GAMPAL: Anomaly Detection for Internet Backbone Traffic by Flow Prediction with LSTM-RNN

Taku Wakui, Takao Kondo, Fumio Teraoka

GAMPAL: Anomaly Detection for Internet Backbone Traffic by Flow Prediction with LSTM-RNN

Taku Wakui[1][0000−0003−0750−2348], Takao Kondo[1,2][0000−0003−3572−9304], and Fumio Teraoka[3][0000−0001−8839−3344]

[1] Graduate School of Science and Technology, Keio University
[2] Headquarters of Information Technology Center, Keio University
[3] Faculty of Science and Technology, Keio University

Abstract. This paper proposes a general-purpose anomaly detection mechanism for Internet backbone traffic named *GAMPAL (General-purpose Anomaly detection Mechanism using Path Aggregate without Labeled data)*. GAMPAL does not require labeled data to achieve a general-purpose anomaly detection. For scalability to the number of entries in the BGP RIB (Routing Information Base), GAMPAL introduces *path aggregates*. The BGP RIB entries are classified into the path aggregates, each of which is identified with the first three AS numbers in the AS_PATH attribute. GAMPAL establishes a prediction model of traffic throughput based on past traffic throughput. It adopts the LSTM-RNN (Long Short-Term Memory Recurrent Neural Network) model focusing on periodicity in weekly scale of the Internet traffic pattern. The validity of GAMPAL is evaluated using the real traffic information and the BGP RIB exported from the WIDE backbone network (AS2500), a nation-wide backbone network for research and educational organizations in Japan. As a result, GAMPAL successfully detects traffic increases due to events and DDoS attacks targeted to a stub organization.

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

1 Introduction

The Internet backbone network contains large amount of traffic originated from various kinds of users and services. The traffic pattern is peaky and jaggy, which changes every moment even in ordinary times. On the other hand, the Internet backbone network might encounter anomalies caused by not only failures of network facilities but also disturbances such as flash crowds from social phenomenon and cyber attacks. Because the disturbances are basically observed only in traffic pattern, it is difficult to find each anomaly from the operators' viewpoints. In order to operate the Internet backbone network stably, it is necessary to establish a general-purpose mechanism for finding these anomalies from traffic information.

Anomaly detection mechanism are categorized into two approaches: signature-based approach and behavior-based approach. The signature-based approach can detect known anomalies. It is suitable for real-time detection[1–3]. However, it fails to detect unknown anomalies such as new attacks. The behavior-based approach can detect unknown anomalies. Most of existing mechanisms use labeled data composed of anomaly and non-anomaly traffic information[4]. However, it is difficult to collect such traffic information. In addition, the labeled data causes overfitting to the target network. Therefore, the behavior-based approach is not suitable for general-purposed anomaly detection. Also, Most of existing anomaly detection mechanisms are specialized for a particular environment such as a DC (Data Center) for Internet Services[5] and SDN (Software-Defined Networking)[4] or they focus on a particular anomaly such as DDoS (Distributed Denial of Service)[6]. This paper proposes a general-purpose anomaly detection mechanism for Internet backbone traffic named *GAMPAL (General-purpose Anomaly detection Mechanism using Path Aggregate without Labeled data)*. GAMPAL establishes a prediction model of traffic throughput based on the past traffic throughput and utilizes the LSTM-RNN (Long Short-Term Memory Recurrent Neural Network) model focusing on periodicity in daily or weekly scale of the Internet traffic pattern. For scalability to the number of entries in the BGP RIB (Routing Information Base), GAMPAL introduces *path aggregates*. The BGP RIB entries are classified into the path aggregates, each of which is identified with the first three AS numbers in the AS_PATH attribute. GAMPAL generates predicted throughput for each path aggregate. In GAMPAL, an indicator named *NSD (Normalized Summation of Differences)* is introduced, which reflects the difference between the predicted throughput and the observed throughput. Anomaly is detected if the NSD value is larger than the threshold.

This paper implements a parser of traffic information produced by NetFlow version 9 and the BGP RIB in the MRT format[7] and a learning mechanism for a prediction model of traffic throughput based on LSTM-RNN model. The learning mechanism utilizes the cuDNN (CUDA Deep Neural Network)[8] library and Chainer library[9] in order to support a GPU computing environment. The evaluation utilizes the real traffic and the BGP RIBs exported from the WIDE backbone network (AS2500)[10], a nation-wide backbone network for research and educational organizations in Japan.

2 Related Work

Anomaly detection mechanisms are categorized into two approaches: signature-based approach and behavior-based approach. The signature-based approach[1] defines some rules to detect anomalies and applies these rules to logging outputs of servers and network facilities. The behavior-based approach monitors activities of end hosts or communication sessions in a networked system and detects some changes compared with the past ones. Because it is almost impossible to define rules to detect any kinds of anomalies in the Internet traffic[2, 3], this paper discusses the existing work based on the latter approach.

