ 2）．先にレベル推定の学習を実施するのは，SOC のイベントを用いた調整がレベル推定に影響しすぎることを防ぐためである．上述のような複数の観点を考慮したモデルは，ニューラルネットワークを複数の損失を考慮して最適化することで構築できることが知られている [6]．そこで，提案システムのモデルもニューラルネットワークで構築する．

ステップ 1 イベントの特徴ベクトル $\mathbf{x}$ と，SOC のアナリストによって付与されたレベル y で構成されるレベル推定データセット $\mathbf{X}_c = \{(\mathbf{x}_i, y_i)\}$ を学習に用いる．レベルは，プライマリの場合 $y = 1$，セカンダリの場合 $y = 0$ とする．ニューラルネットワークは，図 2(a) に示すように，入力に近い層 F_s と，出力に近い層 F_c で構成されるネットワークを用いる．F_s と F_c は，それぞれ多層のニューラルネットワークである．このニューラルネットワークで予測されたレベルは，$\hat{y}_i = F_c(F_s(\mathbf{x}_i))$ となる．

学習では，レベルは 2 値で与えられているため，2 値分類精度が向上するように，クロスエントロピーで損失関数を定義する．

$$\mathcal{L}_c = -\frac{1}{N_c}\sum_{\mathbf{X}_c}(y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)) \quad (3)$$

ここで，N_c は，レベル推定データセットのデータ数である．この損失関数が小さくなるようにニューラルネットワーク F_s と F_c のパラメータ $\boldsymbol{\theta}_s$ と $\boldsymbol{\theta}_c$ を更新する．

$$\min_{\boldsymbol{\theta}_{F_s}, \boldsymbol{\theta}_{F_c}} \mathcal{L}_c \quad (4)$$

SOC では，潜在的なプライマリイベントはセカンダリとラベルづけされているため，正確なラベルではない．しかし，セカンダリイベントと比較して，潜在的なプライマリイベントは少数なため，4.2 節で示す通り問題なくモデルの学習を実施可能である．

ステップ 2 ベンダごとの特徴ベクトルの差異が分類に影響しにくくなるようにモデルを調整する．特徴ベクトルにはベンダごとの差異が明らかに含まれているが，F_s の出力（以後，representation）に，ベンダごとの差異が含まれていなければ，レベル推定には影響していないことになる．そこで，図 2(b) に示すネットワークを用いて，representation にベンダごとの差異が含まれないように制約を追加することでモデルを調整する．このようなネットワークは Siamese network [7] と呼ばれ，類似および非類

似データの組が与えられたときに，類似関係を反映した低次元空間を学習するために用いられている．

イベントの定義のうち *vendor* のみが異なる SOC のイベントの組合わせの集合を，異なるベンダの SOC イベントデータセット $X_{diff} = \{(\mathbf{x}_{i1}, \mathbf{x}_{i2})\}$ として用意する．これらの特徴ベクトルは，ベンダごとの差が含まれているため，他のイベントの定義（イベント名，通信の方向，通信の遮断）が同一であっても，特徴ベクトルは同じ値にならない（$\mathbf{x}_{i1} \neq \mathbf{x}_{i2}$）．そのため，これらの representation $\mathbf{h}_{i1} = F_s(\mathbf{x}_{i1})$ と $\mathbf{h}_{i2} = F_s(\mathbf{x}_{i2})$ も，一般的には，異なる値となる（$\mathbf{h}_{i1} \neq \mathbf{h}_{i2}$）．representation が一致する場合，ベンダごとの差が含まれていないことになるため，representation の L2 距離を損失として用いる．

$$\mathcal{L}_{diff} = \frac{1}{N_{diff}} \sum_{\mathbf{X}_{diff}} ||\mathbf{h}_{i1} - \mathbf{h}_{i2}||_2^2 \tag{5}$$

