GAMPAL: an anomaly detection mechanism for Internet backbone traffic by flow size prediction with LSTM-RNN

Taku Wakui¹ · Takao Kondo² · Fumio Teraoka³

Received: 31 August 2020 / Accepted: 3 August 2021 / Published online: 28 August 2021
© The Author(s) 2021

Abstract

This paper proposes a general-purpose anomaly detection mechanism for Internet backbone traffic named GAMPAL (General-purpose Anomaly detection Mechanism using Prefix Aggregate without Labeled data). GAMPAL does not require labeled data to achieve general-purpose anomaly detection. For scalability to the number of entries in the BGP RIB (Border Gateway Protocol Routing Information Base), GAMPAL introduces prefix aggregate. The BGP RIB entries are classified into prefix aggregates, each of which is identified with the first three AS (Autonomous System) numbers in the AS_PATH attribute. GAMPAL establishes a prediction model for traffic sizes based on past traffic sizes. It adopts a LSTM-RNN (Long Short-Term Memory Recurrent Neural Network) model that focuses on the periodicity of the Internet traffic patterns at a weekly scale. The validity of GAMPAL is evaluated using real traffic information, BGP RIBs exported from the WIDE backbone network (AS2500), a nationwide backbone network for research and educational organizations in Japan, and the dataset of an ISP (Internet Service Provider) in Spain. As a result, GAMPAL successfully detects anomalies such as increased traffic due to an event, DDoS (Distributed Denial of Service) attacks targeted at a stub organization, a connection failure, an SSH (Secure Shell) scan attack, and anomaly spam.

Keywords Network traffic analysis · Anomaly detection · Internet backbone · LSTM-RNN

1 Introduction

The Internet backbone network contains a large amount of traffic originating from various kinds of users and services [1]. The patterns of such traffic are peaked and jagged, and they change every moment, even during ordinary times. However, the Internet backbone network encounters

anomalies caused not only by network facility failures but also by disturbances such as flash crowds from social phenomena and cyberattacks. Because disturbances are basically observed only in traffic patterns, it is difficult to find each anomaly from the operators' viewpoints. To operate the Internet backbone network stably, it is necessary to establish a general-purpose mechanism for finding these anomalies from traffic information.

Existing anomaly detection mechanisms are categorized into two approaches: signature-based and behavior-based methods. The signature-based approach is suitable for the detection of known anomalies and real-time anomaly detection even for a large amount of traffic, such as Internet backbone traffic [2–4]. However, this technique fails to detect unknown anomalies such as new types of attacks. The behavior-based approach can detect unknown anomalies. Most existing mechanisms use labeled data composed of anomalous and nonanomalous traffic information [5]. However, it is difficult to collect such traffic information. In addition, labeled data might cause overfitting of the target network because the labeled data already define a finite number of anomaly types. Therefore, the behavior-based approach with such labeled data is

✉ Taku Wakui
dona@inl.ics.keio.ac.jp

Takao Kondo
latte@itc.keio.ac.jp

Fumio Teraoka
tera@keio.jp

¹ Graduate School of Science and Technology, Keio University, Yokohama, Japan

² Computer Security Incident Response Team, Keio University, Tokyo, Japan

³ Faculty of Science and Technology, Keio University, Yokohama, Japan

not suitable for general-purpose anomaly detection. It is difficult to detect anomalies by merely observing the behaviors of various kinds of traffic because these data are hidden in other normal traffic. Moreover, most existing anomaly detection mechanisms specialize in a particular environment, such as DCs (Data Centers) for Internet services [6] and SDNs (Software-Defined Networkings) [5], or they focus on a particular anomaly, such as Botnet [7].

This paper proposes a general-purpose anomaly detection mechanism for Internet backbone traffic named *GAMPAL* (*General-purpose Anomaly detection Mechanism using Prefix Aggregate without Labeled data*). GAMPAL aims at generally and efficiently detecting anomalous traffic behavior in the Internet backbone network and providing early warnings to the subscribers of the backbone network. GAMPAL establishes a prediction model for traffic sizes based on past traffic sizes and uses an LSTM-RNN (Long Short-Term Memory Recurrent Neural Network) model focusing on the periodicity of Internet traffic patterns at daily and weekly scales. For scalability to the number of entries in the BGP RIB (Border Gateway Protocol Routing Information Base), GAMPAL introduces *prefix aggregate*. The BGP RIB entries that have the same first three AS (Autonomous System) numbers are classified into a single prefix aggregate. A prefix aggregate is identified with the first three AS numbers. GAMPAL introduces an indicator named the *NSD* (*Normalized Summation of Differences*), which reflects the differences between the predicted flow sizes and the observed flow sizes. An NSD value larger than the usual values implies the presence of anomalies. To evaluate the behaviors of Internet traffic affected by anomalies, GAMPAL has versatility. As a result, GAMPAL has difficulty detecting attacks that cause only small traffic changes compared to anomalous events that cause large traffic changes. However, if large relative changes are observed, for example, when connections between two IP addresses that have not been observed are now observed, GAMPAL can detect such anomalies that do not cause large absolute changes. We implement a traffic information parser produced by NetFlow version 9 and the BGP RIB in the MRT (Multi-Threaded Routing Toolkit) format [8]. We also implement a learning mechanism for our traffic size prediction model based on the LSTM-RNN model. The learning mechanism uses the cuDNN (CUDA Deep Neural Network) [9] library and the Chainer library [10] to support a GPU computing environment. In the evaluation, real traffic information and the BGP RIBs are used. Traffic information is exported from the WIDE backbone network (AS2500) [11] and UGR'16 [12], the dataset published by an ISP (Internet Service Provider) in Spain. The BGP RIBs are exported from the WIDE backbone network and the RIPE NCC (Network Coordination Centre) [13]. The WIDE backbone

Table 1 Acronyms and their definitions

ARMS	Apple Remote Manager Service
AS	Autonomous System
BGP	Border Gateway Protocol
CNN	Convolutional Neural Network
DC	Data Center
DDoS	Distributed Denial of Service
DNN	Deep Neural Network
GAMPAL	General-purpose Anomaly detection Mechanism using Path Aggregate without Labeled data
GRU	Gated Recurrent Unit
ISP	Internet Service Provider
LSTM	Long Short-Term Memory
MCODT	Micro-Cluster Outlier Detection in Time series
MRT	Multi-Threaded Routing Toolkit
MSE	Mean Square Error
NCC	Network Coordination Centre
NDR	Network Detection and Response
NOC	Network Operation Center
NSD	Normalized Summation of Difference
P2P	Peer-to-Peer
RIB	Routing Information Base
RMSPE	Root Mean Square Percentage Error
RNN	Recurrent Neural Network
SDN	Software-Defined Network
SOM	Self-Organization Map
SSH	Secure Shell
UDP	User Datagram Protocol

network is a nationwide backbone network for research and educational organizations in Japan.

