A Dissertation for the Degree of Ph.D. in Engineering

GAMPAL: A General-Purpose Anomaly Detection Mechanism for Internet Backbone Network without Labeled Data

January 2025

Graduate School of Science and Technology,
Keio University

Taku Wakui

Abstract

The Internet backbone network is a large-scale network that interconnects various types of networks and is potentially affected by anomalies such as network equipment failures, cyberattacks, social phenomena, etc. To operate the Internet backbone network stably, anomaly detection technology is necessary to deal with such anomalies quickly. However, various types of network traffic generated from various users and services are observed in the Internet backbone network. In addition, the traffic pattern is peaky and jaggy, changing every moment, even during ordinary times. Therefore, detecting various anomalies only by monitoring traffic patterns is difficult. In existing research on anomaly detection techniques for Internet traffic, only some methods consider scalability for large-scale networks or network traffic features. In addition, most methods learn anomaly features using labeled data in which anomalies are identified in advance. For example, anomaly detection methods for botnet and SDN (Software-Defined Networking) architecture have been proposed.

However, labeled data is difficult to generate and limits the detection target to anomaly types defined by labels. As described above, existing methods are unsuitable for anomaly detection for the Internet backbone networks.

This dissertation proposes GAMPAL (General-purpose Anomaly detection Mechanism using Prefix Aggregation without Labeled data), an anomaly detection mechanism for the Internet backbone network. GAMPAL aims to detect anomalous traffic behavior in the Internet backbone network generally and efficiently and prompt early response to the anomaly.

GAMPAL uses network traffic flow as traffic information, which can be analyzed scalably. A flow is traffic data aggregating a packet sequence with a common five-tuple (source IP address, destination IP address, source port, destination port, and protocol number). The number of flows observed in the Internet backbone network is enormous and changes rapidly even in ordinary times. Therefore, flows must be aggregated into multiple classes

and analyzed for each aggregation unit. GAMPAL defines prefix aggregation, a flow aggregation method using the BGP RIB (Border Gateway Protocol Routing Information Base) and PA (Prefix Aggregate), a unit of aggregated flows based on prefix aggregation. Focusing on the locality of Internet traffic, flows are aggregated spatially based on the part of the AS (Autonomous System) path by prefix aggregation. This aggregation method is suitable for the locality of network traffic. In addition, to aggregate flows temporally, a five-minute time slot is applied to flows for each PA, and time series data of the flow size, which is the sum of the bytes of the flows observed during the five minutes, is generated. Using this time series data of flow size, GAMPAL establishes models to predict the flow size for each PA. GAMPAL detects various types of anomalies by comparing the predicted flow size with the observed flow size and evaluating the influence of the anomaly on the flow. This method does not require labeled data in which anomalies are identified for each flow or packet in advance.

GAMPAL has been proposed two versions: GAMPALv1 and GAMPALv2. GAMPALv1 established a model in which the time series data of flow size is the explanatory variable, and the subsequent flow size is the objective variable. Considering the periodicity of Internet traffic, such as daily or weekly, GAMPALv1 adopted a model based on LSTM-RNN (Long Short-Term Memory-Recurrent Neural Network), suitable for predicting time series data. In addition, to evaluate the difference between the predicted and observed flow sizes, GAMPALv1 defined NSD (Normalized Summation of Difference). This indicator represents the ratio of the difference between two time series data.

In the evaluation of GAMPALv1, a connection failure, event traffic, and a cyberattack are used as the evaluation targets, with data for the days on which each anomaly occurred and for the days on which anomalies were not reported. To evaluate anomaly detection performance, NSD for each PA is calculated using predicted and observed values for one day, and the average of NSD for all PAs is calculated. The evaluation results showed that the average of NSDs was larger on days when an anomaly was reported. This result indicates that GAMPALv1 can detect anomalies.

However, GAMPALv1 has four issues. First, the computational com-

plexity of training the prediction model and predicting the flow size is enormous. Second, the semantics related to time cannot be utilized to learn the model. Third, it is difficult to define the threshold for detection. Fourth, it is impossible to identify when and in which PAs an anomaly is detected. These issues prevent the practical use of GAMPALv1.

To address the first and second issues, GAMPALv2 defines the time feature extracted from the time attribute of each time slot, such as month, day, hour, minute, and weekday or not. It proposes a method for modeling the relationship between the time feature and flow size. For the prediction model, RFR (Random Forest Regressor), suitable for new learning approaches, is adopted. The number of dimensions of the variables handled by the prediction model is reduced from 288 in GAMPALv1 to 7 in GAMPALv2. Since RFR has less computational complexity than LSTM-RNN, the computational complexity of the model is significantly reduced. For the third and fourth issues, GAMPALv2 defines a predicted range that can be calculated from the prediction results from the model and proposes anomaly detection through a comparison of the predicted range and the observed values.