ここで，N_{diff} は，異なるベンダの SOC イベントデータセットのデータ数である．

$\mathcal{L}_{diff}$ の最小化は，representation のノルムを小さくするという自明な解に陥ってしまう可能性がある．しかし，この調整で，異なるベンダ間の representation の関係性は調整されることを想定しているが，同一ベンダのイベント間の representation の関係性には影響すべきでない．そこで，$\mathcal{L}_{diff}$ に加えて，同じベンダの SOC のイベントの representation での距離を保存するように制約を追加する．そのために，同じベンダの SOC イベントの組み合わせ $(\mathbf{x}_{i1}, \mathbf{x}_{i2})$ のステップ 1 終了時の representation での L2 距離を d_i として，同一ベンダの SOC イベントデータセット $\mathbf{X}_{same} = \{(\mathbf{x}_{i1}, \mathbf{x}_{i2}, d_i)\}$ を用意する．ステップ 2 学習時の $(\mathbf{x}_{i1}, \mathbf{x}_{i2})$ の L2 距離を $\hat{d}_i = ||\mathbf{h}_{i1} - \mathbf{h}_{i2}||_2^2$ として，d_i との差を損失とする．

$$\mathcal{L}_{same} = \frac{1}{N_{same}} \sum_{\mathbf{X}_{same}} |\hat{d}_i - d_i| \tag{6}$$

さらに，レベル推定に悪影響を与えないように調整するために，これらの損失と $\mathcal{L}_c$ を考慮して，ネットワーク全体の最適化を実施する．

$$\min_{\boldsymbol{\theta}_{F_s}, \boldsymbol{\theta}_{F_c}} \mathcal{L}_c + \alpha\mathcal{L}_{diff} + \beta\mathcal{L}_{same} \tag{7}$$

ここで，α, β はそれぞれの損失の影響を調整するハイパーパラメータである．これらは，クロスバリデーションで調整することができる．

3.4 潜在的なプライマリイベントの特定

構築したモデルを用いてセカンダリイベントから，潜在的なプライマリイベントを特定する．具体的には，セカンダリイベントのレベルを推定し，比較的レベルが高いイベントを潜在的なプライマリイベントに選択した．

セカンダリイベントの集合を SE，イベント $event_i$ の特徴ベクトルを $\mathbf{x}_i$，予測されたレベルを $\hat{y}_i = F_c(F_s(\mathbf{x}_i))$ とする．予測されたレベルは連続値で出力されるため，閾値 t より大きいイベントの集合を，潜在的なプライマリイベントの集合 $PPE = \{event_i \mid \hat{y}_i > t, event_i \in SE\}$ とする．

表 3 ベンダごとのアラート・イベント数

	ベンダのイベント		SOC のイベント	
ベンダ	アラート数	イベント数	アラート数	イベント数
A	1,904,156	1,786	10,906	83
B	1,558,188	5,805	228,539	146
C	751,270	510	168	9
D	193,580	168	3,404	26
E	192,388	318	13,498	65
F	7,652	7	55,352	85
G	5,835	44	85	3
H	14	2	724	30

4. 評価

大規模な SOC で収集されたアラートと，専門的な知識をもつアナリストによって付与されたレベルを用いて，提案システムの評価を実施した．実験は，8-core CPU および 32-GB RAM の Ubuntu マシンを用いて行った．プログラムは，scikit-learn と Keras を用いて実装した．

4.1 実験設定

データセット 2019 年 1 月のアラートを用いてデータセットを作成した．合計のアラート数は 4,925,759，ユニークなイベント数は 9,087，ベンダ数は 8 である．ベンダごとのアラート数とユニークなイベント数を表 3 に示す．SOC のイベントは，観測されたアプライアンスのベンダごとに集計した．プライバシーを侵害しないために，顧客を特定する情報は収集しなかった．

レベル推定データセット $\mathbf{X}_c$ としては，すべてのベンダのイベントを用い，データセットのサイズは，8,640 となった．ベンダごとの特徴ベクトルの差異がレベル推定に影響しにくくするために使用する異なるベンダの SOC イベントデータセット $\mathbf{X}_{diff}$ のサイズはで 382 あり，同一ベンダの SOC イベントデータセット $\mathbf{X}_{same}$ のサイズは 8,141 となった．SOC のイベントには，セキュリティアプライアンスで検知されるイベントの検知結果に基づくものも存在していたが，それらはこれらのデータセットからは除外した．

