LSTM-RNNを用いたインターネットバックボーン汎用異常検知手法GAMPALの改良

和久井 拓[1,a)]　近藤 賢郎[1,2,b)]　寺岡 文男[3,c)]

概要：インターネットの安定運用のために，著者らはインターネットバックボーンの汎用異常検知手法 GAMPAL（General-purpose Anomaly detection Mechanism using Prefix Aggregate without Labeled data）を提案している．GAMPAL は BGP を考慮し経路情報に基づいてフローを集約する．集約したフロー群毎に LSTM-RNN でトラフィックの挙動を予測し，実測値と比較して異常を検知する．既存方式では国内インターネットバックボーン WIDE（AS2500）内の 1 つのエッジ拠点で収集されたデータについて 1 日単位での異常検知性能を確認した．本稿では GAMPAL を AS2500 外で観測されたデータセットにも適用し環境に対する汎用性を確認した．また AS2500 内のエッジ拠点とコア拠点で収集されたデータを用い異常検知性能を比較し評価した．さらに既存研究では実測と予測の比較に 1 日のデータを用いたが，本稿では 1 時間に短縮し異常を評価する指標値を 5 分毎に更新することで，GAMPAL による異常検知のリアルタイム性の改良が可能であると確認した．

キーワード：ネットワークトラフィック分析, 汎用異常検知, インターネットバックボーン,LSTM-RNN

Improvement of GAMPAL: A General-purpose Anomaly Detection Mechanism for Internet Backbone using LSTM-RNN

Taku Wakui[1,a)]　Takao Kondo[1,2,b)]　Fumio Teraoka[3,c)]

Abstract: The authors proposed GAMPAL (General-purpose Anomaly Detection Mechanism using Prefix Aggregate without Labeled data), a general-purpose anomaly detection mechanism for the Internet backbone. GAMPAL aggregates flows based on routing information of BGP. For each aggregated flow, GAMPAL predicts the behavior of the traffic by LSTM-RNN and compares the predicted and observed values to detect anomalies. Our past research confirmed the daily anomaly detection performance of the data collected at an edge NOC in WIDE backbone (AS2500). This paper applies GAMPAL to the dataset observed outside of AS2500 for confirming its versatility. This paper also compares evaluation results at edge and core NOCs in the AS2500. Furthermore, while past research used one day's data for comparison of observed and predicted data, this paper confirms that GAMPAL can improve the performance of real-time anomaly detection by shortening the time to one hour and updating the evaluation value every five minutes.

Keywords: Network Traffic Analysis, General-Purpose Anomaly Detection, Internet Backbone, LSTM-RNN

[1] 慶應義塾大学大学院理工学研究科
Graduate School of Science and Technology, Keio University
[2] 慶應義塾インフォメーションテクノロジーセンター
Information Technology Center, Keio University
[3] 慶應義塾大学理工学部
Faculty of Science and Technology, Keio University
[a)] dona@inl.ics.keio.ac.jp
[b)] latte@itc.keio.ac.jp
[c)] tera@keio.jp

1. はじめに

様々なネットワークを相互に接続するインターネットバックボーンでは，様々なユーザやサービスのインターネットトラフィックが観測される．そのためインターネットバックボーンにおけるトラフィック時系列情報は平常時

© 2020 Information Processing Society of Japan

でも変動が大きく不規則な特徴を持つ．一方で，インターネットバックボーンで発生する異常事象は，ネットワークを構成する機器の故障や攻撃活動，イベントに起因するものなど多岐にわたる．インターネットの安定運用のために，オペレータはこれらの異常事象を早期に発見し対処する必要があるが，トラフィックパターンから異常事象を発見することは困難である．そこで，インターネットトラフィック情報から多種多様な異常事象を検知する汎用的な手法が必要である．

そこで著者らは，インターネットバックボーンにおける汎用異常検知手法 GAMPAL (General-purpose Anomaly detection Mechanism using Prefix Aggregate without Labeled data) [1] を提案している．GAMPAL は検知対象とする環境で観測されたインターネットトラフィックの時系列データをもとに，LSTM-RNN (Long Short-Term Memory Recurrent Neural Network) を用いた予測モデルでトラフィックを予測する．

観測されたデータをスケーラブルに分析するため，GAMPAL では BGP (Border Gateway Protocol) の経路表である RIB (Routing Information Base) を用い，AS パスの観測点側から 3 ホップが共通するプレフィクスを PA (Prefix Aggregate) と定義した．PA 毎にフローを集約し，集約されたフロー群毎に予測モデルでインターネットトラフィックの挙動を予測し，実際に観測されたトラフィックの挙動と比較することで異常を検知する．実測データが予測データと近い挙動の場合，観測されたトラフィックが予測通りであり異常がなく，一方予測データとの差が大きい場合，過去のトラフィックでは予測できなかった異常事象が発生したとして検知する．

著者らによる既存研究 [1] では，国内のインターネットバックボーンである WIDE (AS2500)[2] 内のエッジ拠点である藤沢 NOC (Network Operation Center) で観測されたデータについて，GAMPAL の 1 日単位での異常検知性能を評価した．評価の結果，GAMPAL が外部サービスの接続障害，AS2500 を構成するネットワークの 1 つである慶應義塾大学におけるイベントに起因する異常，DDoS 攻撃の 3 種類の異常事象について検知可能であることを確認した．