This paper uses many acronyms. Table 1 lists the acronyms appearing in this paper and their definitions.

2 Related work

As mentioned in Section 1, existing anomaly detection mechanisms are categorized into two approaches: signature-based and behavior-based methods. The signature-based approach [2] defines some rules to detect anomalies and applies these rules to the logged outputs of servers and network facilities. The behavior-based approach monitors the activities of end hosts or communication sessions in a network system and detects changes in comparison with past activities. Because it is almost impossible to define rules for detecting any kind of anomaly in Internet traffic [3, 4], this paper discusses the existing work on the behavior-based approach.

For an enterprise/DC (Data Center)-scale network, Ibidunmoye et al. [6] propose a performance anomaly

detection mechanism for cloud and Internet services. This mechanism is based on statistical behavior analysis, which includes two techniques: a behavior-based technique with adaptive learning and a prediction-based technique with statistically robust control charts. Flanagan et al. [14] propose a general-purpose anomaly detection mechanism for an enterprise network. This mechanism is derived from a CNN (Convolutional Neural Network)-based classification of the visualizations of traffic information. The traffic information is categorized with the MCODT (Micro-Cluster Outlier Detection in Time series) clustering algorithm and visualized by the SOM (Self-Organization Map) dimensionality reduction algorithm. Tang et al. [5] utilize an intrusion detection mechanism for SDNs. This mechanism uses GRU (Gated Recurrent Unit) RNN-based classification, which is trained on the dataset called NSL-KDD [15].

For Internet-scale networks, Chen et al. [7] propose a botnet traffic detection mechanism based on traffic information in P2P (Peer-to-Peer) networks. This mechanism includes CNN-based classification and a decision tree method for enhancing the anomaly detection rate. Kathareios et al. [16] propose a framework for the real-time anomaly detection of cyberattacks, focusing on Internet traffic. This framework consists of two stages: an unsupervised anomaly detection stage and a supervised anomaly classification stage. The former mechanism is based on an autoencoder neural network, while the latter mechanism is based on a nearest-neighbor classifier model that requires manual operation.

Table 2 shows the comparison between GAMPAL and the existing mechanisms [5–7, 14, 16]. This paper defines the following four metrics: (i) scalability to the Internet, (ii) versatility for any kind of anomaly, (iii) consideration of the periodicity of the traffic patterns, especially for Internet-scale networks, and (iv) necessity of labeled learning data. In terms of scalability, the method in [5] is limited to a small-scale networks. The SOM used in [14] does not have an aggregation mechanism for flow information because it focuses only on enterprise networks, not Internet-scale networks, and does not consider scaling.

In terms of versatility, the approaches in [5–7] are not sufficiently versatile for handling various anomaly types. Tang et al. [5] propose an intrusion detection for SDNs. Ibidunmoye et al. [6] focus on anomalies in cloud and Internet services. Chen et al. [7] have specialized for botnet detection. Flanagan et al. [14] propose a general-purpose anomaly detection mechanism for an enterprise network. Kathareios et al. [16] develop a general-purpose anomaly detection mechanism.

In terms of periodicity, Tang et al. [5] and Flanagan [14] focus on the periodicity of traffic. Tang et al. [5] use a GRU RNN, which can learn data for a longer period than that of a simple RNN. Flanagan et al. [14] uses MCODT, a clustering algorithm for time series data. Chen et al. [7] and Kathareios et al. [16] do not focus on the periodicity of traffic.

In terms of the necessity of labeled data, most existing mechanisms use labeled data. Ibidunmoye et al. [6] use real-world datasets from web services and conducts an evaluation by comparing the validity of the proposed anomaly detection mechanism with that of an open-source package. Flanagan et al. [14] do not use labeled data. The detection validity is evaluated by comparing the moment when the proposed method detects behavior changes and the moment when an event occurs in the real world. Kathareios et al. [16] use labeled data in the supervised anomaly classification stage and unlabeled data in the unsupervised anomaly detection stage.

In terms of the false positive rate, that of the GRU RNN in [5] is approximately 10%. This result is equal to or smaller than those of the other methods compared in [5]. The method in [6] achieves a low false positive rate for datasets from several network services. In [7], the false positive rate of classification with an ANN (Artificial Neural Network) is slightly high, but the utilized confidence testing mechanism with a decision tree helps to reduce the false positive rate. In [16], the final decision of supervised anomaly detection after unsupervised anomaly detection achieves a low false positive rate. Flanagan et al. [14] do not utilize a classification mechanism and does not evaluate the false positive rate.

Table 2 Comparison of GAMPAL and related works

Related work	Enterprise/DC scale			Internet scale		
	[5]	[6]	[14]	[7]	[16]	GAMPAL
Scalability	No	–	No	–	Yes	Yes
Versatility for handling various types of anomalies	No	No	Yes	No	Yes	Yes
Consideration of the periodicity of traffic	Yes	–	Yes	No	No	Yes
Necessity of labeled data	Yes	No	No	Yes	Middle	No
False positive rate	10%	1.5%	–	1.2%	1.3%	Relatively high

In contrast with existing mechanisms, GAMPAL satisfies the first four metrics. Sections 3.2, 3.3, and 3.4 describe how GAMPAL satisfies scalability while considering periodicity, versatility, and the necessity of labeled data. Section 5.6 evaluates the confusion matrices of GAMPAL.

3 Methodology

3.1 Overview of the methodology of GAMPAL methodology

GAMPAL detects anomalies by comparing the observed flow size and the predicted flow size. A flow is identified with a five-tuple, i.e., source and destination addresses, source and destination ports, and a protocol. The flow size is expressed with a time series of the data sizes of the packets included in the flow. The more the packets are observed in a flow, the higher the volume of the flow size information will increase. Since a large number of flows is observed at an observation point, the volume of the flow size information obtained at an observation point is very large. To analyze flow size information efficiently, GAMPAL introduces a *flow size matrix* by aggregating the flow size information *spatially* and *temporally* as shown in Fig. 1-(a).

Focusing on the source and destination addresses in the five-tuple, the order of the number of flows is

$O((\text{number_of_addresses})^2)$, i.e., $O(10^{18})$ in case of IPv4. To spatially aggregate the flow size information, GAMPAL focuses only on the destination address in the five-tuple. Thus, the order of the number of flows is reduced to $O(\text{number_of_addresses})$, i.e., $O(10^9)$ in the case of IPv4. Next, the destination addresses are grouped into the destination address prefixes in the BGP RIB. As of July 2020, the number of IPv4 BGP full routes, i.e., the number of destination address prefixes, is greater than 800,000, i.e., $O(10^5)$, which is still large. In GAMPAL, the destination address prefixes are grouped into the *prefix aggregates* (Fig. 1-(b)), in each of which the first k AS numbers in the AS_PATH attribute are the same. If “3” is adopted as k , the number of prefix aggregates is approximately 30,000, i.e., $O(10^4)$. In Fig. 1, all the observed flows are grouped into n prefix aggregates.