ハイパーパラメータ最適化 ハイパーパラメータは，データセット全体でのクロスバリデーションで決定した．選択された提案システムのパラメータは，$\alpha = 0.1$ と $\beta = 0.01$ となった．ニューラルネットワーク F_s の構成は，7 層のニューラルネットワークであり，1，4，6 層目は活性化関数が ReLU (rectified linear units) で出力のニューロン数が 100 の全結合層，2 層目が batch normalization 層，3，5，7 層目が Dropout となった．ニューラルネットワーク F_c の構成は，1 層のニューラルネットワークであり，1 層目は活性化関数が sigmoid で出力のニューロン数が 1 の全

結合層となった．すべての全結合層には L2 正則化が適用されており，Dropout の割合は 0.1 とした．バッチサイズは 100，3.3 節のステップ 1 およびステップ 2 のエポック数は 10，最適化手法は Adam を使用した．

4.2 実験結果

提案システムの有効性を評価するために 3 つの実験を実施した．最初は，モデルの学習が想定どおり行われているかを評価するために，representation の分析を実施した．次に，潜在的なプライマリイベントの特定精度を評価した．最後に，特定された潜在的なプライマリイベントがインシデント特定に有効であるかを調査した．実験は異なるランダムシードで 10 回実験を行い，それらの平均値と標準偏差を示す．

学習による representation の変化 3.3 節のステップ 2 の最適化によって，representation が想定どおり変化したかを調査した．ステップ 2 では，同じベンダの SOC イベントの距離を保ちながら，ベンダが異なる SOC イベントの距離を小さくする．そこで，ステップ 2 の前後でのこれらの割合の変化を調査した．割合は下記の通り算出した．

$$\frac{\frac{1}{N_{same}}\sum_{\mathbf{x}_{same}}||\mathbf{h}_{i1}-\mathbf{h}_{i2}||_2^2}{\frac{1}{N_{diff}}\sum_{\mathbf{x}_{diff}}||\mathbf{h}_{j1}-\mathbf{h}_{j2}||_2^2} \tag{8}$$

ステップ 1 終了時には，上記指標は 1.127 ± 0.12 で，ステップ 2 では 2.16 ± 0.63 であった．値が増加しているため，ステップ 2 の最適化によって，想定通り representation の最適化が実施されたことが分かる．

潜在的なプライマリイベントの特定精度 提案のシステムによって，潜在的なプライマリイベントの推定をどの程度正確に実施可能か調査した．潜在的なプライマリイベントとして，SOC のアナリストによって 2019 年 1 月には，セカンダリと判定されていたが，2019 年 5 月にはプライマリと判定されていたイベントを使用した．ただし，2019 年 5 月に見逃されているプライマリイベントも存在していると考えられるため，実際の検知率は本節の評価結果よりも高くなることが想定される．SOC のイベントに基づく最適化の効果も調査するため，レベル推定のみを最適化した場合とも比較した．比較手法としては，提案と同じ構造のニューラルネットワーク（NN）と，従来型の機械学習としてランダムフォレスト（RF）を用いた．

提案システムでは，推定されたレベルが閾値以上のイベントを潜在的なプライマリイベントとする．本評価では，閾値に依存しない評価指標として AUC と ROC カーブを用いた．AUC の値は，提案の最適化手法が 0.91 ± 0.02，NN が 0.83 ± 0.05，RF が 0.73 ± 0.02 となり，提案手法が最も高くなった．より詳細に比較するために，図 3 に誤検知が低い範囲の ROC カーブを示す．この ROC カーブから，誤検知率が 0.05 を超える場合に提案手法の検知率が他の手法に比べて高くなることが分かる．