本稿ではまず，GAMPAL が WIDE バックボーン以外の環境でも異常検知が可能であることを確認するため，AS2500 外で観測されたデータセットに GAMPAL を適用する．またインターネットバックボーン内の特徴の異なる観測拠点について GAMPAL を適用し異常検知性能を比較する．さらに既存研究では 1 日分の実測データと予測データを比較し 2 つのデータの差異を評価する指標を計算することで 1 日単位での異常を検知していたが，本稿では比較するデータ単位を 1 時間に短縮し，評価指標を 5 分毎に更新することでリアルタイム性の改良が可能であることを確認する．

2. 関連研究

インターネットを対象とした異常検知手法に関する研究は多く存在するが，近年特に，未知の脅威や異常事象の検知を目的とした深層学習を用いた手法が増えている．ネットワークにおける異常事象は多種多様で，それぞれ複雑な特徴を持つため，シグネチャ型の攻撃検知 [3]-[5] のように検知対象をあらかじめ定義することが困難であり，また未知の脅威への対応も困難である．こうした問題に対応することを目的として深層学習を用いた手法が数多く提案されている．そこで本章では深層学習を用いた既存研究について議論し要求事項を明らかにする．

エンタープライズ/DC (Data Center) 規模のネットワークの異常検知手法について，文献 [6] はクラウド・インターネットサービスを対象とした異常検知手法を提案している．この研究では振舞いベースの異常検知と予測ベースでの異常検知を組み合わせた適応型異常検知アプローチを提案している．文献 [7] ではエンタープライズネットワークを対象とした汎用異常検知手法を提案している．この研究では CNN (Convolutional Neural Network) による分類モデルを用いた異常検知を提案している．トラフィック情報は MCODT (Micro-Cluster Outlier Detection in Time series) というクラスタリングアルゴリズムと SOM (Self Organization Map) という次元圧縮法で可視化され，CNN で学習される．文献 [8] は SDN (Software-Defined Networking) を対象とした侵入検知手法である．この研究ではデータセットとして `NSL-KDD`[9] を用い GRU (Gated Recurrent Unit) RNN による分類モデルを学習させている．

一方インターネットスケールのネットワークを対象とした手法については，文献 [10] は P2P ネットワークにおけるボットネット検知手法である．この研究は CNN による分類モデルに決定木を組み合わせることで検知率の向上を試みている．文献 [11] はインターネットトラフィックを対象とした攻撃検知を目的とした手法を提案している．この研究では，オートエンコーダーによる教師なし学習と最近傍探索による教師あり学習を組み合わせた手法で，後者には手動でのオペレーションが必要である．

表 1 に GAMPAL と関連研究 [6]-[8][10], [11] との比較を示す．インターネットバックボーンにおける異常検知手法の要求事項として (i) インターネットに対するスケーラビリティ，(ii) 異常事象に対する汎用性，(iii) インターネットトラフィックの周期性への考慮，(iv) ラベル付きデータの要否，の 4 つが挙げられる．これらの要求事項について関連研究の手法について議論する．

深層学習を用いた手法は未知の脅威を検知できる可能性がある一方，シグネチャ型よりも計算時間がかかることが知られている．特にインターネット規模のネットワークは

観測される情報が膨大なためスケーラブルに分析することが求められる．文献 [8] はインターネットに対して小規模な SDN ベースのネットワークにおける汎用異常検知を提案している．文献 [7] もインターネットに比べ小規模なエンタープライズネットワークを対象とした異常検知を提案しておりスケーラビリティの必要性が低い．

また，インターネットバックボーンにおける異常検知では 1 章で述べたように，多種多様な異常事象に対し汎用的異常を検知する必要がある．しかし既存研究には検知対象を特定の異常事象や攻撃活動に特化した手法が多い．文献 [8] は検知するネットワークを SDN に限定している．文献 [6] はクラウドとインターネットサービスにおける異常事象に限定している．文献 [10] は検知する異常事象をボットネットに限定した異常検知手法である．

またインターネットトラフィックを時系列データとして扱う際，1 日単位や 1 週間単位といった周期性を考慮する必要がある．文献 [8] は従来の RNN に比べ長期の時系列データの学習が可能な GRU RNN を用いている．文献 [7] は時系列データに対するクラスタリングアルゴリズムである MCODT を用いている．文献 [10], [11] についてはトラフィックの周期性を考慮していない．

さらに，ネットワークトラフィックの異常検知によく用いられるラベル付きデータは，フローやパケットなどに対し 1 つずつラベル付けする必要があり，どのようなネットワーク環境でも容易に生成できるものではない． またラベル付きデータを用いた分類モデルは，データが収集された環境と全く異なる特徴を持つ環境においても同様に検知できるとは限らない．そのためラベル付きデータを用いない異常検知手法が必要である．文献 [6] はラベル付きでないウェブサービスデータを用いた適応学習によりデータの振る舞いから異常を検知する．文献 [7] もラベル付きデータを用いておらず，提案手法が検知した時刻に検知対象としているエンタープライズネットワークにおいて異常が発生しているか確認することで検知性能を評価している．文献 [11] は教師あり学習による分類にラベル付きデータを用いているが，教師なし学習においてはラベル付きデータを使用していない．

著者らの提案する GAMPAL はこれらの要求事項を全て満たしている．

表 1　関連研究の比較
Table 1　Comparison of related work.