To temporally aggregate flow size information, GAMPAL introduces a *flow size aggregation interval* (Fig. 1-(c), e.g., 5 min). During a flow size aggregation interval (e.g., from 00:00 to 00:05), the data sizes of the observed packets in a prefix aggregate are summed up in its *flow size aggregation slot* (Fig. 1-(d)). GAMPAL collects the aggregated flow sizes during the *flow size learning interval* (Fig. 1-(e), e.g., 1 day). As a result, each prefix aggregate has a time series of the aggregated flow sizes, which is named the *flow size vector* (Fig. 1-(f)). If the flow size learning interval is 1

Fig. 1 Overview of the GAMPAL methodology

day and the flow size aggregation interval is 5 min, the number of the flow size aggregation slots in a flow size vector is 288. The observed flow size matrix (Fig. 1-(g)) is composed of the observed flow size vectors of all the prefix aggregates.

The observed flow size matrix is applied to the LSTM-RNN to generate a model. We denote the model as the predicted flow size matrix (Fig. 1-(h)). In Fig. 1, the predicted flow size matrix for the flow size learning interval from t_y to t_{y+j} is generated from the observed flow size matrix for the flow size learning interval from t_x to t_{x+j} . In parallel, GAMPAL obtains the observed flow size matrix for the flow size learning interval from t_y to t_{y+j} (Fig. 1-(l)).

GAMPAL compares the observed flow size and the predicted flow size of each prefix aggregate by sliding the flow size comparison window (Fig. 1-(m), e.g., 1 h). GAMPAL detects anomalies if the differences between the two aggregated flow sizes exceed a preset threshold.

3.2 Prefix aggregates

As mentioned in the previous subsection, GAMPAL introduces prefix aggregates, in which the first k AS numbers in the AS_PATH attribute are the same. The question is what number is appropriate for the parameter k .

Figure 2 shows the distribution of the AS_PATH lengths of the IPv4 BGP full routes observed in AS2500 on July 17, 2020. The minimum value, the maximum value, the mode value, and the median value are 0 (iBGP routes), 44, 3, and 4, respectively. Since the distribution of the AS_PATH length is heavily biased to small values and has a long and thin tail, it is appropriate to define the prefix aggregates with a short AS_PATH length. GAMPAL adopts the mode value, i.e., 3, as the parameter k for defining the prefix aggregates. The combination of the first three AS numbers is used as the identifier of the prefix aggregate. As a result, 808,775

Fig. 2 Histogram of AS_PATH length

IPv4 BGP full routes (as of August 2020) are grouped into 30,871 prefix aggregates.

On the other hand, at an observation point, a large number of destination addresses that are close to the IP address of the observation point are observed, while a small number of destination addresses that are distant from the IP address of the observation point are observed. This feature is called the locality of Internet traffic. By introducing the prefix aggregates, the destination addresses close to the observation point can be grouped into fine-grained prefix aggregates, while the destination addresses distant from the observation point can be grouped into coarse-grained prefix aggregates.

3.3 Training approach: the day of the week

An Internet backbone network, such as a nationwide backbone network, usually consists of several branch NOCs (Network Operation Centers). As the Internet traffic pattern per NOC typically exhibits periodicity at a daily or weekly scale, there are two approaches for training a prediction model: *weekly training model* and *day of the week training model*. The former approach uses continuous data for a week, e.g., from Sunday to Saturday, as the training data and predicts the traffic for the next week. The latter uses prior data for the same day of the week, e.g., every Monday for the past few weeks, as training data. In a preliminary measurement, we made prediction models based on both approaches and compared them. As a result, the latter approach achieved more valid predictions than the former approach. Furthermore, the traffic pattern of the commodity Internet in Japan exhibits weekly periodicity [17]. Therefore, GAMPAL adopts the day-of-the-week training approach.

3.4 Example of the GAMPAL procedure

Figure 3 shows an example of the prefix aggregation procedure. As described in Section 3.2, the prefixes that have the same first three AS numbers are grouped into a single prefix aggregate. In Fig. 3, the four prefixes are grouped into three prefix aggregates, each of which is identified with the prefix aggregate index. Figure 4 shows an example of the observed traffic information, in which seven packets are recorded. For example, the first and fourth packets are classified into prefix aggregate 1, and their data sizes are summed in the flow size aggregation slot of 00:00. The second packet is classified into prefix aggregate 2, and its data size is summed in the flow size aggregation slot of 00:00. After generating the observed flow size matrix, it is applied to the LSTM-RNN, and then the predicted flow size matrix for the same day of the next week is generated.

Fig. 3 Example of prefix aggregation

4 Implementation

This section describes the implementation of GAMPAL. Figure 5 shows the overall procedures of GAMPAL.

4.1 Implementation environment

GAMPAL is implemented in Python 3.7.0 on a server running Ubuntu Server 18.04.1. Chainer 5.1.0 is used to implement LSTM for training and prediction. nfdump version 1.6.17 [18] is used to convert the packet information. bgpdump version 1.4.99.13 [19] is used to convert the BGP RIB. A GPU is used for the calculations of the LSTM-RNN. The GPU platform is CUDA 9.0.

4.2 Data preprocessing

First, the binary packet information and binary BGP RIB exported from the Internet backbone network are converted to human-readable packet information and a human-readable BGP RIB. This subsection describes the preprocessing of these data (Fig. 5-(1), (2a), (2b), (3a), (3b)).

4.2.1 Processing of NetFlow

NetFlow, which is used as the packet information format in this paper, is recorded in a binary file format. The binary packet information contains the time stamp, five-tuple, and data size of the flow. It is converted to a text file, i.e., the human-readable packet information, using nfdump (Fig. 5-(2a)). Because the binary file is recorded per hour, the text file also contains per-hour packet information.

Fig. 4 Example of flow data aggregation by AS_PATH

4.2.2 Processing of BGP RIB

The BGP RIB is recorded in the MRT format. This binary BGP RIB is converted to a human-readable BGP RIB using bgpdump (Fig. 5-(2b)). Next, the AS_PATHs are extracted from the human-readable BGP RIB and saved in the per-day AS_PATH file (Fig. 5-(3a)). Prefixes are extracted from the human-readable BGP RIB and saved in the per-day prefix file (Fig. 5-(3b)). Figure 6 shows a part of the human-readable BGP RIB, a part of the per-day AS_PATH file, and a part of the per-day prefix file. The procedure numbers in Fig. 6 correspond to those in Fig. 5. From each BGP RIB entry, the AS_PATH is extracted and saved in the per-day AS_PATH file, while the prefix is extracted and saved in the per-day prefix file. Thus, an entry in the per-day AS_PATH file corresponds to the entry in the per-day prefix file at the same line number. For example, as shown in Fig. 6, the first line of the per-day AS_PATH file (4713 2914 13335 13336) corresponds to the first line of the per-day prefix file (1.0.0.0/24).

