GAMPALv2.5: Enhancing Large-scale Scanning Attack Inference in General-purpose Internet Traffic Anomaly Detection Mechanism

Keisuke Tachibana
Graduate School of Information Science and Technology
Hokkaido University
Sapporo, Hokkaido, Japan
tachibana.keisuke.q6@elms.hokudai.ac.jp

Takao Kondo
Information Initiative Center
Hokkaido University
Sapporo, Hokkaido, Japan
latte@iic.hokudai.ac.jp

Hiroyuki Minami
Information Initiative Center
Hokkaido University
Sapporo, Hokkaido, Japan
min@iic.hokudai.ac.jp

***Abstract*—To detect anomalies in the Internet backbone traffic, Wakui *et al.* (2025) proposed GAMPALv2 (General-purpose Anomaly detection Mechanism using Prefix Aggregate without Labeled data, version 2). It detects anomalies by comparing the predicted flow sizes with the real observation on aggregated flow traffic, using random forest regression models. It has detected YouTube service disruption, some event traffic and DDoS attacks. However, it does not have enough potential to detect a sort of initial attack activities like scanning and spam. In this paper, we extend it as GAMPALv2.5 to detect them, employing three traffic indicators, specifically flow size, packet count, and session count. We also evaluate and compare the performance with GAMPALv2 through a benchmark dataset.**

***Index Terms*—Network Traffic Analysis, Internet Backbone, Early Anomaly Detection, Random Forest**

I. Introduction

Toward the secure Internet, early detection and prevention against cyber-attack is essential as highlighted in the concept of the Cyber Kill Chain [1], a model of attacker behavior. It is important for defenders to infer it and take countermeasures at every stage to break the attack chain [2]. In particular, traffic monitoring enables early detection of malicious activities and responds quickly before large-scale cyber-attacks like DDoS.

Wakui *et al.* [3], [4] proposed an anomaly detection mechanism for the Internet backbone traffic named *GAMPAL* (*General-Purpose Anomaly Detection Mechanism using Prefix Aggregate without Labeled data*). *GAMPALv2* [4] focuses primarily on the flow size, and detects anomalies in the Internet backbone traffic with analytics of flow and routing data with machine learning (random forest regressor). They have successfully detected YouTube service disruption, some event traffic by a campus festival, and DDoS attacks.

However, for the response to cyber-attacks at the early stage, it is important to detect a broad spectrum of anomalies in the Internet backbone traffic, for instance, scanning activities and spam communications. It is tough to detect them because flow size, the key attribute of GAMPALv2, does not have enough potential to distinguish anomalies of cyber-attacks in the early stage. In this paper, we propose *GAMPALv2.5* to improve anomaly detection performance of large-scale scanning attacks by considering packet and session counts as the metrics of anomaly detection.

TABLE I
Comparison of GAMPALv2.5 and Related Works

Related Work	[5]	[6]	[7]	[9]	[10]	GAMPALv2.5
Scalability	✓	✓	-	×	×	✓
Locality	×	×	×	×	×	✓
Versatility	×	×	×	×	✓	✓
Unnecessary labeled data	✓	×	×	×	✓	✓

II. Related Work

Hwang *et al.* [5] and Aldualij *et al.* [6] propose an anomaly detection method, which make them scalable to network size. [5] focuses on early packets in a flow, and [6] does on flow-level feature extraction. A. Zaheer *et al.* [7] proposed a hybrid method to detect botnet attacks in network traffic. This model combined k-means, rule-based systems, and decision tree algorithms. While [5]–[7] detect DDoS attacks, they did not discuss anomalies due to typical irregularities like network failure. Using the hybrid mechanism based on NIN (Network in Network) [8] and CNN, Kong *et al.* [9] proposes a traffic classification method for anomaly traffic detection. While [9] achieves high detection accuracy for cyber-attacks, including DDoS attacks and port scanning attacks, it is primarily evaluated in the context of cyber-attack detection, out of the context of the no-cyber-attack anomalies such as network failures.

To detect behavioral changes in network traffic, Flanagan *et al.* [10] adopt MCODT (Micro-Cluster Outlier Detection in Time series) and CNN. It also detects non-cyber-attack anomalies, with versatility, while flow aggregation and scalability to network size have not been mentioned.