Related work	Enterprise/DC Scale			Internet Scale		
	[6]	[7]	[8]	[10]	[11]	GAMPAL
Scalability	-	No	No	-	Yes	Yes
Versatile to the types of anomaly	No	Yes	No	No	Yes	Yes
Consideration on periodicity of traffic	-	Yes	Yes	No	No	Yes
Necessity of labeled data	No	No	Yes	Yes	Middle	No

3. GAMPAL の設計

3.1 GAMPAL の概要

GAMPAL は実測フローサイズと予測フローサイズとの比較を通じ異常を検知する．フローは始点/終点アドレス，始点/終点ポート番号，プロトコルの 5 タプルで識別され，GAMPAL ではフローのバイト数を時系列データとして扱う．フローの 5 タプルのうち始点/終点アドレスに着目したとしても，フローはアドレス数の自乗のオーダー ($O((number\_of\_addresses)^2)$) で分類される．IPv4 の場合オーダーは $O(10^{18})$ となる．この膨大な種類のフローを集約するため，まず GAMPAL ではフローの終点アドレスを BGP の経路表である RIB (Routing Information Base) の終点アドレスのプレフィクスで集約する．終点アドレスのオーダーは $O(10^9)$ であり，これをプレフィクスで集約した場合，2019 年 6 月の時点でプレフィクスは 700,000 種類以上存在するため， オーダーは $O(10^5)$ である．これをさらに集約するため，GAMPAL では PA (Prefix Aggregate) を定義し終点アドレスのプレフィクスを集約する．PA は BGP RIB の AS_PATH 属性を構成する AS 番号のうち先頭 k 個が共通するプレフィクスの集合である．WIDE バックボーンで観測される BGP RIB では，$k = 3$ のとき PA は 30,000 種に分類され，オーダーは $O(10^4)$ となる．

図 1 に GAMPAL の設計概要を示す．図 1-(a) はフローデータを PA と時間で集約した *flow size matrix* であり，n 個の PA (図 1-(b)) に集約されている．

また GAMPAL では時間的な集約について *flow size interval* (図 1-(c)) を定義する．flow size interval が 5 分のとき，5 分毎に観測されるフローのデータサイズ (バイト数) の合計を記録する．各 PA において，flow size interval ごとの区切りを *flow size aggregation slot* (図 1-(d)) とする．この flow size aggregation slot を LSTM-RNN へ入力するデータ長である *flow size learning interval* (図 1-(e)) の時間集めた時系列データを *flow size vector* (図 1-(f)) と定義する．flow size aggregation interval が 5 分で flow size learning interval が 1 日であったとき，各 PA の flow size vector は 288 個の flow size aggregation slot で構成される．また実測フローから得られた *observed flow size matrix* (図 1-(g)) は全 PA の flow size vector で構成される．

observed flow size matrix を LSTM-RNN に入力して得られた予測結果が *predicted flow size matrix* (図 1-(h)) である．図 1 中の t_x から t_{x+j} までの observed flow size matrix に基づいて予測された t_y から t_{y+j} までの predicted flow size matrix は，t_y から t_{y+j} で観測されたフローの observed flow size matrix (図 1-(l)) と比較される．比較する際，*flow size comparison window* (図 1-(m)) 単位で

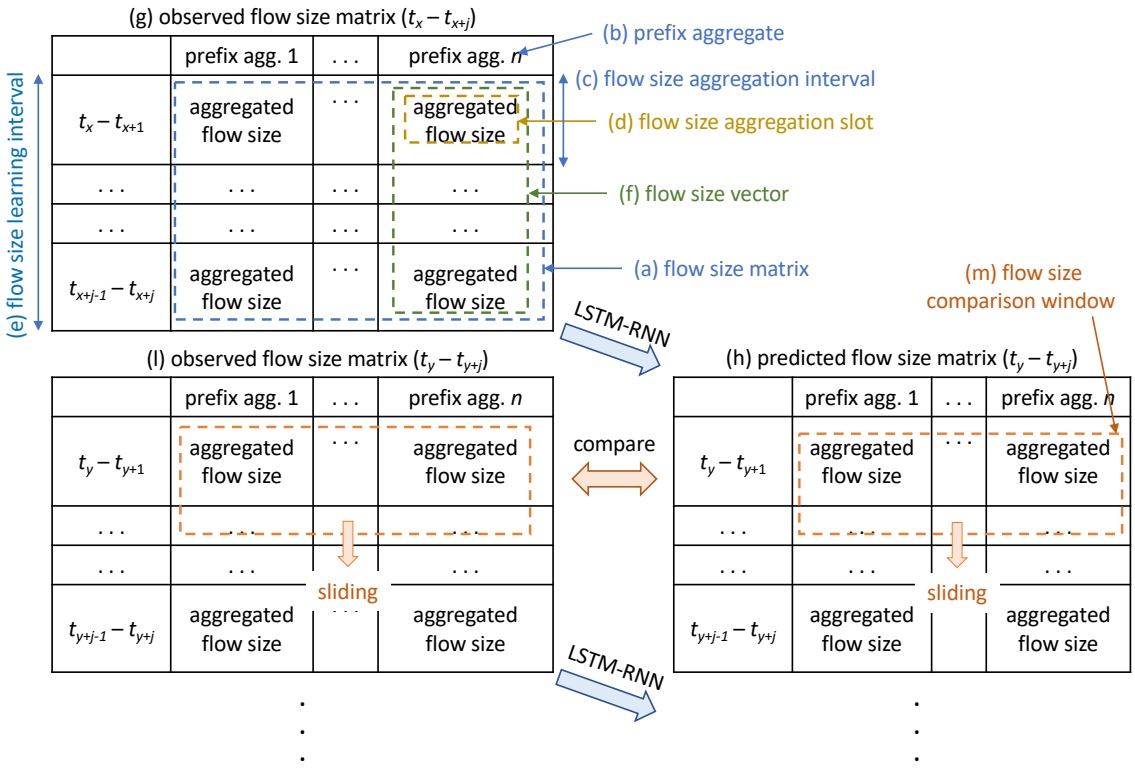

図 1　GAMPAL の設計
Fig. 1　The Implementation of GAMPAL.