4.3 Generating the prefix aggregate identifier list and flow size matrix

The blue area in Fig. 5 shows the procedure executed after the preprocessing of the packet information. This subsection describes the generation of a prefix aggregate identifier list and the flow size matrix (Fig. 5-(4)–(7)).

4.3.1 Generating the prefix aggregate identifier list

The per-day AS_PATH file created from the human readable BGP RIB for the latest date in the training data is used to

Fig. 5 Overall procedures of GAMPAL

define the prefix aggregate identifiers and create the prefix aggregate identifier list. The prefix aggregate identifier list includes all of the aggregated AS_PATHs in the BGP RIB without duplication (Fig. 5-(4a)). As described in Section 3.2, the combination of the first three AS numbers is defined as the prefix aggregate identifier. Figure 7 shows a part of the prefix aggregate identifier list created from the

AS_PATH file for May 19, 2018. For example, line 1 shows a prefix aggregate identifier defined with AS4713, AS2914, and AS13335.

4.3.2 Generating the observed flow size matrix

Figure 8 shows the structure of the observed flow size matrix. It has a two-dimensional structure. Each row of the matrix corresponds to a flow size aggregation interval (e.g., 5 min). Each column of the matrix corresponds to a flow size vector. Each element of the matrix (flow size aggregation slot) contains the sum of the data sizes of the corresponding flow for that flow size aggregation interval.

Figure 8 shows that the number of the prefix aggregates in the observed flow size matrix is N . GAMPAL adopts 5 min as the flow size aggregation interval. Note that the temporal aggregation at a 5-min interval is relatively fine-grained in the related work about flow prediction [20–22].

Fig. 6 Examples of the BGP RIB, prefix file, and AS_PATH file **Fig. 7** Example of the prefix aggregate identifier list

Fig. 8 Structure of the flow size matrix

In the case where the flow size learning interval is one day, the number of rows is 288, as shown in Fig. 8.

Figure 9 shows a detailed diagram of the process of generating the *prefix aggregate index*, which is the index in the prefix aggregate identifier list. The procedure numbers in Fig. 9 correspond to those in Fig. 5. The *RB-tree BGP RIB* file is converted from the corresponding prefix file and the *AS_PATH* file (Fig. 9-(4a), (4b)). The RB-tree BGP RIB file adopts a self-balancing binary search tree (Red-Black Tree [23]), in which the prefixes are the main values.

Since the number of prefixes in the BGP RIB will be on the order of the number of BGP full routes, it is necessary to reduce the search time for the destination IP addresses in the human-readable packet information. The observed flow size matrix is generated from the human-readable packet information and the RB-tree BGP RIB file for the same date. The destination IP address of each flow in the human-readable packet file is queried with the prefix in the RB-tree BGP RIB (Fig. 9-(5)). When the prefix is found, the *AS_PATH* corresponding to the prefix is outputted (Fig. 9-(6)). The prefix aggregate identifier list is searched for the

outputted *AS_PATH* to find its index (Fig. 9-(7a)). Finally, as shown in Fig. 10, the observed flow size matrix is generated from the prefix aggregate identifier list and the human-readable packet information. The prefix aggregate index in the prefix aggregate identifier list and the time stamp in the human-readable packet information are used to select the correct element in the observed flow size matrix (Fig. 5-(7a), (7b)). The sum of the data sizes of the flow is added to the corresponding element of the observed flow size matrix.

4.4 Training of the traffic prediction model

The LSTM-RNN model for traffic prediction is implemented with Chainer [10], an open-source deep learning framework, and the `NstepLSTM` class, a class for supporting LSTM-based learning in Chainer. The implementation is optimized to use the `cuDNN` [9] library for a GPU computing environment.

In the LSTM-RNN model, the flow size learning interval must be longer than the expected periodicity. As described in Section 3.3, since the traffic pattern of the commodity Internet in Japan exhibits weekly periodicity, it is sufficient to focus on daily periodicity with GAMPAL. Therefore, GAMPAL adopts one day as the flow size learning interval. As a result, the number of the flow size aggregation slots in the flow size learning interval is 288, as described in Sec. 4.3.2.

Figure 11 shows the procedure for inputting the flow size vector from the observed flow size matrix into the LSTM-RNN. For example, the values of the 1st to 288th

Fig. 9 Overview of prefix aggregate index generation

Fig. 10 Flow size matrix generation

flow size aggregation slots are inputted into the LSTM-RNN to generate a predicted value for the 289th flow size aggregation slot. This value is compared to the value of the 289th flow size aggregation slot. The hidden parameters of the LSTM-RNN are adjusted according to the results of this comparison. The flow size learning interval slides forward with a step size of one.

5 Evaluation

5.1 Datasets

In the evaluation, the flow data (NetFlow) and the BGP RIBs exported from two types of networks are used to verify the versatility of GAMPAL. One of the two networks is the WIDE backbone network (AS2500) [11]. The WIDE backbone network is a nationwide layer-2 and layer-3 network and includes core and branch NOCs (Network

Operation Centers), some of which provide connectivity to stub organizations such as universities. The WIDE backbone network is not only used as an external connection network for each organization but is also frequently used as a testbed for experimentation with new technologies. NetFlow is observed at the branch NOCs (the Fujisawa NOC, etc.) accommodated in universities and the core NOC (the Otemachi NOC). The BGP RIB is observed at a route server in the WIDE backbone network.

The other network is that of a tier-3 ISP in Spain. The ISP makes a labeled dataset called UGR'16 [12] public. UGR'16 has been designed to enable the training of anomaly detection algorithms that consider long-term evolution and traffic periodicity. UGR'16 is a collection of NetFlow traces from over more than 3 months in the ISP, which includes the traffic data of real networking attacks.

5.2 Evaluation indicator

GAMPAL predicts traffic volume, i.e., the data size per unit time (the flow size aggregation interval), for each prefix aggregate. As a result of the analysis, the flow data of WIDE are aggregated to approximately 30,000 prefix aggregates, while those of UGR'16 are aggregated to 4,000 prefix aggregates. The data size per unit time varies by prefix aggregate. The data size of some prefix aggregates consists of zero to several bytes, while those of some prefix aggregates have hundreds of thousands or millions of bytes. It is necessary to define an indicator that can evaluate such a wide range of traffic volumes on the same scale. Therefore, indicators with different scales that are dependent on the underlying data such as the *MSE* (Mean Square Error) are not suitable. In addition, the observed and predicted values may include zero, which denotes that no packets in the prefix aggregate are observed for 5 min. Therefore, indicators that cannot be calculated with data containing

Fig. 11 Input data for the LSTM-RNN and training

zero, such as the *RMSPE* (Root Mean Square Percentage Error) are not suitable.

Thus, this paper defines an indicator named the *NSD* (Normalized Summation of Differences). Let m_i be the i -th observed value, p_i be the i -th predicted value, and T be the number of input values.