In order to adopt GAMPALv2 in the Internet backbone environment, [4] defined the following four metrics as the requirements: (i) scalability to the backbone Internet, (ii) consideration for locality of Internet traffic, (iii) versatility for any kind of anomaly, and (iv) unnecessary use of labeled learning data. Table I compares of GAMPALv2.5 and related works with respect to these requirements. GAMPALv2.5 is

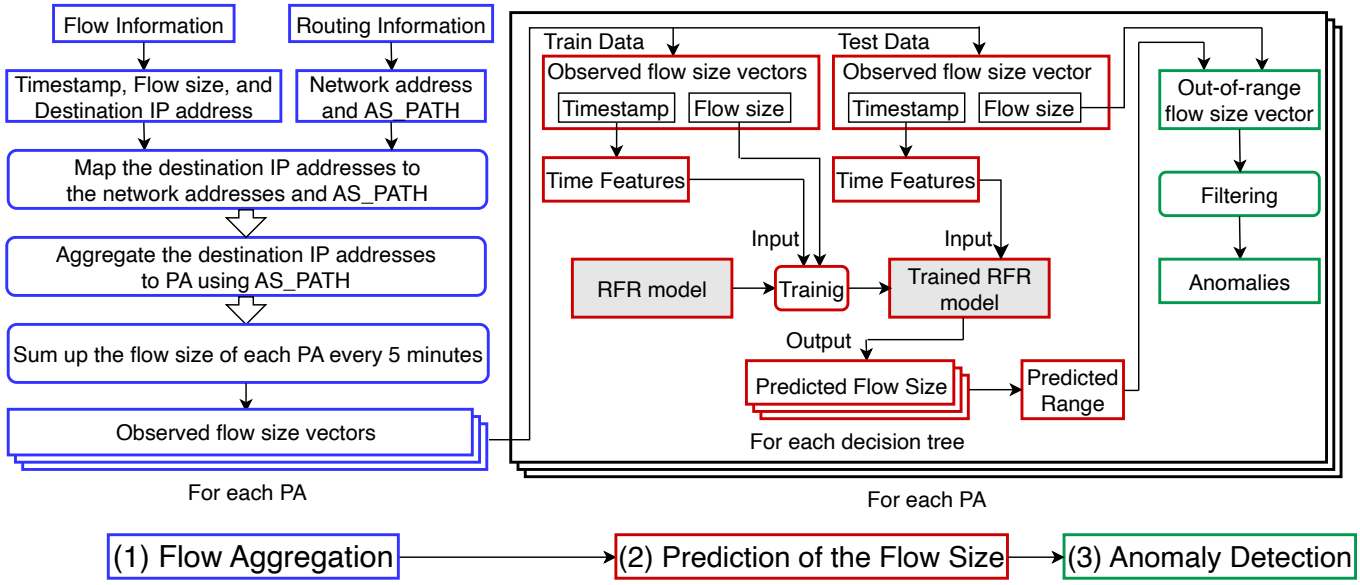

Fig. 1. GAMPALv2 Design

satisfied with these requirements and improves the performance to detect malicious activities at the early stages in the Cyber Kill Chain rather than GAMPALv2. It learns models from unlabeled flow data and detect both cyber-attacks and other anomalies, such as YouTube connection failures, which demonstrates its versatility in anomaly events.

III. Overview of GAMPALv2

A. Architecture

GAMPALv2 consists of three main processes: Flow Aggregation, Predict the flow size, and Anomaly Detection (see Fig. 1).

1) Flow Aggregation: GAMPALv2 performs flow aggregation for scalability with respect to network size and the locality of the Internet backbone traffic [11], [12]. Since the number of destination IP addresses is huge —for IPv4, it does not make sense to build 2^{32} models. Thus, GAMPALv2 aggregates flow using BGP (Border Gateway Protocol) routing information. It maps destination IP addresses (gotten by the flow information) to network addresses and AS_PATH attribute (gotten by the routing information). Then, it defines *Prefix Aggregates* (*PAs*) as the aggregation of destination IP addresses that share the same initial part (typically, the length of this part is the mode value of AS_PATH length) of AS numbers in the AS_PATH attribute in the routing information. For each PA, GAMPALv2 sums up the flow size every five minutes, and defines *observed flow size vector*, where each element is a timestamp and the corresponding flow size. The aggregation achieves a fine-grained classification of the flows that are close to the observation point in the network topology, and reduces the number of models to several 10^4.