observed flow size matrix と predicted flow size matrix を比較し，2 つの差異が閾値を超えるものを異常として検知する．flow size comparison window は 1 時間や 1 日 (24 時間) など，評価の単位に応じて定義する．

3.2　PA (Prefix Aggregate) の定義

3.1 節で述べたように，GAMPAL では AS_PATH 属性の先頭 k 個の AS 番号が共通する PA を定義している．本節ではこの k に適切な値の決定について議論する．

図 2 に 2019 年 6 月 22 日に AS2500 で観測された IPv4 BGP のフルルートにおける AS_PATH 長のヒストグラムを示す．最小値と最大値は 0 (iBGP) と 44 であり，最頻値は 3，中央値は 4 であった．AS_PATH 長の分布は小さい値に大きく偏り，大きい値に向かってロングテールが伸びている．このことから PA を定義する k は小さい値が適切であり，GAMPAL では最頻値である 3 を採用する．AS_PATH の先頭 3 つの AS 番号を PA の識別子と定義する．結果，IPv4 BGP フルルートで 727,261 件 (2019 年 6 月) あるアドレスは 31,258 件の PA に集約される．

インターネットには観測点から経路的に近い IP アドレス宛てのトラフィックは多く観測され，遠い IP アドレス宛てのトラフィックはあまり観測されない局所性が存在する．PA は観測点から近い終点 IP アドレスを細かく分類し，観測点から遠い終点 IP アドレスについては粗く分類することができるため，局所性の観点から適した集約方法

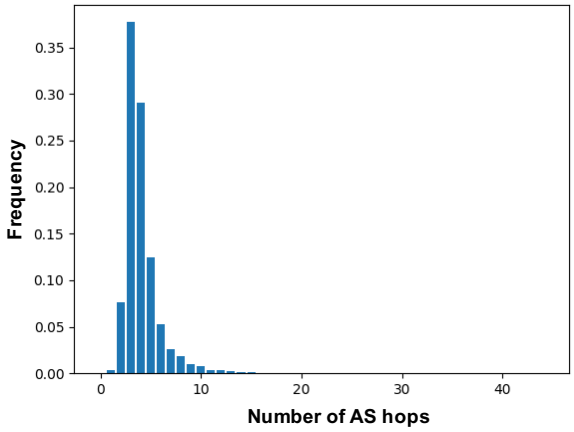

図 2　AS_PATH 長のヒストグラム
Fig. 2　Histogram of AS_PATH length.

である．

3.3　予測モデルの学習戦略

インターネットバックボーンの多くは複数のエッジ NOC とコア NOC から構成されている．インターネットトラフィックは観測される NOC 毎に，1 日単位や週単位 (曜日毎) の周期性を持っている．それらを考慮すると GAMPAL の予測モデルに対し 2 種類の学習戦略が考えられる．1 つ目は週単位学習である．これは 1 週間連続した

時系列データ，例えば日曜日から土曜日までの 7 日間のデータを用意し学習させ，翌週のデータを予測させるアプローチである．2 つ目の曜日別学習は予測対象の日付と同じ曜日のデータを過去数週分用意し学習させるものである．

我々の過去の調査において，この 2 種類の学習戦略で予測モデルを学習させ予測結果を比較した結果，週単位学習の方が高い予測精度を記録している．さらに，日本のインターネットにおけるトラフィックの挙動は曜日毎に異なり週単位の周期がある [12]．そこで GAMPAL では曜日別学習による予測モデルの学習を採用する．

3.4 本稿での改良

既存研究 [1] では，国内のインターネットバックボーンである WIDE (AS 2500) 内のエッジ拠点である藤沢 NOC で観測されたデータについて，GAMPAL の異常検知性能を評価した．検知対象は外部の事象として 2018 年 10 月 17 日に発生した YouTube の接続障害 [13] を，内部の事象として慶應義塾大学の学園祭である三田祭が開催された 2018 年 11 月 22 日を選択した．藤沢 NOC は慶應義塾大学湘南藤沢キャンパスに設置され，学園祭が開催される期間中，全キャンパスの授業が休講になるため内部イベントに起因する異常事象として三田祭を選んだ．

さらに攻撃活動として 2019 年の 6 月末から 7 月上旬にかけて発生した ARMS (Apple Remote Management Service) を用いた DDoS (Distributed Denial of Service) 攻撃の一種である UDP リフレクション攻撃 [14] を対象にした．慶應義塾大学でも DDoS 攻撃が観測され，7 月 9 日に当該 port を利用した接続を遮断し対処したため，対処の直前である 7 月 6 日から 8 日を DDoS 攻撃の検知対象とした．

予測データと実測データの差異を評価する独自指標 NSD (Normalized Summation Difference) 値 (4 章参照) を用いた評価の結果，異常が確認されていない通常日に比べ，3 種類の異常事象が発生した日付の NSD 値が大きく，GAMPAL により異常事象が検知可能であることが確認された．