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

The NSD is the ratio of the sum of the differences between the observed and predicted values to the sum of the larger value of the observed and predicted values. The NSD takes a value between 0 and 1 regardless of the scale of the values. Additionally, the NSD can be calculated even if any observed or predicted value is zero. The NSD shows how different a predicted value is different from the corresponding observed value; that is, it indicates the validity of the prediction. If the predicted value is the same as the observed value, the NSD value is 0, while if either the predicted value or the observed value is 0 and the other is nonzero, the NSD value is 1. The time unit for evaluation with the NSD can be adjusted flexibly. When calculating the NSD per day, the values for one day (288 values in GAMPAL) are used for calculation. Note that the NSD value is affected by the number of values to be evaluated. Therefore, to evaluate the prediction accuracy via the comparison of NSD values, it is necessary to compare NSD values that are calculated with the same number of data values (e.g., per day and per hour). The threshold for the NSD value difference to detect anomalies depends on the observation points. In other words, it is assumed that the network operator, who wants to employ GAMPAL, first finds the threshold particular to their network before real operation.

5.3 Verification of versatility for detecting abnormal events

5.3.1 Dataset of the WIDE backbone network

We determined whether GAMPAL could detect anomalous traffic caused by connection failures, event, and DDoS attacks by calculating and comparing the NSD values obtained from the Fujisawa NOC in the WIDE backbone network.

We selected October 17, 2018, November 22, 2018, and July 6–8, 2019 as abnormal days because a connection failure occurred on October 17, 2018, an event was held on November 22, 2018, and a DDoS attack was observed on July 6–8, 2019. It was reported that a connection failure to YouTube occurred on October 17, 2018 [24] (a connection failure). On November 22, 2018, a campus festival was held at the university accommodating the NOC

where traffic information was observed (an event). It was reported that a UDP reflection/amplification attack using the ARMS (Apple Remote Manager Service) was observed at the end of June 2019 [25]. This attack was observed at the university, and the university blocked ARMS traffic on July 9, 2019. Therefore, it is assumed that the attack was observed at the university just before July 9, 2019, i.e., July 6–8, 2019 (a DDoS attack).

As the normal days to the abnormal days defined above, we selected the same days of one or two weeks before the abnormal dates. That is, October 10, 2018, to the date of the connection failure (October 17); November 8, 2018, to the date of the event (November 22); and June 22–24, 2019, to the dates of the DDoS attack (July 6–8) were selected. It does not seem that any incident affecting the Internet occurred on the selected normal days.

Figure 12 shows the NSD values for the normal day and the abnormal day when the YouTube connection failure occurred. Figure 13 shows the NSD values for the normal day and the abnormal day when the event was held. Figure 14 shows the NSD values for the normal days and the abnormal days when the DDoS attack was observed. All the NSD values for the normal days were smaller than 0.400, while those for the abnormal days were larger than 0.420. In the connection failure and the event traffic cases, the differences between the NSD values for the abnormal days and the normal days were approximately 0.02. The differences between the NSD values for the abnormal days with the DDoS attack and the normal days just before the DDoS attack were approximately 0.04. The largest NSD value (0.443) was observed on July 8, 2019, which was one day before the university blocked the DDoS attack. From these results, we concluded that a value of 0.02 or larger was appropriate for the threshold particular to the WIDE backbone network.

Fig. 12 NSD values for normal day and abnormal day with connection failure observed in the WIDE backbone network

Fig. 13 NSD values for normal day and abnormal day with event observed in the WIDE backbone network

Figure 15 shows the ratio of the number of flow aggregates that had NSD values of 1.0 to the number of total flow aggregates. A flow aggregate that has an NSD value of 1.0 means that the flow aggregate has some traffic in the observed data, but it has no traffic in the predicted data (i.e., the DDoS data), or vice versa. The values for the abnormal days were approximately 0.3–0.4 points larger than those for the normal days.

Thus, these results indicate that GAMPAL can detect anomalous traffic caused by connection failures, events, and DDoS attacks observed in the WIDE backbone network.

Fig. 14 NSD values for the normal days and abnormal days with DDoS attacks observed in the WIDE backbone network

Fig. 15 Ratio of the number of flow aggregates that have the NSD values of 1.0 to the number of total flow aggregates

5.3.2 UGR'16

UGR'16 is a labeled dataset. UGR'16 has labels for three types of anomalies: SSH (Secure Shell) scan attacks, UDP (User Datagram Protocol) scan attacks, and spam attacks. However, some data are lost for the UDP scan attacks. Then, we tested the detection validity of GAMPAL with respect to the other two types of anomalies. An SSH scan attack was reported in the middle of April 2016, and a spam attack was reported on June 20, 2016. Regarding the ISP for which the UGR'16 data were collected, the flows related to spam attacks were observed every day. While tens to hundreds of spam e-mails are usually observed per day by the ISP, approximately 6 million spam e-mails were observed on June 20. Then, we selected April 12, 13, and 14, 2016, and June 20, 2016, as abnormal days. We selected April 21, 22, and 23, 2016, as normal days because the SSH scan attack ended on April 21, 2016. Due to some data loss in UGR'16, it was impossible to predict the traffic volumes for the normal days just before the SSH scan attack. Additionally, we selected June 13, 17, 18, 24, and 25, 2016, as normal days because these days were just before or after the spam attack. The BGP RIBs collected at Barcelona, Spain were exported from the RIPE NCC.

Figure 16 shows the NSD values for the abnormal days when the SSH scan attack was reported, and Fig. 17 shows the NSD values for the abnormal days when the spam attack was reported. The NSD values of the normal days are the average of those of the normal days just before or after each abnormal day. The error bars in each figure show the standard deviation of each normal day set. The difference between the NSD values for the normal days and the abnormal days when the SSH scan attack was

Fig. 16 NSD values for the normal days and abnormal days with SSH scan attacks observed in UGR'16

reported was larger than 0.030. The difference between the NSD values for normal days and the abnormal day when the spam attack was reported was larger than 0.020. These differences were larger than the standard deviations of normal days. These results also indicate that GAMPAL can detect anomalies caused by the spam attacks and the SSH scan attacks observed in UGR'16.

5.4 Validity of anomaly detection in the WIDE backbone network

Since GAMPAL is an anomaly detection mechanism for backbone networks, it is necessary to validate that GAMPAL can detect anomalies not only at one NOC but also at other NOCs in a backbone network. This subsection describes the evaluation of GAMPAL on other NOCs with different characteristics in the WIDE backbone network.

Fig. 17 NSD values for the normal days and the abnormal day with spam attack observed in UGR'16

The data observed at the Otemachi NOC were also used for evaluation purpose via the same method described in Section 5.3.1. The Fujisawa NOC is a leaf NOC located at the edge of the WIDE backbone network topology, and the Otemachi NOC is one of the core NOCs near the center of the topology.

Figure 18 shows the NSD values observed at the two NOCs (Fujisawa and Otemachi) on the day of the YouTube connection failure.