Fig. 2 shows the example of flow aggregation. This example assumes that the mode value of AS_PATH length is 4 in the

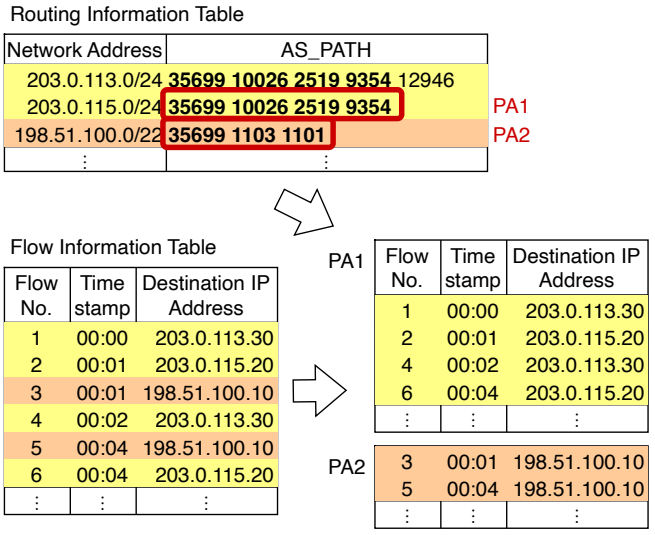

Fig. 2. Example of flow aggregation

routing information. Therefore, two PAs (PA1 and PA2 in Fig. 2) are defined to aggregate the destination IP addresses in the flow information. Fig. 2 shows the destination address and the timestamp extracted from each flow information. By mapping the destination IP address in the flow information to the network address in the routing information, the flow information is aggregated into the two PAs depicted in Fig. 2.

2) Prediction of the flow size: GAMPALv2 defines *time features* to learn the traffic periodicity, including seven variables: month, day, day of the week, week of the month, weekday or not, hour and minute. It uses Random Forest Regression (RFR) model to predict the flow size. Train data

is observed flow size vectors collected over several days (e.g., seven days), and test data is one day of observed flow size vector. The input for model training is the time features and the flow size of the train data. Using the trained model, GAMPALv2 predicts the flow size per PA every five minites of the test data based on their time features. At each PA, a *predicted range* is set based on the maximum and minimum values among the outputs of the decision trees.

3) Anomaly Detection: GAMPALv2 detects anomalies with the predicted range defined in Sec. III-A2. For the detection, it gets an out-of-range flow size vector when the observed flow size is outside the prediction range. Then, GAMPALv2 performs filtering to exclude a short out-of-range flow size vector, taking account of false positive augmentation. The filtering is composed of the following procedures. It computes a difference between the observed value and the predict range every five minutes, and apply a simple moving average with window size w. GAMPALv2 regards what it exceeds the threshold as anomalies. The threshold is defined as the product of constant α and the width of the predicted range. Consequently, GAMPALv2 detects the periods longer than the cutoff T (minutes). The concrete values of these parameters in this paper are shown in Sec. V.

B. Achievements and Problems

GAMPALv2 detected three sort of anomalies from WIDE BackBone (WIDE-BB; AS2500) [13] traffic, including YouTube connection failure, event traffic by a campus festival, and DDoS attacks. It accomplished 90% recall on DDoS attacks with the dataset for evaluation, however, it might bring less performance on the early activities of Cyber Kill Chain such as Reconnaissance and Delivery (e.g., scanning and spam attack).

IV. GAMPALv2.5 Design

We design GAMPALv2.5, an extended version of GAMPALv2, that predicts packet and session counts, in addition to the flow size. It employs random forest regression models corresponding to the three quantities. It specifies the anomaly period based on the derived outputs by the models. With the extension, we expect to be able to detect attacks that were difficult to identify using only flow size, such as scanning and spam. Fig. 3 shows GAMPALv2.5 overview. We enhance the performance with the emphasized parts with bold line in the Fig. 3.

This section describes the architecture of GAMPALv2.5 from the viewpoint of the three aspects: flow aggregation, prediction of the flow size, packet and session counts, and anomaly detection. These three aspects correspond to the design of GAMPALv2 (see Sec. III-A).

A. Flow Aggregation

Same as GAMPALv2, GAMPALv2.5 handles flow size. It also treats packet count in the same way as flow size. It accumulates session count for each PA. Flows with identical 5-tuples (source IP address, destination IP address, source port number, destination port number, and protocol) are treated as a same session. Then, it sums up each quantities per PA every five minutes and defines *observed packet count vector* and *observed session count vector* in additon to observed flow size vector. They have a timestamp and corresponding quantity.

Fig. 4 shows the example of observed vectors. The assumed flow information and routing information is the same as shown in Fig. 2. Fig. 4 shows the aggregated flows in the PA1 and the PA2, in which each flow information includes the 5-tuples (the source IP address, the destination IP address, the source port number, the destination port number, and the protocol), the flow size, and packet count in the flow. In the observed flow seize vectors, each PA has a sum of flow sizes (e.g., 10, 20, 30, and 40 bytes for the PA1) over the 5-minute period from 00:00 to 00:05. In the observed packet count vectors, each PA has a sum of packet count (e.g., 1, 2, 3, and 4 packets for the PA1) over the 5-minute period from 00:00 to 00:05. In the observed session count vectors, each PA has a sum of session count over the 5-minute period from 00:00 to 00:05. In the session count, flows with the same 5-tuples within the same 5-minute period are considered a session. In Fig. 4, the flows with the same color are considered a session. Therefore, the PA1 has two sessions (the blue and red ones) in the 5-minute period from 00:00 - 00:05. The PA2 has a session (the green one) in the 5-minute period from 00:00 - 00:05.