For enterprise / DC (Data Center) scale network, [5] proposes 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. [11] proposes a general-purpose anomaly detection mechanism for an enterprise network. This mechanism is based on CNN-based classification of visualization of traffic information. The traffic information is categorized with the MCODT (Micro-Cluster Outlier Detection in Time series) cluster algorithm and visualized by the SOM (Self Organization Map) dimentionality reduction algorithm. [4] is an intrusion detection mechanism for SDN (Software-Defined Networking). This mechanism utilizes GRU (Gated Recurrent Unit) RNN based classification which is learned by the `NSL-KDD`[12] labeled data set.

For Internet scale network, [6] proposes a botnet traffic detection mechanism based on traffic information in P2P networks. This mechanism includes CNN-based classification and a decision tree method for enhancing anomaly detection rate. [13] proposes a framework for real-time anomaly detection of cyber-attacks focusing on the Internet traffic. This framework combines unsupervised and supervised classification mechanisms. The former is based on an auto-encoder neural network while the latter is based on a nearest neighbor classifier model in which the manual operation is required.

Table 1 shows the comparison between GAMPAL and the existing mechanisms[4–6, 11, 13]. There are four metrics as follows: (i) scalability to the Internet, (ii) versatility to any kinds of anomalies, (iii) consideration on periodicity of the traffic pattern especially for Internet-scale network, and (iv) necessity of labeled learning data. In terms of scalability, [4] proposes an anomaly detection for small scale network. The SOM used in [11] does not have an aggregation mechanism because it focuses only on an enterprise network, not an Internet-scale network, and does not consider scaling. In terms of versatility, [4–6] are not versatile to anomaly types. [4] proposes an intrusion detection for SDN. [5] focuses on on anomalies in cloud and Internet services. [6] is a mechanism specialized for botnet detection. [11] proposes a general-propose anomaly detection mechanism for an enterprise network. [13] proposes a general-purpose anomaly detection mechanism. In terms of consideration on periodicity, [4, 11] focus on periodicity of traffic. [4] uses GRU RNN which can learn data for a longer period than simple RNN. [11] uses MCODT, a clustering algorithm for time-series data. [6, 13] do not focus on periodicity of traffic. In terms of necessity of labeled data, most of existing mechanisms use labeled data. [5] uses real-world datasets of Web services and evaluates the validity of anomaly detection by comparing with that of an open source package. [11] does not use labeled data. The detection validity is evaluated by comparing the time when the proposed method detects behavior changes and the time when an event occurs in the real-world. [13] uses labeled data in supervised classification and un-labeled data in unsupervised classification. In contrast to existing mechanisms, GAMPAL satisfies the four metrics.

Table 1. Comparison of related work.

Related work	Enterprise/DC Scale			Internet Scale		
	[4]	[5]	[11]	[6]	[13]	GAMPAL
Scalability	No	-	No	-	Yes	Yes
Versatile to the types of anomaly	No	No	Yes	No	Yes	Yes
Consideration on periodicity of traffic	Yes	-	Yes	No	No	Yes
Necessity of labeled data	Yes	No	No	Yes	Middle	No

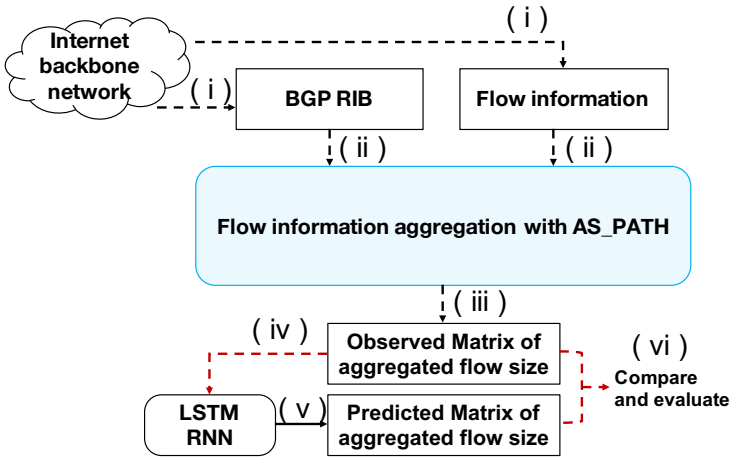

Fig. 1. Overview of GAMPAL methodology.

3 Methodology

3.1 Overview of GAMPAL methodology

Figure1 shows the overview of the GAMPAL methodology. GAMPAL is an anomaly detection mechanism using a prediction model based on the LSTM-RNN model. First, the flow information and the BGP RIB used in flow information aggregation are exported from an Internet backbone network (Fig.1-(i)). *The observed matrix of aggregated flow size* is generated from the flow information and the AS_PATH attribute of the BGP RIB (Fig. 1-(ii), (iii)). Next, the matrix of aggregated flow size is inputted to the LSTM-RNN (Fig.1-(iv)). As a result, the predicted matrix of aggregated flow size is outputted. GAMPAL detects anomalies with a metric which measures the difference between the predicted flow size and the observed flow size (Fig.1-(vi)).