The scale of the NSD values is different for each NOC. This is because the prediction accuracy is affected by the traffic characteristics such as the amount of traffic and the number of prefix aggregates. Since the configuration of the LSTM-RNN is the same for all prefix aggregates, it is generally more difficult to predict the traffic volume of a prefix aggregate with a large volume and large changes than that of a prefix aggregate with small traffic volume changes. The flow data at the Fujisawa NOC contain approximately 130,000 records per hour in 2018 and 15,000,000 records per hour in 2019. The flow data of the Otemachi NOC contain approximately 100,000 records per hour. These differences in the scales of traffic volume and the types of traffic at each NOC yielded differences in the difficulty of prediction.

Comparing the two NOCs, the NSD values for the day of the connection failure are larger than those for the normal day. In addition, the difference for the Otemachi NOC is smaller than that for the Fujisawa NOC. This result indicates that the connection failure affected the NSD values at both NOCs and that the accuracy of anomaly detection at a leaf NOC is better than that at a core NOC.

5.5 Anomaly detection for a short interval

NSD values are calculated per flow size comparison window. For practicality, the flow size comparison window should be as short as possible for timely detection. Therefore, this subsection describes an anomaly detection evaluation with NSD values calculated for a short flow size comparison window (the NSD values for a short-interval, for short). Since the value of a flow size aggregation slot is the sum of the data sizes in bytes over 5 min, the number of values per prefix aggregate is 12 per hour. Therefore, the NSD value per hour is calculated with 12 predicted values and 12 observed values. In other words, the NSD value from 0:00 to 0:59 is calculated with 12 values from the 1st value (the sum of the bytes for 0:00–0:04) to the 12th value (the sum of the bytes for 0:55–0:59). The next NSD value per hour is calculated with 12 values from the 2nd value to the 13th value. The NSD value is updated every 5 min, by repeating this calculation.

In the discussion of anomaly detection with short-interval NSD values, the NSD value is calculated for the day

Fig. 18 NSD values observed at the Fujisawa NOC and Otemtchi NOC

of the YouTube connection failure (October 17, 2018). This incident is suitable for the evaluation of anomaly detection with short-interval NSD values because its start and end times are clear. The connection failure occurred at approximately 10:00 a.m. and was fixed at 12:00 p.m. in local time. First, we describe the relationship between

the length of the flow size comparison window and the NSD values. Figure 19 shows the NSD values for flow size comparison window of 30 min, 1 h, and 2 h. Comparing the three results, it is found that the longer the flow size comparison window is, the larger the NSD value is. It is also found that the smaller the flow size comparison

Fig. 19 NSD values at short intervals: 30 min, 1 h, and 2 h

Table 3 Detection results according to the NSD values for each time unit

Flow size comparison window	15 min	30 min	45 min	1 h	2 h
Level 1 ($a+\sigma$)	33	31	33	29	34
Level 2 ($a+2\sigma$)	9	11	6	10	6
Approximately 10:00 AM	(10:00)	10:00	(10:15)	9:45	10:10
Approximately 12:00 PM	12:25	12:15	12:30	12:50	13:00

window is, the larger the change in the NSD value at each moment is. Focusing on the changes in the NSD values at approximately 10:00 a.m., the NSD value of each flow size comparison window rises significantly (Fig. 19-(a1), (a2), (a3)). This is probably caused by a decrease in the number of users due to the connection failure. Additionally, at approximately 12:00 p.m., the NSD values rise significantly (Fig. 19-(b1), (b2), (b3)). This is probably caused by simultaneous access to YouTube by many users who knew about the recovery of YouTube connectivity. Furthermore, focusing on the local maximum values of the rises in the NSD values at approximately 10:00 a.m. and 12:00 p.m., the longer the flow size comparison window is, the later the local maximum value is reached (Fig. 19-(c1), (c2), (c3), (d1), (d2), (d3)).

Next, since the NSD values rise significantly at approximately 10:00 a.m. and 12:00 p.m., we analyze the changes in the NSD values at a short interval for each time point. For the normal day (October 10, 2018), the mean values and the standard deviations of the changes in the NSD values at each time stamp are calculated. Let a be the average value and σ be the standard deviation. We define level 1 as changes greater than $a + \sigma$. We also define level 2 as changes greater than $a + 2\sigma$ with two or more consecutive level 1 changes. The number of significant increases in the NSD value in each level is counted. Table 3 shows the times of the changes at each level and whether there is a change at approximately 10:00 a.m. and 12:00 p.m. The time values in the bottom two rows of Table 3 are the moments at which significant rises in the NSD value are detected at level 2. The time values in brackets are those at level 1. The change at approximately 12:00 p.m. is detected at level 2 for each NSD value, while the change at approximately 10:00

a.m., when the connection failure starts, is not detected at level 2 with the flow size comparison windows of 15 min and 45 min. This may be because the change in the NSD value becomes significant throughout the day, i.e., the standard deviation becomes larger, when shortening the flow size comparison window. From these results, it is necessary to evaluate the NSD values calculated with an appropriate period of the flow size comparison window for each network.

Next, we evaluate the YouTube connection failure observed in the WIDE backbone network in more detail with NSD values every 30 min and 1 h during periods containing anomalies associated with the connection failure and those in which recovery can be detected. Figure 20 shows the NSD values per 30 min for the abnormal day (October 17, 2018) and the average of the NSD values per 30 min for the normal days (October 3 and 10, 2018) just before the abnormal days. Before 10:00 a.m., the behaviors of the two NSD values per 30 min are similar. At approximately 10:00 a.m., only the NSD value for the abnormal day rises significantly (Fig. 20-(a)). This is probably caused by a decrease in the number of users and accesses due to the YouTube connection failure. Additionally, at approximately 12:00 p.m., only the NSD value for the abnormal day rises significantly (Fig. 20-(b)). This is probably caused by simultaneous access to YouTube by many users who knew about the recovery of YouTube connectivity. Figure 21 shows the NSD value per hour for the abnormal day (October 17, 2018) and the average of the NSD values per hour for the normal days just before the abnormal day (October 3 and 10, 2018). Similarly, only the NSD value for the abnormal day rises significantly at approximately 10:00 a.m. and 12:00 p.m. In addition,

Fig. 20 NSD values per 30 min for the abnormal day with a connection failure day and normal days

Fig. 21 NSD values per 1 h for abnormal day with connection failure day and normal days

after the rise in the NSD values, the high NSD values are maintained for 1–2 h without decreasing (Fig. 20-(c), (d)). In both results, some significant rises in the NSD values are observed at times other than 10:00 and 12:00 (Fig. 21-(a), (b)). Although no anomalies other than the connection failure to YouTube were reported on October 17, 2018, there might have been some unusual events (e.g., network experimentation in the WIDE backbone network) at the times when the NSD value rises.