B. Prediction of the flow size, packet and session counts

GAMPALv2.5 separately handles flow size, packet count, and session count. Thus, it utilizes three RFR models per PA, each corresponding to one of the quantities. Each model inputs the time features and quantities to train, and outputs the predicted value for the test data. We set a predicted range per model just same as Sec. III-A2. After the processes, we get three predicted ranges for each PA, which indicate flow size, packet count, and session count.

C. Anomaly Detection

In GAMPALv2.5, we detect anomalies for each quantity and take the union of the results. We compare the prediction ranges on each PA with the observed vectors of the test data, to specify the three out-of-range vectors respectively. After that, we apply filtering (in Sec. III-A3) to them and regard the outputs as anomalies. In this paper, we use fixed filtering parameters for each quantity, while we later discuss the possibility of improving them in Sec. VI-B4. Each of the above steps is performed independently for the three quantities. As a result, we identify anomaly period per PA as a union of the outputs. For example, at a certain time, suppose that the flow size and packet count are within the predicted ranges but the session count are detected as anomalous. In this case, GAMPALv2.5 detects anomalies at the corresponding time. The same applies when only the flow size or packet count are detected as anomalous.

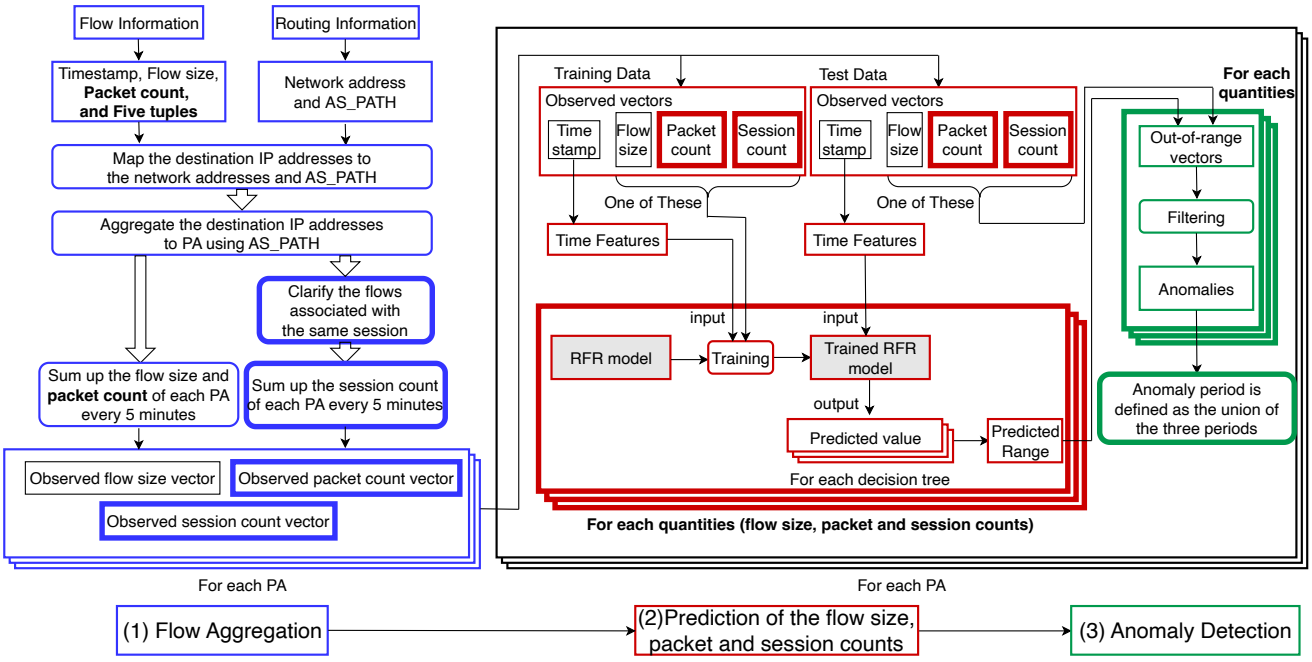

Fig. 3. GAMPALv2.5 Design

PA1

Flow No.	Time stamp	Source IP Address	Destination IP Address	Source Port number	Destination Port Number	Protocol	Flow Size (bytes)	Packet Count
1	00:00	IP1	203.0.113.30	PORT1	PORT4	TCP	10	1
2	00:01	IP2	203.0.115.20	PORT2	PORT4	TCP	20	2
4	00:02	IP1	203.0.113.30	PORT1	PORT4	TCP	30	3
6	00:04	IP2	203.0.115.20	PORT2	PORT4	TCP	40	4
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