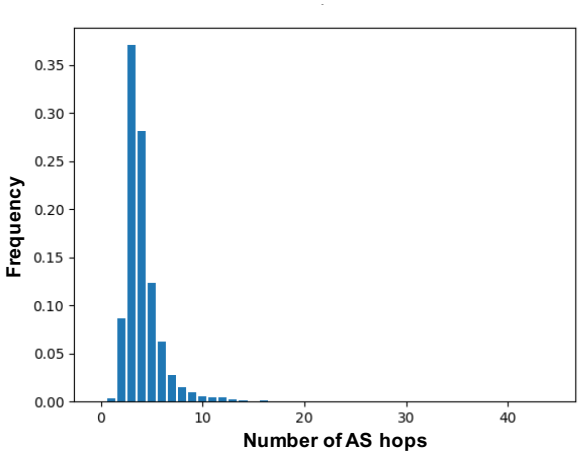

Fig. 2. Histogram of AS_PATH length.

3.2 Flow data aggregation with AS_PATH

GAMPAL adopts throughput of each flow as a general-purpose metric of traffic pattern in the Internet backbone network. A flow can be identified with the five tuples, i.e., source/destination IP addresses, source/destination ports, and protocol number. In a backbone network in which the BGP full routes are maintained, the order of the number of flows will be the square of the number of the BGP full routes. To make GAMPAL scalable to the Internet, the observed flows are mapped into groups named *the path aggregates*.

GAMPAL utilizes the AS_PATH attribute of the BGP RIB to define the path aggregates. At a traffic measurement node in a backbone network, a large number of destination addresses close to the IP address of the measurement node will be observed while a small number of destination addresses distant from the IP address of the measurement node will be observed. Therefore, the observed flows that have destination addresses close to the IP address of the measurement node should be classified in more detail to effectively detect anomalies. In contrast, it is sufficient to roughly classify the observed flows that have destination addresses distant from the IP address of the measurement node to detect anomalies. Figure 2 shows the distribution of the AS_PATH length of the IPv4 BGP full routes observed in AS2500 on June 17, 2018. The minimum value, the maximum value, the mode value, and the median value are 0 (iGP 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 sufficient to define path aggregates with a short AS_PATH length.

GAMPAL adopts the mode value of the AS_PATH length, i.e., 3, to define the path aggregates. That is, the first three AS numbers of the AS_PATH attribute defines a single path aggregate and they are used as *the path aggregate identifier*.

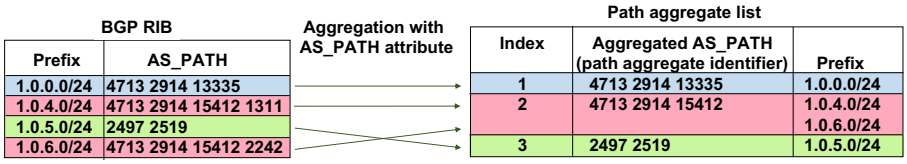

Fig. 3. Example of AS_PATH aggregation.

Consequently, 727,261 IPv4 BGP full routes (as of in January 2019) can be classified into 31,258 path aggregates.

Each observed flow is mapped to a single path aggregate to which the BGP route for the destination address prefix of the observed flow is classified. Thus, a path aggregate is composed of the path aggregate identifier and IP address prefixes that are mapped to the path aggregate. As a result, the number of observed flows can be aggregated to the number of the path aggregates at the most.

3.3 Training Approach: the Day of the Week

An Internet backbone network, such as a nation-wide backbone network usually consists of several branch NOCs (Network Operation Centers). As the Internet traffic pattern per NOC typically has periodicity in a daily or weekly scale, there are two approaches for training the prediction model: *the weekly training model* and *the day of the week training model*. The former uses continuous data of a week, e.g., from Sunday to Saturday, as the training data and predicts the traffic of the next week. The latter uses past data on the same day of the week, e.g., every Monday of the past two months, as training data. In a preliminary measurement, we made prediction models based on both approaches and compared them. As a result, the latter approach showed more valid prediction than the former one. Furthermore, the traffic pattern of the commodity Internet in Japan shows a weekly periodicity [14]. Therefore, GAMPAL adopts the latter approach, i.e., the day of the week training approach.

3.4 Overview of Prediction Procedures

Figure 3 shows an example of AS_PATH aggregation. First, GAMPAL creates the path aggregate list with the flow aggregation method described in Sec.3.2. As shown in Fig. 3, the entries in the BGP RIB are classified into the path aggregates with the first three AS numbers of the AS_PATH attribute. For example, the two entries of the prefix `1.0.4.0/24` and the prefix `1.0.6.0/24` in the BGP RIB are classified to a single path aggregate (the *Path aggregate 2* in the table of the path aggregate list), because the first three AS numbers of the AS_PATH attribute are the same.