In the above discussion on the short-interval NSD values, the NSD values per hour for July 8, 2019 when the DDoS attack was observed and for June 24, 2019 (the normal day) are shown in Fig. 22. Since this attack does not have clear start or end times on these days, the differences in the short-interval NSD values for the anomaly day remain 0.01 or more larger than those of the normal day throughout the day.

As described above, GAMPAL can detect anomalies in a timely manner, even if anomalies occur outside the network being observed or the network suffers from DDoS attacks.

5.6 Evaluation with confusion matrices

This subsection evaluates the detection ability of GAMPAL with confusion matrices. Since GAMPAL does not detect anomalies per flow or per packet, i.e., it does not classify traffic into normal or abnormal types, the false positive rate of classification is not calculated in a similar way to

that used in the related work. To provide insight regarding the false positive rate and false negative rate of GAMPAL, confusion matrices are generated based on whether an anomaly is detected in each time unit.

In this evaluation, we used UGR'16 as the dataset, which was also used in Section 5.3.2. Labeled data long-term data, which can be used for day-of-the-week training, are rare, and UGR'16 is the only such dataset among those used in this paper. First, we examined whether an anomaly exists every 5 min based on the labels in UGR'16. Next, we compare this result with the detection result of GAMPAL. As an evaluation indicator, we use the NSD value per hour, which is updated every 5 min. We define that an anomaly is detected when the NSD value per hour exceeds a threshold. The threshold is calculated based on the NSD value per hour for normal days.

In this evaluation, we selected four periods and generate confusion matrices for each of them. Period 1 is from April 12 to 14, 2016, when the flows of the SSH scan attack were frequently observed. Period 2 is from April 21 to 23, 2016, when the flows of the SSH scan attack were very few and most flows were normal. Period 3 consists of June 13, 2016, when the flows of the spam attack were very few and most flows were normal. Period 4 contains June 20, 2016, when the flows of the spam attack were frequently observed. In Period 1, the attack was observed throughout the day, and the detection results also exhibits high NSD

Fig. 22 NSD values per 1 h for abnormal day with DDoS attacks and normal day

Fig. 23 Confusion matrix for the SSH scan attack

		Detection results		
		Normal	Anomaly	
Actual	Normal	0	0	0
	Anomaly	0	825	825
		0	825	825

		Detection results		
		Normal	Anomaly	
Actual	Normal	774	17	791
	Anomaly	3	31	34
		777	48	825

(a) Period 1
(April 12, 13 and 14, 2016). (b) Period 2
(April 21, 22 and 23, 2016).

values throughout the day. Then, the accuracy is 100%, as shown in Fig. 23(a). In Period 2, anomalous flows were observed in 34 out of 825 time units. Thirty-one time units were observed before the end of the SSH scan attack on April 21, 2016. The other 3 time units were observed on April 22, 2016, and anomalous flows were not observed on April 23. Figure 23(b) shows the confusion matrix of the evaluation for Period 2. All anomalies on April 21, 2016, are detected, and 3 anomalies on April 22, 2016, are not detected. The precision is 64.6%, the false positive rate is 2.1%, and the false negative rate is 8.8%. The anomalies on April 22, 2016, were one SSH scan attack for each time unit, and the SSH scan attack is not detected because it seems that the scale of the SSH scan attack was too small.

Figure 24(a) shows the confusion matrix of the evaluation for Period 3. In Period 3, anomaly flows are observed in 7 out of 275 time units. The precision is 33.3%, the false positive rate is 3.0%, and the false negative rate is 42.9%. Figure 24(b) shows the confusion matrix of the evaluation

for Period 4. In Period 4, anomalous flows are observed in 161 out of 275 time units. The precision is 57.5%, the false positive rate is 82.5%, and the false negative rate is 21.1%. Since the false positive rates in Periods 1, 2 and 3 are low, the false positive rate in Period 4 is high. As a result of the detailed analysis for Period 4 (on June 20, 2016), 70 false positives out of 94 are caused by a feature of the NSD value that remains high just after anomaly convergence. The NSD value is calculated by comparing the observed and predicted traffic behavior for the past an hour to reduce the effect of outliers. Therefore, if there is a large anomaly in the past an hour, it may be detected as an anomaly even if the anomaly has been resolved. Seventeen false positives in Period 1 and 8 false positives in Period 3 are caused by the same reason. If the false positives due to this problem are removed, the precision in Period 4 is 84.1%, and the false positive rate is 21.1%. Compared to those of the anomaly detection methods that utilize classification [5–7, 16], the false positive rate in Period 4 is still slightly higher.

Fig. 24 Confusion matrix for the spam attack

		Detection results		
		Normal	Anomaly	
Actual	Normal	260	8	268
	Anomaly	3	4	7
		263	12	275

		Detection results		
		Normal	Anomaly	
Actual	Normal	20	94	114
	Anomaly	34	127	161
		54	221	275

(a) Period 3
(June 13, 2016). (b) Period 4
(June 20, 2016).

Table 4 Computational complexity for generating the observed flow size matrix

	Memory Usage	CPU Utility	Duration
Converting the packet information. / Fig. 5-(2a)	3.6 MB	100%	25 min
Converting the BGP RIB. / Fig. 5-(2b)	6.7 MB	100%	10 s
Generating the AS_PATH file and Prefix file. / Fig. 5-(3a), (3b)	0.1 MB	100%	5 s
Generating the prefix aggregate identifier list. / Fig. 5-(4a)	0.48 GB	100%	3 s
Generating the RB-tree BGP RIB. / Fig. 5-(4b)	0.6 GB	100%	20 min
Extraction from the packet information. / Fig. 5-(5)	37.8 GB	100%	5 min
Matching to the AS_PATH and adding the byte value. / Fig. 5-(6), (7)	44.5 GB	3500%	24 h

In addition, we also find that some anomalies are easily detected by GAMPAL, but some are not. Although anomalous traffic is observed in the time units during which false negatives occur, spam attacks are not detected because it seems that the scale of the spam attacks is too small. As described in Section 5.3.2, a certain number of spam attacks are observed every day, and the flows related to spam attacks are also included in the data of the normal days used for training. Additionally, it is difficult to detect anomalies in e-mail traffic from only the 5 tuples and the behaviors of flows. Therefore, it is considered that small-scale attacks are not detected by GAMPAL. In contrast, SSH scan attacks are easy to detect because SSH scan attack traffic tends to originate from IP addresses that are not usually observed, and attackers repeatedly attempt to achieve access within a short-period of time.

5.7 Computational complexity

In this section, we evaluate the memory/CPU usage when generating the observed flow size matrix described in Section 4. Table 4 shows the computational complexity evaluation result for generating the observed flow size matrix for a day. Note that the computation time and memory usage may change slightly depending on the number of logs of flow information. Each operation in the table corresponds to a number in Fig. 5. The most computationally complex operations involves searching for the AS_PATH with the destination IP address of the flow in the RB-tree BGP RIB and adding the byte value to the corresponding element in the matrix (Fig. 5-(6), (7)). In this operation, the memory usage is particularly large due to multiprocessing with 35 cores. In terms of the computation time, since GAMPAL adopts the day-of-the-week training approach, a one-week interval is used for prediction, i.e., between the day on which the data of the previous week are collected and the day on which the current data are collected. Although it takes 24 h to completed the operation of Fig 5-(6), (7), the total time required to generate all the observed flow size matrices for training and to output the

prediction results with the LSTM-RNN is less than one week.