本稿ではまず，GAMPAL が WIDE バックボーン以外の環境でも異常検知が可能であることを示すため，スペインの ISP (Internet Service Provider) で観測されたネットワークトラフィックデータである `UGR'16`[15] を用い評価した．またバックボーンネットワーク内に複数存在する観測点について，観測点の違いによる異常検知性能を比較するため，エッジ拠点である藤沢 NOC に加えコア拠点である大手町 NOC のデータについても異常検知性能を評価した．さらに既存研究では異常が発生した日付について異常が検出されるかを 1 日単位で評価したが，実用上では異常事象が発生した後，可能な限り早く異常を検知する必要がある．そこで，1 日分の予測データと実測データを比較していた NSD 値による評価を 1 時間分の予測データと実測データを比較し，5 分毎に 1 時間単位での NSD 値を更新することで GAMPAl のリアルタイム性を改善した．

4. 評価指標の提案

4.1 評価指標 NSD の提案

GAMPAL では単位時間 (flow aggregation interval) あたりのデータサイズを PA 毎に予測する．このとき，予測値も実測値もデータサイズのスケールは PA 毎に異なり，flow size aggregation slot に 0 ～ 数バイトしか記録されない PA がある一方で，10 万 ～ 100 万バイト程度が記録される PA もある．これらはサービスやユーザの特性によるものであるが，GAMPAL の予測値と実測値とを PA 毎に比較するには，この様な幅広いトラフィックボリュームを一律のスケールで比較可能な評価指標を定義する必要がある．そのため *MSE (Mean Square Error)* の様な指標値のスケールがデータのスケールに依存する指標値は適切ではない．また，実測値や予測値はともに，flow size aggregation interval が 0 になる，すなわち当該 PA で単位時間 (flow size aggregation interval) の間トラフィックが観測されない場合がある．そのため評価するデータに 0 を含んだ場合計算できない *RMSPE (Root Mean Square Percentage Error)* の様な指標値を用いるのは適切でない．

これらを踏まえ，既存研究 [1] では独自の指標値 *NSD (Normalized Summation of Difference)* を定義した．実測データの i 番目の値を m_i，予測データの i 番目の値を p_i，入力する予測値および実測値のデータ長 (値の数) を T としたとき，NSD は以下の式で表される．

$$NSD = \frac{\sum_{i=1}^{T} |m_i - p_i|}{\sum_{i=1}^{T} \max(m_i, p_i)} \tag{1}$$

NSD は，比較する時間単位 (flow size comparison window) 中の，予測値と実測値のうち大きい方の値の合計値に対する，2 値の差分の合計値の割合である．NSD は評価対象の値のスケールに関わらず [0,1] のスケールで計算され，また分母が 0 になることがないため 0 を含むデータでも計算が可能な評価指標である．NSD は予測値と実測値の差異の度合いを示す指標であり，予測精度を評価できる．flow size comparison window 中の予測データが実測値と同じであった場合 NSD は 0 であり，大きく異なる場合 1 に近付く．1 日単位で NSD を評価する場合，flow size comparison window は 1 日であり，flow size aggregation interval が 5 分の場合，予測値と実測値それぞれ 288 個ずつの値で計算する．NSD の値は評価のための期間，すなわち flow size comparison window に依存するため，NSD で予測精度を比較する際，1 時間単位，1 日単位といった flow size comparison window を統一させる必要がある．

4.2 NSD の定義式に関する議論

本節では，予測値と実測値との差異を評価する指標値とし

て NSD を式 1 の様に定義した理由について議論する．前節でも述べたとおり，GAMPAL で評価する予測値，実測値は PA 毎に様々なスケールの値であり，また 0 を含むことがある．MSE などを指標値とした場合，flow size aggregation interval の間に数バイトしか観測されない PA，数百バイト観測される PA，数十万バイト観測される PA とで指標値のスケールが大きく異なってしまう．また RMSPE などを指標値とした場合，0 より大きい数字を 0 で割ることになり計算が出来ない．そのためこれら 2 つの特徴に考慮した評価指標値が必要であった．

そこで提案されたのが式 1 と以下の式 2 であった．

$$\sum_{i=1}^{T} \frac{|m_i - p_i|}{\max(m_i, p_i)} \tag{2}$$

2 種類の指標値は共に，予測値，実測値の値のスケールに関わらず 0 ～ 1 のスケールで評価され，また 0 を含んでいても，分子が 0 の場合のみ分母が 0 になるため計算可能である．この 2 種類の指標値について様々な PA の値で評価した結果，式 2 は，時間軸に対する値の変化が激しい PA の場合，予測値が増減を予測できていても大きな値を記録してしまった．またこれらの式の m_i，p_i にそれぞれの値ではなく，値の変化量，すなわち $m_{i+1} - m_i$，$p_{i+1} - p_i$ を代入した式も提案されたが，変化量は正の値 (増加) だけでなく負の値 (減少) もあり，負の値を考慮していないこれら 2 種類の指標値は適切でない．そこで GAMPAL では式 1 を評価指標値として採用した．

5. 実装

5.1 実装環境

GAMPAL は OS に `Ubuntsu Server 18.04.01` を，言語は `Python 3.7.0` を使用して実装した．予測モデル LSTM-RNN の実装にはソフトウェアライブラリ `Chainer 5.1.0`[16] を使用した．トラフィック情報の変換には `nfdump version 1.6.17`[17] を，BGP RIB の変換には `bgpdump version 1.4.99.13`[18] をそれぞれ用いた．また LSTM-RNN の計算には GPU (Graphic Processing Unit) を用いた．GPU プラットフォームは `CUDA 9.0`[19] である．