After creating the path aggregate list, the observed matrix of aggregated flow size are created with the path aggregate list. As shown in Fig.4, the observed

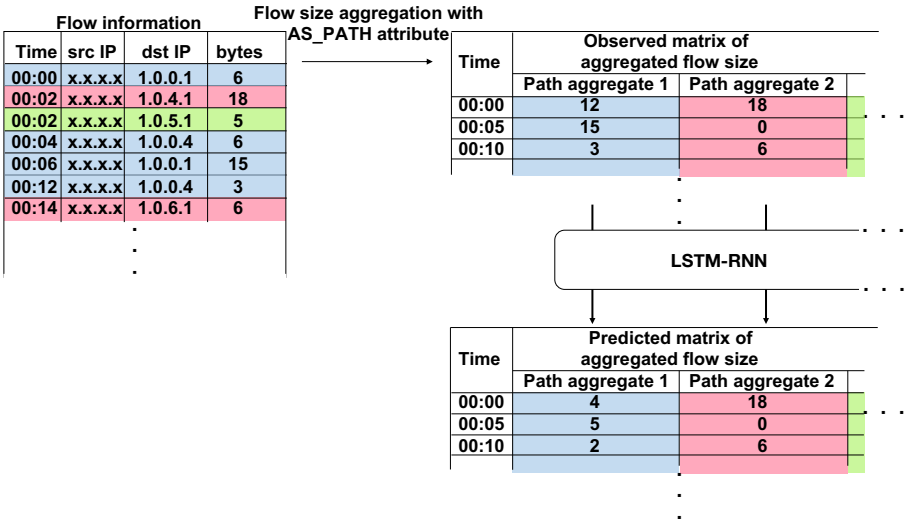

Fig. 4. Example of flow data aggregation by AS_PATH.

matrix of aggregated flow size has time-series entries, each of which contains the sum of the flow size during the time period. The data size of an observed flow is aggregated into an entry of the observed matrix of aggregated flow size. For example, as shown in Fig. 4, the entries whose destination address matches the prefix `1.0.4.0/24` and the prefix `1.0.6.0/24` in the Flow information table are mapped to the `Path aggregate 2` in the observed matrix of aggregated flow size. Each entry of the observed matrix of aggregated flow size contains the sum of the bytes for 5 minutes.

Finally, GAMPAL generates the predicted matrix of aggregate flow size per path aggregate with the LSTM-RNN model.

4 Implementation

Figure 5 shows overall procedures of GAMPAL. This section describes the implementation 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`[15] is used to convert the flow information. `bgpdump version 1.4.99.13`[16] is used to convert the BGP RIBs. GPU (Graphics Processing Unit) is used for calculations of LSTM-RNN. The GPU platform is `CUDA 9.0`.

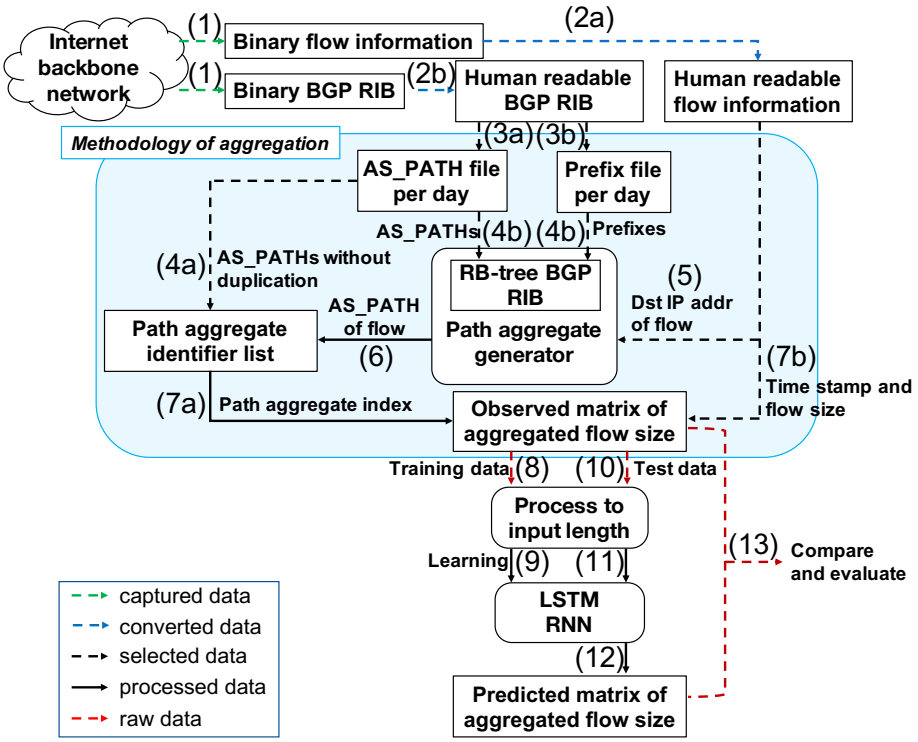

Fig. 5. Overall Procedures of Traffic Prediction

4.2 Data Pre-processing

First, *binary flow information* and *binary BGP RIB* exported from the Internet backbone network are converted to *human readable flow information* and *human readable BGP RIB* (Fig.5-(1),(2a),(2b)).