PA2

3	00:01	IP1	198.51.100.10	PORT2	PORT5	UDP	100	10
5	00:04	IP1	198.51.100.10	PORT2	PORT5	UDP	200	20
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Observed Flow Size Vectors

	First Four AS number	00:00	00:05
PA1	35699 10026 2519 9354	100	...
PA2	35699 1103 1101	300	...
	⋮	⋮	⋱

Observed Packet Count Vectors

	First Four AS number	00:00	00:05
PA1	35699 10026 2519 9354	10	...
PA2	35699 1103 1101	30	...
	⋮	⋮	⋱

Observed Session Count Vectors

	First Four AS number	00:00	00:05
PA1	35699 10026 2519 9354	2	...
PA2	35699 1103 1101	1	...
	⋮	⋮	⋱

Fig. 4. Example of Observed Vectors

V. Implementation

A. Environment

We implement GAMPALv2.5 in Ubuntu 22.04 server, Intel Xeon (24 cores at 2.50 GHz), 128GB RAM, and Python 3.7.8.

B. Dataset

For evaluation, we use UGR16 dataset [14], collected from the NetFlow V9 [15] collector of a tier-3 ISP in Spain. It includes labels such as normal, SSH scan attack, and spam attack, while we do not use them in the models. In addition, we use *RRC18* [16], IPv4 route information observed at *Routing Information Service* (*RIS*) [17] by RIPE NCC on March 1, 2016, in Barcelona (Spain). It contains BGP full route information.

C. Setup for Evaluation

Since the mode value on AS_PATH length in RRC18 is four, we aggregate the flows according to the first four numbers in the attribute. Just same as GAMPALv2, we adopt RandomForestRegressor [18] from scikit-learn and set the number of decision trees to 200 and the maximum depth to 12. We code time features as follows: 1-Monday , 2-Tuesday, ..., 7-Sunday. If weekday, the flag is set to 1, otherwise 0. As the filtering parameters (see Sec. III-A3), we use $w = 3$, $\alpha = 0.15$, $T = 30$. Wakui *et al.* examined α by

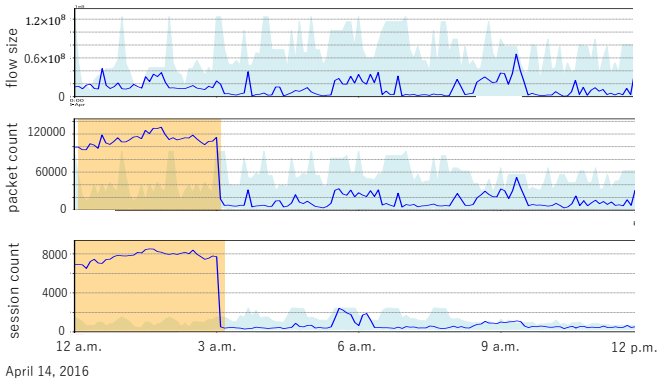

Fig. 5. Observed values and prediction ranges for the PA "49835 2914 209"

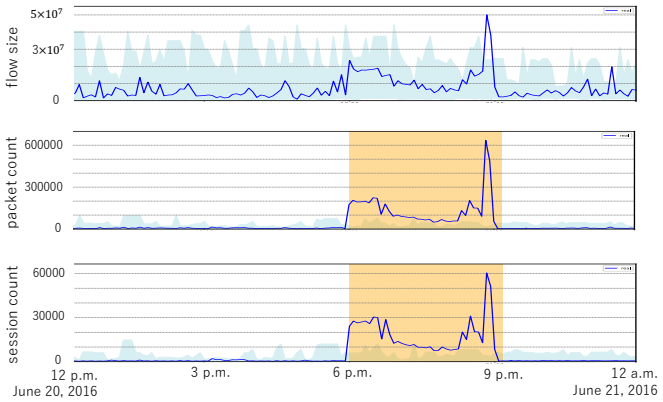

Fig. 6. Observed values and prediction ranges for the PA "49835 2914 2914 3292"

comparing multiple values and we use the value that achieved the best performance. We use the same values for the other parameters as those employed in GAMPALv2.

VI. Evaluation

We attempt to detect early-stage attacks contained in UGR16 dataset, and evaluate the anomaly detection performance with our own verification dataset from UGR16.