5.2 1 時間単位での NSD 評価

既存研究 [1] では異常事象が発生した日について予測値と実測値とを 1 日単位で比較し NSD 値を計算した．しかし実用上では異常事象が発生した後，可能な限り早く異常を検知する必要がある．すなわち，NSD 値による評価を 1 日分より短い flow size comparison window での評価が必要である．

そこで本稿では，1 時間分の予測データと実測データを比較し NSD 値を計算し評価した．本稿では flow size aggregation interval は 5 分であるため，予測，実測それぞれ 12 個の flow size aggregation slot を比較する．例えば，ある日付の 0:00 ～ 0:59 の 1 時間の NSD 値は 1 番目 (00:00 ～ 00:04 の flow size aggregation slot) から 12 番目 (00:55 ～ 00:59 の flow size aggregation slot) までの 12 個の値を用い NSD 値を計算し，次の NSD 値は 2 番目から 13 番目までの値で計算する．この様に，flow size comparison window を 1 つずつスライドさせていくことで，5 分毎に 1 時間単位での NSD 値を更新することができる．

6. 評価

6.1 UGR'16 における異常検知

GAMPAL の環境に対する汎用性を検証するため，本稿では UGR'16 で観測されたトラフィックデータを用いる．UGR'16 はラベル付きデータであるが，本稿では異常検知の対象とする日付の決定と，異常事象のデータを学習データから排除する目的にのみラベルを使用した．

本稿は UGR'16 のデータから，異常日として 2016 年 4 月 14 日と 6 月 20 日を選択した．前者は 4 月中旬に確認された SSH スキャン攻撃を検知するために選択した．また，この異常日と比較するため，この異常日の直後の同じ曜日である 4 月 21 日とその直後の日付である 22，23 日を通常日として選択した．なお 4 月中旬に攻撃が確認されたのは 14 日のみではないが，データの欠損により評価が取れなかったため 4 月 14 日のみを SSH スキャン攻撃異常日として選択した．同様の理由で通常日として異常日の直前の日付は選択していない．一方，後者の 6 月 20 日は不審なスパムが多く観測された日付である．この異常日と比較するため，この異常日の直前の同じ曜日である 6 月 13 日とそれらの直後の日付である 17，18，24，25 日を通常日として選択した．UGR'16 は PA による集約に用いる経路情報を提供していないため，RIPE NCC (Network Coordination Center) [20] のスペインで収集された経路情報 rrc18.ripe.net を使用した．

図 3 に SSH スキャン攻撃のあった 4 月 14 日の NSD 値を，**図 4** にスパム攻撃があった 6 月 20 日の NSD 値をそれぞれ示す．通常日として選択した複数の日付の NSD 値は平均と分散を示した．SSH スキャン攻撃のあった日付については通常日より 0.03 程度，スパム攻撃があった日付については通常日より 0.02 程度大きい値が記録された．この結果は WIDE バックボーンと同様に，GAMPAL で複数種類の異常の検知が可能なことを示している．

6.2 WIDE バックボーンにおける拠点毎の評価比較

インターネットバックボーンは複数の NOC で構成されている．GAMPAL はインターネットバックボーンを対象とした異常検知手法であるため，複数の NOC について検証する必要がある．そこで本節では，WIDE バックボーン内の性質の異なる 2 つの NOC について GAMPAL による

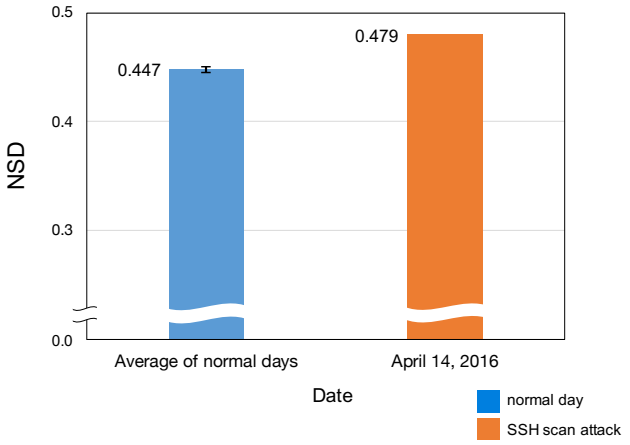

図 3　UGR'16 における SSH スキャン攻撃の検知
Fig. 3　Result of evaluation of SSH scan attack in UGR'16.

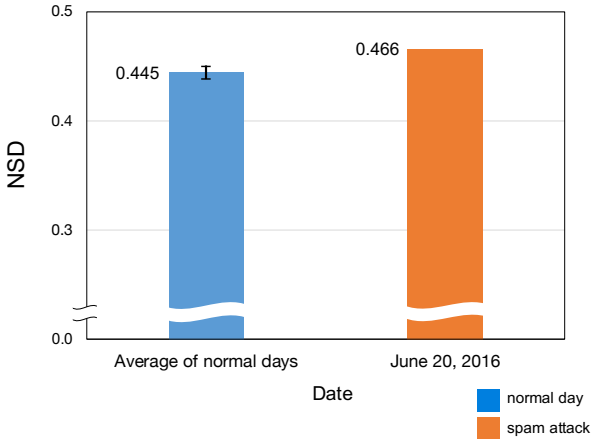

図 4　UGR'16 におけるスパム攻撃の検知
Fig. 4　Result of evaluation of spam attack in UGR'16.