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

Processing of BGP RIB The BGP RIB is recorded in the MRT format. This binary BGP RIB is converted to the 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 AS_PATH file per day* (Fig 5-(3a)). Prefixes are extracted

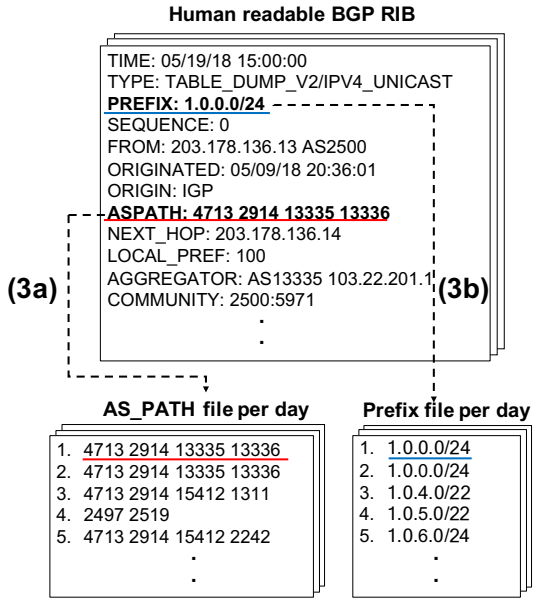

Fig. 6. Examples of BGP RIB, Prefix file, and AS_PATH file.

from the human readable BGP RIB and saved in *the Prefix file per day* (Fig.5-(3b)). Figure 6 shows a part of the human readable BGP RIB, a part of the AS_PATH file per day, and a part of the Prefix file per day. 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 AS_PATH file per day while the prefix is extracted and saved in the Prefix file per day. Thus, an entry in the AS_PATH file per day corresponds to the entry in the Prefix file per day at the same line number. For example, as shown in Fig. 6, the first line of the AS_PATH file per day (`4713 2914 13335 13336`) corresponds to the first line of the Prefix file per day (`1.0.0.0/24`).

4.3 Generating path aggregate identifier list and matrix of aggregate flow size

The blue area in Fig. 5 shows the procedure after the pre-processing of the flow information. This section describes the definition and generation of a *path aggregate identifier list*, generation of a matrix of aggregate flow size (Fig. 5-(4)-(7)).

Generating path aggregate identifier list The AS_PATH file per day created from the human readable BGP RIB of the latest date in the training data is used to define the path aggregate identifier and create the path aggregate

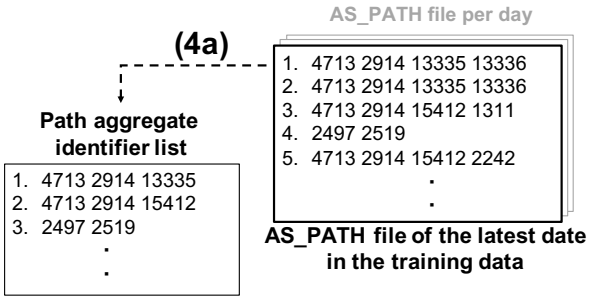

Fig. 7. Example of the path aggregate identifier list.

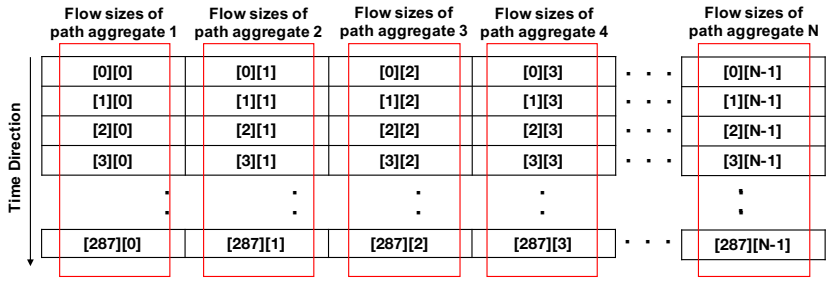

Fig. 8. The structure of observed matrix of aggregated flow size.

identifier list. The path aggregate identifier list includes all of the aggregated AS_PATH in the BGP RIB without duplication (Fig. 5-(4a)). As described in Sec. 3.2, the combination of the first three AS numbers is defined as the path aggregate identifier. Figure 7 shows a part of the path aggregate identifier list created from the AS_PATH file on May 19, 2018. For example, the line 1 of the Path aggregate identifier list in Fig. 7 shows a path aggregate identifier defined with AS4713, AS2914, and AS13335.