Since UGR16 dataset is based on real network traffic traces, the number of attack labels varies on the observation date. In this paper, we focus on SSH scan attacks on April 14, 2016, and spam attacks on June 20, 2016 as they were observed frequently on the days. In addition, we use seven days of flow data before the attack as training.

In the following evaluation, the flow size entries in the GAMPALv2.5 results can be interpreted as the baseline performance of GAMPALv2.

A. Detecting early-stage attacks

In this section, we focus on PA that showed notable activity, and attempt to detect it.

1) Detecting SSH scan attacks: Fig. 5 shows the observation and predicted ranges of the flow size (top), packet (middle), and session (bottom) counts for the PA "49835 2914 209" from midnight to noon on April 14, 2016.

The blue solid line in the figure indicates the actual observation, the light blue area is the predicted range, and the orange area is the anomaly period detected by GAMPALv2.5. From midnight to 3 a.m., GAMPALv2.5 detects anomalies on packet and session counts, but not on flow size. UGR16 dataset observed SSH scans during that time, and GAMPALv2.5 detects the anomaly while the previous version missed them.

2) Detecting spam attacks: Fig. 6 shows the observation and predicted ranges for the PA "49835 2914 2914 3292" from noon on June 20, 2016 to midnight on June 21, 2016 (same illustrated as Fig. 5). GAMPALv2.5 detects anomaly in packet and session counts between 6 p.m. and 9 p.m., never in flow size. During this period, UGR16 dataset has records on spam attack, and GAMPALv2.5 also succeeds in the detection.

Both results holds the efficacy on GAMPALv2.5.

B. Evaluation of the anomaly detection performance

UGR16 dataset has labels assigned to flows. Because the lengths of attack and non-attack periods vary, it is difficult to evaluate anomaly detection performance using evaluation quantities. In this paper, we make a dataset for evaluation based on GAMPALv2 and assess the performance of GAMAPLv2.5 to detect SSH scan and spam attacks. Since it is impossible to distinguish anomaly by attack from others, we apply GAMPALv2.5 to flows labeled as normal to eliminate PAs that contain unlabeled anomalies. Through the process, we treat the PAs considered normal as truly anomaly-free, and use them as the evaluation (We named these PAs as "*target PA*").

1) Dataset for evaluation: We make a dataset for evaluation by inserting attack flows, whose scale is adjusted to match the normal flows, shown in Fig. 7. We insert five (similar to GAMPALv2) hours of simulated attacks into one day's flows, starting at 1 a.m., 6 a.m., 11 a.m., 4 p.m., and 9 p.m. (Step A in Fig. 7). We perform scale adjustment using $A_{base} = \frac{1}{12}(a_{00} + a_{05} + \cdots + a_{55})$ and $B_{base} = \frac{1}{12}(b^m_{00} + b^m_{05} + \cdots + b^m_{55})$. Furthermore, s is a scale adjustment parameter to control simulated attack flow. In this paper, we set $s = 1$. Here, if $B_{base} = 0$, we cannot insert the attack flow. Then, we set the lower limit of scale adjustment attack flows $(at'_{00}, at'_{05}, \cdots, at'_{55})$ to the smallest positive value. On the other hand, during other times, we use the normal flows (Step B in Fig. 7).

2) Anomaly detection evaluation of SSH scan attacks: We use flows from UGR16 dataset observed on April 14, 2016. The number of target PAs is 13,270, and a total of $66,350$ $(13,270 \times 5)$ simulated attacks are inserted. We check for anomalies at 30 minutes past each hour throughout the day, and classify the results into True Positive (TP), False Negative (FN), False Positive (FP), and True Negative (TN). Furthermore, we count the number of PAs that have n True Positives $(n = 0, 1, \cdots, 5)$.

Table II, III and IV show the results. The first three columns show the results based on each quantity individually, and the rightmost column shows the result of taking the union (see