異常検知性能を評価し比較する．既存研究 [1] では WIDE バックボーンの藤沢 NOC で観測されたデータで異常検知性能を評価していたが，本稿ではそれに加えて大手町 NOC についても同様に評価し比較する．藤沢 NOC は WIDE バックボーンのトポロジにおいて末端に位置するエッジ拠点であり，大手町 NOC はより中央に近いコア拠点である．

図 5 に 2 拠点における YouTube 接続障害発生が発生した 10 月 17 日の NSD 値を示す．NSD 値のスケールは NOC 毎に異なっている．これは予測精度がトラフィック量や PA の数などの特徴に依存するためである．LSTM-RNN の設計は全ての PA について共通しているため，トラフィックが多く流れ，流量の増減が大きい PA は，流量の変化が少ない PA よりも一般に予測が困難になる．そのため流量の多い観測点においては流量が少ない観測点と比べ予測が困難になり，NSD 値が大きくなる．YouTube の接続障害が発生した 2018 年には藤沢 NOC では 1 時間あたり 130,000 程度のフローが観測される．一方大手町 NOC では 100,000 程度のフローが観測されていた．この NOC 毎の流量の差が NSD 値に影響を与えている．

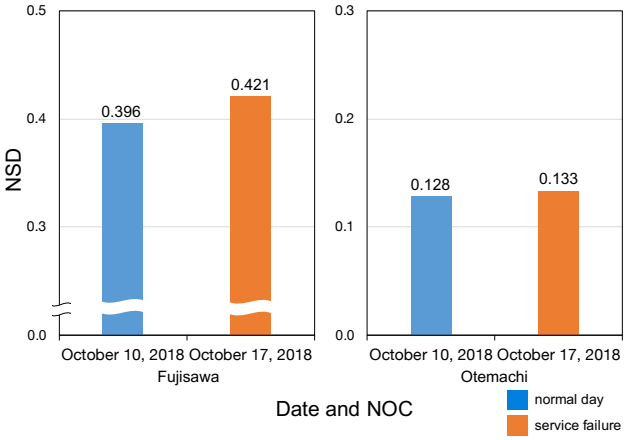

図 5　藤沢 NOC と大手町 NOC の評価結果
Fig. 5　Result of evaluation for 2 NOCs.

ここで，NOC 毎に NSD 値を比較すると，どちらの拠点でも YouTube の接続障害があった 2018 年 10 月 17 日の方が通常日に比べ大きい NSD 値を記録した．この結果両拠点で接続障害が NSD 値に影響を与えていることが分かるが，コア拠点である大手町 NOC の方が通常日との差が小さくなっている．このことから GAMPAL によるインターネットバックボーンの異常事象についてはコア拠点よりもエッジ拠点の方が検知しやすいことが分かる．

6.3　1 時間単位での NSD 評価

既存研究 [1] では異常事象が発生した日について予測値と実測値とを 1 日単位で比較し NSD 値を計算していたが，5.2 節で述べたとおり，実用上ではより短い時間単位での評価が必要である．本稿では YouTube の接続障害が発生した 2018 年 10 月 17 日について 1 時間単位での NSD 値を評価する．YouTube の接続障害は WIDE バックボーンで発生した他の異常事象と異なり異常事象が発生した時刻と回復した時刻が分かっているため，この評価に適している．YouTube の接続障害は日本時間の午前 10 時頃発生し，午前 12 時頃に回復している．

YouTube の接続障害が発生した 10 月 17 日との比較のため，複数の通常日について 1 時間単位での NSD 値の平均値を計算した．1 時間単位の NSD 値は比較する値が 1 日単位より少ないため，外れ値の影響を大きく受ける．そのため通常日を 4 日分選択し平均を計算した．WIDE バックボーンはラベル付きデータでないため，異常日，通常日の選択が難しい．また LSTM-RNN の学習には異常なデータを含まない通常日を選択する必要がある．そこで，普段の平日とは挙動の異なる祝日のデータを学習データから除外している．本節の評価での通常日の予測をするにあたり，日本で 5 月の中旬から 6 月まで祝日がなく，異常事象の報告がないため，6 月 24 ～ 27 日の 4 日分を通常日として選択し評価した．

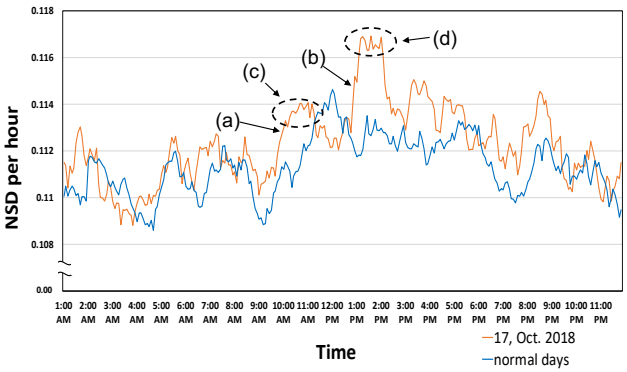

図 6　1 時間単位の NSD 値の結果

Fig. 6　Evaluation of the NSD value per hour.