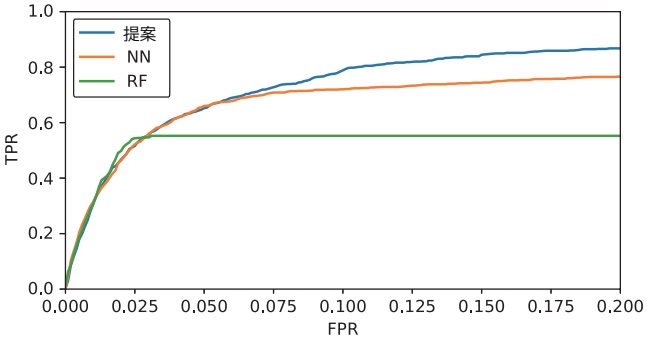

図 3 ROC カーブ

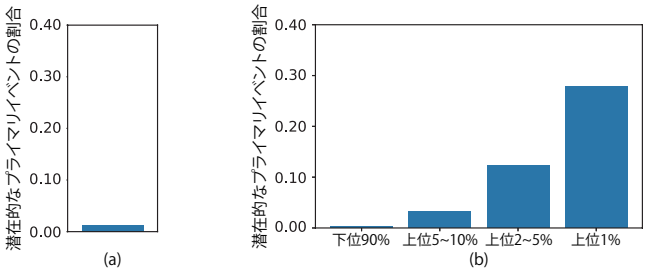

図 4 潜在的なプライマリイベントの割合

インシデント特定における効果 SOC によって決定されたレベルだけでなく，提案システムによって推定されたレベルを用いることで，インシデント特定の効率化に効果があるか調査した．インシデント特定にどの程度時間がかかるかは，インシデントと関連の強いイベントを優先的に解析できるかに依存している．SOC によって付与されたレベルがセカンダリのイベント全体からインシデントを特定する場合，図 4(a) に示すように，インシデントとの関連の強い潜在的なプライマリイベントの割合は非常に低いため，非常に時間がかかってしまう．そこで，提案システムでセカンダリイベントのレベルがプライマリである確率を出力し，その確率が高いイベントから解析する場合にインシデント特定を効率化できるかを考察する．図 4(b) は，確率の大きい順にイベントを並べた際に，どの範囲に潜在的なプライマリイベントがどの程度含まれているかを示している．この図から，確率が上位のイベントには，潜在的なプライマリイベントが多く含まれているため，確率が上位のイベントから解析することで，多くのインシデントを特定する時間を削減可能であると考えられる．

5. 議論

アラート収集 提案システムの評価では，特徴ベクトルを抽出するために，1 ヶ月間アラートを収集した．特徴ベクトルには，統計的な値や，集合として表現されている特徴も含まれているため，高精度にレベル推定を実施するためには，1 ヶ月間程度の期間がアラート収集に必要だと考えられる．本稿の評価では，この条件でも SOC のアナリス

トより早く潜在的なプライマリイベントを特定することができることを示した．つまり，提案システムではアラート収集に時間が必要であるが，目的は十分に達成可能である．

モデル構築 提案システムは，SOC のイベントを基準として，representation を調整している．SOC のイベントがベンダのイベントに比べて数が少ないことを考慮すると，イベント全体の representation を調整できていない可能性がある．提案システムの最適化は，局所的影響しかないかもしれないが，本稿では潜在的なプライマリイベントの特定性能を向上させるためには，十分であることを示した．

6. 関連研究

アラート解析 SOC でのアラート解析およびインシデント特定の効率化のために，多くのシステムが提案されている．最も関連しているシステムは，過去のインシデントとイベントの関連性から，発見されていないインシデントを特定するシステムである [3]．このシステムでは，イベントとホストを頂点とし，あるイベントがあるホストに関連する通信で検知された場合に，それらの頂点間にエッジを作成する方法で 2 部グラフを作成する．このグラフを用いて，インシデントが発生したホストからランダムウォークを開始し，到着確率でインシデントとの関連の強さが推定される．関連が強いホストが発見されていないインシデント，関連が強いイベントがプライマリイベントとして特定される．このシステムは，過去のインシデントとの関連に基づく手法であるが，提案システムでは，イベントが検知された際のアラートに基づいてイベントのレベルを推定している．