Generating observed matrix of aggregated flow size Figure 8 shows the structure of the observed matrix of aggregated flow size. It has a two dimensional structure. Each row of the matrix corresponds to a specific time period (e.g., 5 minutes). Each column of the matrix corresponds to a path aggregate. Each element of the matrix contains the sum of bytes of the corresponding flow for the time period. Figure 8 shows that the number of the path aggregates in the observed matrix of aggregated flow size is N. GAMPAL adopts 5 minutes as the time period of each row. In case that the observed matrix of aggregated flow size are divided per day, the number of rows is 288 as shown in Fig. 8.

Figure 9 shows a detailed diagram for generating *the path aggregate index*, which is the index in the AS_PATH file per day and the Prefix file per day. The procedure numbers in Fig. 9 correspond to those in Fig. 5. The RB-tree RIB

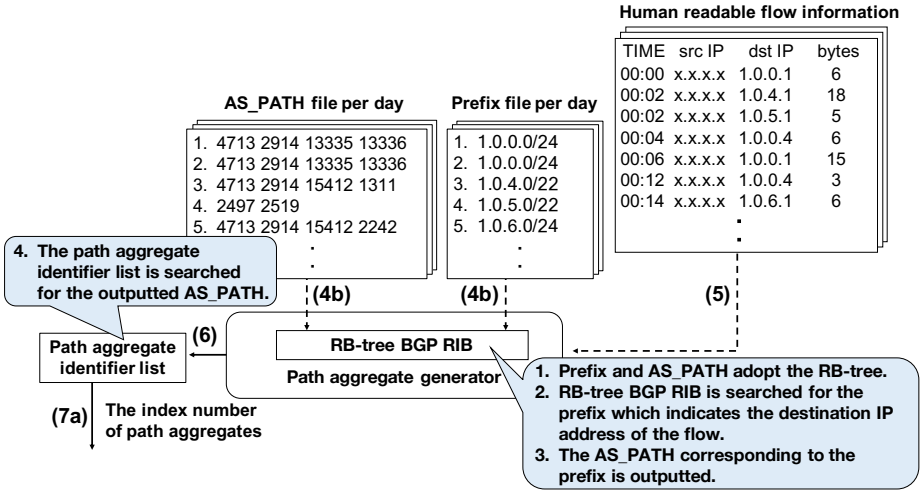

Fig. 9. Overview of path aggregate index generation.

file is converted from the corresponding Prefix file and the AS_PATH file (Fig. 9-(4a), (4b)). The RB-Tree RIB file adopts a self-balancing binary search tree (Red-Black-Tree[17]) in which the prefixes are the main values. Since the number of prefixes in the BGP RIB will be in the order of the number of the BGP full routes, it is necessary to reduce the search time for the destination IP addresses in the human readable flow information. The observed matrix of aggregated flow size is generated from the human readable flow file and the RB-tree RIB file of the same date. The destination IP address of each flow in the human readable flow file is queried with the prefix in the RB-tree RIB (Fig. 9-(5)). When the prefix is found, the AS_PATH corresponding to the prefix is outputted (Fig. 9-(6)) and the path aggregate identifier list (Fig. 9-(7a)). Finally, as shown in Fig. 10, the observed matrix of aggregated flow size is generated from the path aggregate identifier list and the human readable flow information. The path aggregate index in the path aggregate identifier list and the time stamp in the human readable flow information are used to select the element in the observed matrix of aggregated flow size (Fig. 5-(7a),(7b)). The sum of bytes of the flow is added to the corresponding element of the observed matrix of aggregated flow size.

4.4 Training of Traffic Prediction Model

The LSTM-RNN model for traffic prediction is implemented with `Chainer`[9], 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 `cuDNN` (CUDA Deep Neural Network)[8] library for a GPU computing environment.

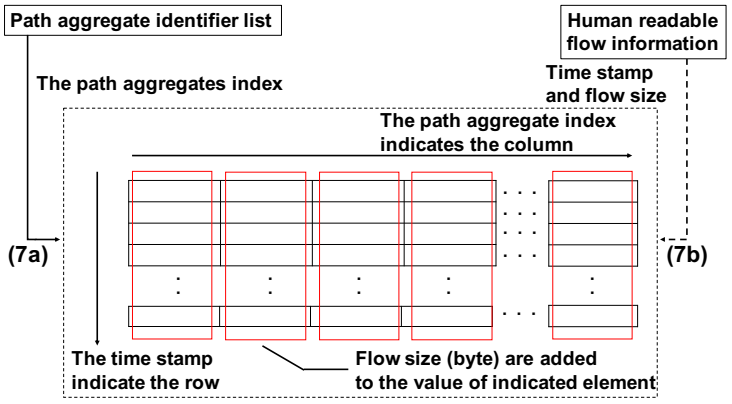

Fig. 10. The matrix of aggregated flow size generation.