図 6 に 1 時間単位での NSD 値による評価結果を示す．接続障害が発生した午前 10 時以前は，2 つの NSD 値は似た挙動を示している．しかし午前 10 時を過ぎたところで，10 月 17 日の NSD 値の方が急激に増加している (図 6-(a))．また接続障害が解消した午前 12 時を過ぎた後，10 月 17 日の方が先ほどよりも激しく増加している (図 6-(b))．またこの 2 つの急増は，値が大きくなってから 1 時間程度大きい値のままの状態が続き (図 6-(c),(d))，その後値が小さくなっている．通常日や異常日の他の時間でも値の急増は観測されているが，これらの挙動とは異なりすぐに値が小さくなっている．この挙動が接続障害が 1 時間あたりの NSD 値に与えた影響であると考えられる．10 時頃の急増は接続障害が発生したことによるユーザおよび YouTube へのアクセスの急な減少によるもので，12 時の急増は YouTube の復旧を知ったユーザによるアクセスの急増によるものであると考えられる．このように，1 時間単位での NSD 値を評価することで，GAMPAL は異常検知のリアルタイム性を改良することが可能である

7. おわりに

本稿ではインターネットバックボーンを対象とした汎用異常検知手法 GAMPAL の改良として，既存研究 [1] に加え，AS2500 以外の環境での異常検知性能の評価，インターネットバックボーン中のエッジ拠点とコア拠点での異常検知性能の比較，およびリアルタイム性の改良を提案した．評価の結果，既存研究で評価に用いた WIDE バックボーン (AS2500) に加え， スペインの ISP で観測されたネットワークトラフィックデータである UGR'16 でも複数の種類の異常事象の検知が可能であった．またインターネットバックボーン中ではエッジ拠点の方がコア拠点と比べ，GAMPAL での異常検知が有効に機能することが分かった．さらに GAMPAL による予測値について，評価する時間単位を 1 日から 1 時間へ短縮し，評価指標を 5 分毎に更新すること異常検知のでリアルタイム性の改良が可能であることを確認した．

参考文献

[1] Taku. Wakui, Takao. Kondo, and Teraoka. Teraoka. GAMPAL: Anomaly Detection for Internet Backbone Traffic by Flow Prediction with LSTM-RNN. In *Proc. of 2nd IFIP International Conference on Machine Learning for Networking*, pp. 196–211, 2019.

[2] WIDE backbone. `http://two.wide.ad.jp/`.

[3] H. Liao, C.R. Lin, Y. Lin, and K. Tung. Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications*, Vol. 36, No. 1, pp. 16–24, 2016.

[4] R. Kumar and Sharma D. HyINT: Signature-Anomaly Intrusion Detection System. In *Proc. of ICCCNT 2018*, pp. 1–7, 2018.

[5] J. Kwon, J. Leea, H. Lee, and A. Perrig. PsyBoG: A scalable botnet detection method for large-scale DNS traffic. *Computer Networks*, Vol. 97, pp. 48–73, 2016.

[6] O. Ibidunmoye, A. Rezaie, and E. Elmroth. Adaptive Anomaly Detection in Performance Metric Streams. *IEEE Trans. on Network and Service Management*, Vol. 15, No. 1, pp. 217–231, 2018.

[7] K. Flanagan, E. Fallon, P. Jacob, A. Awad, and P. Connolly. 2D2N: A Dynamic Degenerative Neural Network for Classification of Images of Live Network Data. In *Proc. of IEEE CCNC 2019*, pp. 1–7, 2019.

[8] T.A. Tang, L. Mhamdi, D. McLernon, S. Zaidi, and M. Ghogho. Deep Recurrent Neural Network for Intrusion Detection in SDN-based Networks. In *Proc. of IEEE NetSoft 2018*, pp. 202–206, 2018.

[9] NSL-KDD dataset. `https://www.unb.ca/cic/datasets/nsl.html` (Last accessed 20 Aug 2019).

[10] S. Chen, Y. Chen, and W. Tzeng. Effective Botnet Detection Through Neural Networks on Convolutional Features. In *Proc of IEEE TrustCom/BigDataSE 2018*, pp. 372–378, 2018.

[11] G. Kathareios, A. Anghel, A. Mate, R. Clauberg, and M. Gusat. Catch It If You Can: Real-Time Network Anomaly Detection with Low False Alarm Rates. In *Proc. of IEEE ICMLA 2017)*, pp. 924–929, 2017.

[12] K. Cho, K. Fukuda, H. Esaki, and A. Kato. The Impact and Implications of the Growth in Residential User-to-user Traffic. In *Proc of ACM SIGCOMM 2006*, pp. 207–218, 2006.

[13] TeamYouTube. `https://twitter.com/TeamYouTube/status/1052393799815589889?ref_src=twsrc%5Etfw`.

[14] NETSCOUT. `https://www.netscout.com/blog/asert/call-arms-apple-remote-management-service-udp`.

[15] G. Maciá-Fernández, J. Camacho, R. Magán-Carrión, P. García-Teodoro, and R. Therón. UGR'16: a new dataset for the evaluation of cyclostationarity-based network IDSs. *Computers & Security*, Vol. 73, , 2017.

[16] Chainer: A flexible framework for neural networks. `https://chainer.org/`.

[17] nfdump. `http://nfdump.sourceforge.net` (Last accessed 20 Aug 2019).

[18] bgpdump. `https://bitbucket.org/ripencc/bgpdump/wiki/Home`(Last accessed 20 Aug 2019).

[19] NAVIDIA cuDNN. `https://developer.nvidia.com/cudnn`(Last accessed 20 Aug 2019).

[20] RIPE NCC RIS Raw Data. `http://www.ripe.net/projects/ris/rawdata.html`.