6 Conclusion

This paper proposes a general-purpose anomaly detection mechanism called GAMPAL for Internet backbone traffic derived from an LSTM-RNN-based prediction model. To make GAMPAL scalable to the number of the Internet full routes, each flow is mapped to a single prefix aggregate identified with the first three AS numbers of the AS_PATH attribute of the BGP RIB. GAMPAL aims at detecting various kinds of anomalies, not at classifying anomalies.

This paper evaluates the validity of GAMPAL using the observed flow information and the BGP RIB exported from the WIDE backbone network (AS2500), a nationwide backbone network for research and educational organizations in Japan, and UGR'16, a dataset published by an ISP in Spain.

The evaluation shows that when a stub organization of the backbone network suffers from a DDoS attack, GAMPAL can detect the difference between the predicted and observed flow sizes. The evaluation also shows that a leaf NOC located at an edge of the backbone network topology is the most effective observation point for anomaly detection.

GAMPAL also detects anomalies caused by a connection failure and event traffic. The evaluation conducted using UGR'16 shows that GAMPAL can detect anomalies caused by a SSH scan attacks and a spam attacks. These evaluations demonstrate the versatility of GAMPAL to address anomalies and datasets.

In addition, anomaly detection evaluation with a short interval indicates that GAMPAL can detect anomalies in a timely manner by evaluating the difference between the predicted and observed flow sizes for time units shorter than one day. Therefore, GAMPAL properly reflects the state of the Internet backbone with only the traffic size. However, the false positive rate of GAMPAL with short-interval NSD values is relatively large. Thus, GAMPAL may fit the

defense in depth strategy [26]. The defense in depth strategy is an information security concept of defending against threats with multiple layers of security controls. GAMPAL detects anomalies as soon as they occur and is expected to prompt further action, such as localizing the failure point or blocking troubled networks. In order to reduce the computational complexity of GAMPAL, we implement an LSTM-RNN with a few layers and fewer units. Future work will reduce the numbers of false positives and false negatives by improving the computational environment to enable it to deal with heavy computational complexity and by introducing more complex prediction models that can learn network traffic for GAMPAL more accurately.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Fraleigh C, Tobagi F, Diot C (2003) Provisioning ip backbone networks to support latency sensitive traffic. In: IEEE INFOCOM 2003. Twenty-second annual joint conference of the IEEE computer and communications societies (IEEE Cat. No.03CH37428), vol 1, pp 375–385
2. Liao H, Lin CR, Lin Y, Tung K (2016) Intrusion detection system: A comprehensive review. *J Netw Comput Appl* 36(1):16–24
3. Kumar R, Sharma D (2018) HyINT Signature-anomaly intrusion detection system. In: *Proc. of ICCNT 2018*, pp 1–7
4. Kwon J, Leea J, Lee H, Perrig A (2016) PsyBoG: A scalable botnet detection method for large-scale DNS traffic. *Comput Netw* 97:48–73
5. Tang TA, Mhamdi L, McLernon D, Zaidi S, Ghogho M (2018) Deep recurrent neural network for intrusion detection in SDN-based networks. In: *Proceedings of IEEE NetSoft 2018*, pp 202–206
6. Ibidunmoye O, Rezaie A, Elmroth E (2018) Adaptive anomaly detection in performance metric streams. *IEEE Trans Netw Serv Manag* 15(1):217–231
7. Chen S, Chen Y, Tzeng W (2018) Effective botnet detection through neural networks on convolutional features. In: *Proceedings of IEEE TrustCom/BigDataSE 2018*, pp 372–378
8. Petrie C, King T (2017) Multi-threaded routing toolkit (MRT) routing information export format with BGP additional path extensions. RFC 8050 IETF
9. NVIDIA cuDNN. <https://developer.nvidia.com/cudnn> (Last accessed 20 Aug 2019)
10. Chainer: A flexible framework for neural networks. <https://chainer.org/>
11. WIDE backbone. <http://two.wide.ad.jp/>
12. Maciá-Fernández G, Camacho J, Magán-Carrión R, García-Teodoro P, Therón R (2017) UGR'16: a new dataset for the evaluation of cyclostationarity-based network IDSs. *Computers & Security*: 73
13. RIPE NCC RIS Raw Data. <http://www.ripe.net/projects/ris/rawdata.html>
14. Flanagan K, Fallon E, Jacob P, Awad A, Connolly P (2019) 2D2N A dynamic degenerative neural network for classification of images of live network data. In: *Proceedings of IEEE CCNC 2019*, pp 1–7
15. NSL-KDD dataset. <https://www.unb.ca/cic/datasets/nsl.html> (Last accessed 20 Aug 2019)
16. Kathareios G, Anghel A, Mate A, Clauberg R, Gusat M (2017) Catch it if you can: Real-time network anomaly detection with low false alarm rates. In: *Proceedings of IEEE ICMLA 2017*, pp 924–929
17. Cho K, Fukuda K, Esaki H, Kato A (2006) The impact and implications of the growth in residential user-to-user traffic. In: *Proceedings of ACM SIGCOMM 2006*, pp 207–218
18. nfdump. <http://nfdump.sourceforge.net>. (Last accessed 20 Aug 2019)
19. bgpdump. <https://bitbucket.org/ripenncc/bgpdump/wiki/Home>. (Last accessed 20 Aug 2019)
20. Lazaris A, Prasanna VK (2019) An lstm framework for modeling network traffic. In: *Proceedings of 4th IFIP/IEEE symposium on integrated network and service management*, pp 19–24
21. Papagiannaki K, Taft N, Zhang Z, Diot C (2003) Long-term forecasting of internet backbone traffic: observations and initial models. In: *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No.03CH37428)*, vol 2, pp 1178–1188
22. Barakat C, Thiran P, Iannaccone G, Diot C, Owezarski P (2003) Modeling internet backbone traffic at the flow level. *IEEE Trans Signal Process* 51(8):2111–2124
23. Red-Black-Tree. <https://developer.nvidia.com/cudnn> (Last accessed 20 Aug 2019)
24. TeamYoutube. https://twitter.com/TeamYouTube/status/1052393799815589889?ref_src=twsrc
25. NETSCOUT. <https://www.netscout.com/blog/asert/call-arms-apple-remote-management-service-udp>
26. Smith CL (2003) Understanding concepts in the defence in depth strategy. In: *IEEE 37th Annual 2003 International Carnahan Conference on Security Technology, 2003. Proceedings*, pp 8–16

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.