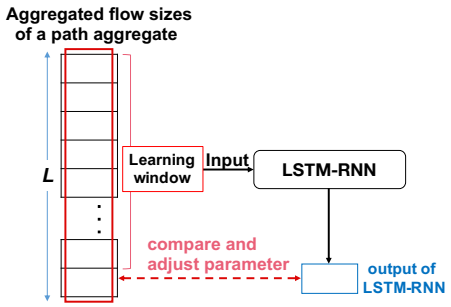

Fig. 11. Input data to LSTM-RNN and training.

In the LSTM-RNN model, the time period of the learning data must be longer than that of expected periodicity. As described in Sec.3.3, since the traffic pattern of the commodity Internet in Japan shows weekly periodicity, it is sufficient to focus on daily periodicity in GAMPAL. Because Sec. 4.3 describes that each element in the observed matrix of aggregated flow size is the sum of the bytes per path aggregate within 5 minutes, the number of rows of the observed matrix of aggregated flow size is 288. Therefore, the time period of expected periodicity is 288 in GAMPAL.

Figure 11 shows the way to input the elements of a path aggregate in the observed matrix of aggregated flow size. Suppose that the value of L is larger than the expected periodicity (i.e., 288 elements in the matrix of aggregated flow size) of the traffic pattern. The learning window specifies $L-1$ out of L elements. The specified elements can be inputted and the remaining element is compared with the output. The parameters for LSTM-RNN are adjusted according to the result of this comparison. The learning window slides forward one by one.

5 Evaluation

5.1 Datasets

In the evaluation, the flow data (NetFlow) and the BGP RIB exported from WIDE backbone Network (AS2500)[10] are used. The backbone network is a nation-wide Layer-2 and Layer-3 network and includes branch NOCs, some of which provide connectivity to stub organizations such as universities. The backbone network is not only used as an external connection network for each organization, but also frequently used as a testbed for experimentation of new technologies. NetFlow is observed at a branch NOC accommodated in a university and the BGP RIB is observed at a route server in the backbone network.

5.2 Evaluation Indicator

GAMPAL predicts throughput, i.e., the number of bytes per unit time, for each of approximately 30,000 path aggregates. The number of bytes per unit time varies for each path aggregate. Some path aggregates have zero to several bytes while some path aggregates record hundred thousands or millions bytes. It is necessary to define an indicator that can evaluate these path aggregates in the same scale. Therefore, indicators with different scales depending on the data such as MSE (*Mean Square Error*) are not suitable. In addition, the measured and predicted values may include zero, which means there was no flow for 5 minutes. Therefore, indicators that cannot be calculated with data containing zero such as $RMSPE$ (*Root Mean Square Percentage Error*) are not suitable. Thus, this paper defines an indicator named NSD (*Normalized Summation of Differences*) where m_i denotes the i th observed value, p_i denotes the i th predicted value, and T denotes the number of input values.

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

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. NSD takes a value between 0 and 1 regardless of the scale of value. Also, NSD is the indicator that can be calculated even if the observed or predicted value is zero. NSD shows how much the predicted value is different from the observed value, that is, it shows the validity of prediction. If the difference between the observed value and the predicted value is small, the NSD value is small.

5.3 Validity of General-Purpose Anomaly Detection

In the evaluation, the NSD value is calculated for normal and abnormal days. On normal days, there seems to be no incident affecting the network. On abnormal days, an incident may have occurred. In the evaluation, June 24-25, 2018, and June 22-24, 2019 are selected as normal days, while October 17, 2018, November

Table 2. Dates of event traffic and normal traffic.

Attribute	Target date of evaluation	Training data
Normal	Jun. 24, 2018	May 6,13 20,27, Jun. 3,10,17, 2018
Normal	Jun. 25, 2018	May 5,14,21,28, Jun. 4,11,18, 2018
Event	Oct. 17, 2018	Sep. 5,12,19,26, Oct. 3,10, 2018
Event	Nov. 22, 2018	Oct. 11,18,25, Nov. 1,8,15, 2018

Table 3. Dates of DDoS traffic and normal traffic.

Attribute	Target date of evaluation	Training data
Normal	Jun. 22, 2019	Jun. 1,8,15, 2019
Normal	Jun. 23, 2019	Jun. 2,9,16, 2019
Normal	Jun. 24, 2019	Jun. 3,10,17, 2019
DDoS	Jul. 6, 2019	Jun. 8,15,22, 2019
DDoS	Jul. 7, 2019	Jun. 2,9,16,23, 2019
DDoS	Jul. 8, 2019	Jun. 3,10,17,24, 2019

22, 2018, and July 6-8, 2019 are selected as abnormal days. Using the data on those days, this paper tries to detect event traffic and DDoS attacks. On October 17, 2018, connection failure to YouTube [18] occurred. On November 22, 2018, there was a campus festival of the university that accommodates the measurement NOC. At the end of June 2019, a UDP reflection/amplification attack using ARMS (Apple Remote Management Service) was observed around the world[19]. This attack was also observed at the university. The university blocked communications for ARMS on July 9, 2019. Therefore, it is assumed that an abnormal state due to the attack was observed just before July 9, 2019. Tables 2 and 3 show the normal and abnormal dates and their training data. If the prediction model created with the data of the normal days is used to predict the data of the abnormal days, the difference between the measured data and the predicted data should be large.