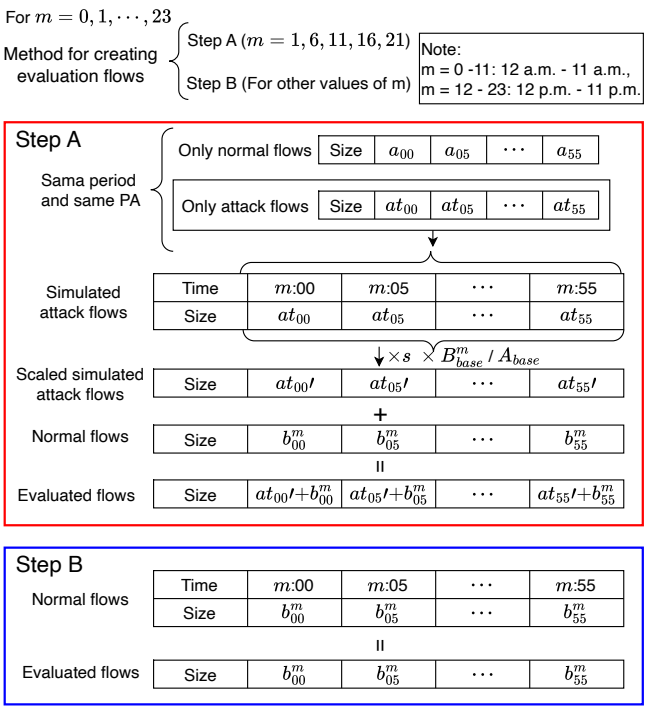

Fig. 7. Create a dataset for evaluation

TABLE II
ANOMALY DETECTION RESULTS OF SSH SCAN ATTACKS

	Flow Size	Packet Count	Session Count	Union
TPs	11,862	38,124	49,313	50,180
FNs	54,488	28,226	17,037	16,170
FPs	144	543	577	1,133
TNs	251,986	251,587	251,553	250,997

Sec. IV-C) of the anomaly detection results. Compared to flow size, the number of TPs approximately increased by 3.2 times for packet count and by 4.1 times for session count. In addition, the number of PAs with five TPs was substantially higher for new quantities. Furthermore, for flow size, over half of the PAs failed to detect all simulated attacks, whereas the amount was much smaller for packet and session counts. The F1 score doubled with the addition of new quantities.

The results prove that new populated quantities contribute to improve detection performance.

3) Anomaly detection evaluation of spam attacks: We investigate the performance for spam attacks same as the previous one. We use flows from UGR16 dataset observed on June 20, 2016. The number of target PAs is 17,605, and a total of $88,025$ $(17,605 \times 5)$ simulated attacks are inserted.

Table V, VI and VII show the results. What each column represents is explained in Sec. VI-B2. As with the results for the SSH scan attack, the addition of new quantities led to an improvement in detection performance. The F1 score increased over four times, indicating higher performance compared with

TABLE III
ANOMALY DETECTION RESULTS OF SSH SCAN ATTACKS

Number of PAs with n True Positives	Flow Size	Packet Count	Session Count	Union
$n = 5$	271	2,841	4,732	5,022
$n = 4$	706	2,620	3,592	3,602
$n = 3$	920	2,246	2,450	2,316
$n = 2$	1,261	2,316	1,623	1,544
$n = 1$	2,401	2,069	689	626
$n = 0$	7,711	1,178	184	160

TABLE IV
ANOMALY DETECTION EVALUATION OF SSH SCAN ATTACKS

	Flow Size	Packet Count	Session Count	Union
Accuracy (%)	82.85	90.97	94.47	94.57
Precision (%)	98.80	98.60	98.84	97.79
Recall (%)	17.88	57.46	74.32	75.63
FP rate (%)	0.06	0.22	0.23	0.45
F1 score	0.30	0.73	0.85	0.85

the detection of SSH scan attacks.

The results also show that detection performance improvements from the inclusion of new quantities, but both results indicate that the recall is at most 75%. We still lose several attacks, and it is a sort of our future works.

4) Considerations about Filtering Parameters: To evaluate the improvement of performance, we use the same parameters in GAMPALv2. We also use the same values for three quantities. We discuss the issues arising from applying the same α (see Sec. III-A3) to the three quantities, and propose feasible improvements to the parameter tuning.

In Fig. 5 and Fig. 6, each quantity has its original scale. Then, the difference between the maximum and minimum of

TABLE V
ANOMALY DETECTION RESULTS OF SPAM ATTACKS

	Flow Size	Packet Count	Session Count	Union
TPs	6,127	37,060	47,049	48,733
FNs	81,898	50,965	40,976	39,292
FPs	71	257	277	532
TNs	334,424	334,238	334,218	333,963

TABLE VI
ANOMALY DETECTION RESULTS OF SPAM ATTACKS

Number of PAs with n True Positives	Flow Size	Packet Count	Session Count	Union
$n = 5$	15	1,454	3,252	3,439
$n = 4$	97	2,263	2,663	2,830
$n = 3$	415	3,119	3,273	3,381
$n = 2$	896	3,619	3,562	3,549
$n = 1$	2,627	4,143	3,194	2,977
$n = 0$	13,555	3,007	1,661	1,429

TABLE VII
ANOMALY DETECTION EVALUATION OF SPAM ATTACKS

	Flow Size	Packet Count	Session Count	Union
Accuracy (%)	80.60	87.88	90.24	90.57
Precision(%)	98.85	99.31	99.41	98.92
Recall (%)	6.96	42.10	53.45	55.36
FP rate (%)	0.02	0.08	0.08	0.16
F1 score	0.13	0.59	0.70	0.71

the prediction range also varies by quantities. If we adopt the same unique α for all quantities, FP and FN would increase.