他のアラート解析効率化のためのシステムとしては，関連するアラートをメタアラートとして集約するシステムが提案されている [8], [9]．これらのシステムでは，複数のアラートの順序関係等を考慮してインシデントが発生しているかどうかを判断する際の手間の削減をしているが，イベントのレベルの推定は実施していない．

Metric Learning 提案システムでは，SOC のイベント間の距離を損失として representation の調整を実施した．このような学習は，metric learning と呼ばれている．最も基本的な手法が，提案システムに応用した Siamese network [7] である．データ間の距離を直接最適化するのではなく，サンプル間の距離の大小関係を保存するように最適化する triplet network [10] も提案されている．この手法は，教師データとしてデータ間の距離を与える必要がないため，データセットの作成が容易なことが利点である．提案システムでは，SOC イベント間の距離が使用可能であったため，Siamese network を応用した．

7. おわりに

インシデントが実際に発生する前であっても，その兆候を示す様々なイベント（攻撃の失敗や誤検知等）は発生している．そこで，イベントのインシデントに対する関連性の強さ（レベル）に関して，潜在的に発生しうるインシデントに対するレベルについても正確に推定するシステムを提案した．具体的には，多くの顧客から収集されたアラートから，攻撃の種類や誤検知のしやすさに関する特徴量を抽出して，教師あり学習でレベルを推定し，セカンダリイベントの中で推定されたレベルが高いイベントを潜在的なプライマリイベントとして特定する．さらに，イベントが検知されるベンダごとの特徴量の差異が，レベル推定に悪影響を与えないような最適化手法を提案した．大規模な SOC において 1 ヶ月間で収集された約 500 万件のアラートを用いた評価で，提案システムの最適化によって representation が想定どおりに変化していること，レベル推定を高精度に実施可能であること，インシデントの迅速な特定に貢献できることを示した．

参考文献

[1] Symantec Corporation: Symantec Managed Security Services (2019). https://www.symantec.com/services/cyber-security-services/managed-security-services.

[2] Oprea, A., Li, Z., Norris, R. and Bowers, K.: MADE: Security Analytics for Enterprise Threat Detection, *Proceedings of the 34th Annual Computer Security Applications Conference*, pp. 124–136 (2018).

[3] Roundy, K. A., Tamersoy, A., Spertus, M., Hart, M., Kats, D., Dell'Amico, M. and Scott, R.: Smoke detector: cross-product intrusion detection with weak indicators, *Proceedings of the 33rd Annual Computer Security Applications Conference*, pp. 200–211 (2017).

[4] Stock, B., Livshits, B. and Zorn, B.: Kizzle: A signature compiler for exploit kits, *International Conference on Dependable Systems and Networks* (2015).

[5] Fano, U.: Ionization Yield of Radiations. II. The Fluctuations of the Number of Ions, *Phys. Rev.*, Vol. 72, pp. 26–29 (1947).

[6] Bousmalis, K., Trigeorgis, G., Silberman, N., Krishnan, D. and Erhan, D.: Domain separation networks, *Advances in neural information processing systems*, pp. 343–351 (2016).

[7] Bromley, J., Guyon, I., LeCun, Y., Säckinger, E. and Shah, R.: Signature verification using a" siamese" time delay neural network, *Advances in neural information processing systems*, pp. 737–744 (1994).

[8] Valeur, F., Vigna, G., Kruegel, C. and Kemmerer, R. A.: Comprehensive approach to intrusion detection alert correlation, *IEEE Transactions on dependable and secure computing*, Vol. 1, No. 3, pp. 146–169 (2004).

[9] Sadighian, A., Zargar, S. T., Fernandez, J. M. and Lemay, A.: Semantic-based context-aware alert fusion for distributed Intrusion Detection Systems, *Proceedings of the 2013 International Conference on Risks and Security of Internet and Systems*, pp. 1–6 (2013).

[10] Hoffer, E. and Ailon, N.: Deep metric learning using triplet network, *International Workshop on Similarity-Based Pattern Recognition*, pp. 84–92 (2015).