Figure 12 shows the result of the evaluation. The value on top of a bar is the average NSD value of all "path aggregates" on each day. The NSD values on the days marked as "Event" (October 17 and November 22, 2018) are larger than those of the normal days. The NSD values on the days marked as DDoS attack are larger than those of the normal days. The NSD values on June 22-25 are all below 0.40, but those on July 6-8 are all above 0.43. Furthermore, the maximum NSD value for the six days is observed on July 8 (0.443), the day before the university settled the DDoS attacks. This indicates that the flows on the abnormal days cannot accurately be predicted. In other words, the behavior on the abnormal days was different from that of the normal days. This result shows that GAMPAL can detect anomalies caused by the event traffic and the DDoS attack.

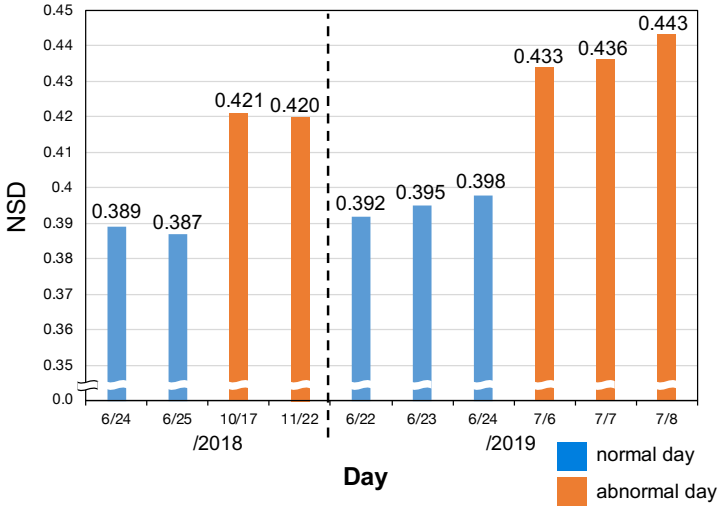

Fig. 12. Result of evaluation.

6 Conclusion

This paper proposed a general-purpose anomaly detection mechanism for Internet backbone traffic based on a LSTM-RNN-based prediction model. To make GAMPAL scalable to the number of the Internet full routes, each flow is mapped to a single path aggregates identified with the first three AS numbers of the AS_PATH attribute of the BGP RIB. This paper evaluated the validity of GAMPAL using the observed flow data and the BGP RIBs exported from the WIDE backbone network (AS2500), a nation-wide backbone network for research and educational organizations in Japan. The evaluation showed that when a stub organization of the backbone network suffers from DDoS attacks, the difference between the predicted and observed values is significantly different. Therefore, GAMPAL properly reflected the state of the Internet backbone with only the traffic throughput.

References

1. 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.
2. R. Kumar and Sharma D. HyINT: Signature-Anomaly Intrusion Detection System. In *Proc. of ICCCNT 2018*, pp. 1–7, 2018.
3. 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.
4. 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.

5. 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.
6. 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.
7. C. Petrie and T King. Multi-Threaded Routing Toolkit (MRT) Routing Information Export Format with BGP Additional Path Extensions. RFC 8050, sl IETF, 2017.
8. NAVIDIA cuDNN. `https://developer.nvidia.com/cudnn`(Last accessed 20 Aug 2019).
9. Chainer: A flexible framework for neural networks. urlhttps://chainer.org/.
10. WIDE backbone. urlhttp://two.wide.ad.jp/.
11. 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.
12. NSL-KDD dataset. `https://www.unb.ca/cic/datasets/nsl.html` (Last accessed 20 Aug 2019).
13. 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.
14. 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.
15. nfdump. `http://nfdump.sourceforge.net` (Last accessed 20 Aug 2019).
16. bgpdump. `https://bitbucket.org/ripencc/bgpdump/wiki/Home`(Last accessed 20 Aug 2019).
17. Red-Black-Tree. `https://developer.nvidia.com/cudnn` (Last accessed 20 Aug 2019).
18. TeamYoutube. `https://twitter.com/TeamYouTube/status/1052393799815589889?ref_src=twsrc%5Etfw`.
19. NETSCOUT. `https://www.netscout.com/blog/asert/call-arms-apple-remote-management-service-udp`.