Thus, we present several approaches to improve the parameter α. One approach is to configure α on each quantity independently. Another is to standardize the scales of the metrics and apply the same α. Dynamic reduction for α, for instance, based on statistical consideration including mean and variance on each quantity would be effective. These refinements is our future work since any ideas must pay extra costs.

VII. CONCLUSION

In this paper, we aim to detect exploring attacks, which were not effectively by GAMPALv2 [4], as preliminary steps taken by attackers. To this end, we develop GAMPALv2.5, an expansion of GAMPALv2. To improve detection performance, we introduce some additional quantity, packets and session counts. We apply the same approach to the target per quantities and gather the outputs.

We attempt to detect SSH scan attacks and spam attacks, and evaluate the anomaly detection performance. As a result, we prove the improvement with our new version. We still have some future works regarding detection capability.

REFERENCES

[1] Lockheed Martin, "Cyber Kill Chain," Last Accesed September 12, 2025. [Online]. Available: https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html

[2] MITRE Corporation., "MITRE ATT&CK," Last Accesed September 12, 2025. [Online]. Available: https://attack.mitre.org

[3] T. Wakui, T. Kondo, and F. Teraoka, "GAMPAL: an anomaly detection mechanism for Internet backbone traffic by flow size prediction with LSTM-RNN," *Annales des Telecommunications/Annals of Telecommunications*, vol. 77, no. 5-6, pp. 437–454, Jun. 2022.

[4] T. Wakui, F. Teraoka, and T. Kondo, "GAMPALv2: An Anomaly Detection Mechanism for Internet Traffic by Predicting Flow Size Range from Time Features," *IEICE Trans. on Info. and Sys.*, vol. E108-D, no. 6, pp. 505–516, 2025.

[5] R. Hwang, M. Peng, C. Huang, P. Lin, and V. Nguyen, "An Unsupervised Deep Learning Model for Early Network Traffic Anomaly Detection," *IEEE Access*, vol. 8, pp. 30 387–30 399, 2020.

[6] M. Alduailij, Q. W. Khan, M. Tahir, M. Sardaraz, M. Alduailij, and F. Malik, "Machine-Learning-Based DDoS Attack Detection Using Mutual Information and Random Forest Feature Importance Method," *Symmetry*, vol. 14, no. 6, 2022.

[7] A. Zaheer, S. Tahir, M. F. Almufareh, and B. Hamid, "A Hybrid Model for Botnet Detection using Machine Learning." in *In Proc. of ICBATS 23'*, 2023, pp. 1–8.

[8] M. Lin, Q. Chen, and S. Yan, "Network In Network," 2014. [Online]. Available: https://arxiv.org/abs/1312.4400

[9] Y. Li, X. Kong, J. Hou, X. Li, K. Zhao, W. Liang, T. Jiang, Y. Xin, and Z. Liu, "NIN-DSC: A Network Traffic Anomaly Detection Method Based on Deep Learning," in *In Proc. of ICSI 22'*, 2022, pp. 390–394.

[10] 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 *In Proc. of IEEE CCNC '19*, 2019, pp. 1–7.

[11] C. Williamson, "Internet traffic measurement," *IEEE Internet Computing*, vol. 5, no. 6, pp. 70–74, 2001.

[12] M. Barthelemy, B. Gondran, and E. Guichard, "Spatial structure of the internet traffic," *Physica A: Statistical Mechanics and its Applications*, vol. 319, pp. 633–642, 2003.

[13] "WIDE backbone." [Online]. Available: http://two.wide.ad.jp/

[14] G. Maciá-Fernández, J. Camacho, R. Magán-Carrión, P. García-Teodoro, and R. Therón, "UGR'16: A new dataset for the evaluation of cyclostationarity-based network IDSs," *Computers & Security*, vol. 73, pp. 411–424, 2018.

[15] B. Claise, "Cisco Systems NetFlow Services Export Version 9," *IETF*, RFC 3954, 2004.

[16] RIPE NCC, "Route Collectors, RRC18," March 1, 2016 Midnight. [Online]. Available: https://data.ris.ripe.net/rrc18/2016.03/bview.20160301.0000.gz

[17] ——, "Routing Information Service (RIS)." [Online]. Available: https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris/

[18] scikit-learn, "RandomForestRegressor." [Online]. Available: https://scikit-learn.org/1.5/modules/generated/sklearn.ensemble.RandomForestRegressor.html