In GAMPALv2, the predicted ranges are calculated for the days when the three anomalies discussed in GAMPALv1 occurred and for the days when no anomalies were reported and are compared with the observed values. As a result, the observed values are out of the predicted range in the PAs, to which the traffic is destined for the university network, where the observation point is located only on the days when the anomalies are confirmed. This result indicates that GAMPALv2 can detect anomalies. In the accuracy evaluation using a dataset generated with labeled data, the recall is 93.1%. Although the evaluation method and dataset differed, the recall improved from 81.8% in GAMPALv1. In addition, GAMPALv1 requires approximately four days on a GPU to train the model and calculate the predicted flow sizes for one day, while GAMPALv2 takes approximately 23 minutes on an 8-core CPU. Through these evaluations, this dissertation confirms that GAMPAL improves its detection accuracy and practical utility.

Acknowledgments

First, I would like to express my profound appreciation to my main supervisor, Prof. Fumio Teraoka. Prof. Teraoka has guided my research and encouraged me in my work and research activities. I am grateful to have joined Teraoka Laboratory.

I would also like to express my profound appreciation to the vice-chairs of my dissertation committee: Prof. Hiroshi Shigeno, Prof. Hiroaki Nishi, and Prof. Kensuke Fukuda. I was able to complete my dissertation thanks to precise advice from all of them.

I would also like to thank Asst. Prof. Takao Kondo (Hokkaido University), a.k.a. latte-san. Thank you very much for accompanying me in this research from the beginning. My research activities have been very fruitful, thanks to latte-san.

I would also like to thank Asst. Prof. Kosuke Mori, a.k.a. tomas-san. Tomas-san has always supported me and lab members kindly.

Finally, I would like to express my appreciation to my family, friends, colleagues, and lab mates.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

The Internet backbone network is a large-scale network that interconnects various networks [1]. Since the Internet backbone network connects to various networks, it can be affected by various anomalies such as failures, social phenomena, and cyber attacks. For example, failures include unexpected outages in the service of telecommunications carriers and primary services. Anomalies caused by social phenomena include flash crowds when users and accesses are concentrated due to real-world events. Cyber attacks include DoS (Denial of Service) attacks, DDoS (Distributed Denial of Service) attacks, vulnerability exploitation, spam mail, port scan attacks, and others. Among them, the failure of services with many users and large-scale cyber attacks such as DDoS attacks can affect a wide area, including many users and organizations. When such anomalies occur, it is important to notice

them and deal with them as soon as possible. In addition, various types of network traffic generated by various services and users are observed in the Internet backbone network. Since traffic with various features is intermingled, the traffic pattern is peaky and jaggy and changes rapidly even during ordinary times. These features of the Internet backbone network make it difficult to detect anomalies only by observing traffic patterns.

The existing anomaly detection mechanisms for Internet traffic are classified into two approaches: signature-based and anomaly-based methods. Signature-based methods use a pre-defined list of signatures, which are features of known threats. Signatures are generated by analyzing the features of attacks, malware, and failures that have previously appeared. As a signature-based anomaly detection system monitors Internet traffic, it compares a packet or flow with the signatures and checks if it matches any known anomalies [2–9]. Signature-based methods are suitable for detecting known anomalies and real-time anomaly detection even for a large amount of traffic, such as the Internet backbone network traffic. However, these methods are unsuitable for detecting unknown anomalies such as new types of attacks. Anomaly-based methods use ML (Machine Learning) to learn the features of ordinary or anomalous Internet traffic behavior and are suitable for detecting unknown anomalies [10–20]. Most existing mechanisms use labeled data in which the existence of anomalies is indicated. However, it is difficult to collect such data. In addition, labeled data might cause overfitting because they have already defined a finite number of anomaly types. For example, a model

trained using labeled data for intrusions cannot detect failure in a service. Moreover, since anomalies are hidden in other regular traffic, it is difficult to detect them by observing the behaviors of various kinds of traffic. Furthermore, most of the existing mechanisms are specific to particular anomalies or in particular environments. For example, the anomaly detection methods in [10] and [17] are for a cloud environment, and the system in [11] is for the SDN (Software-Defined Networking) architecture. [13] aims to detect botnet attacks, and [14] detects malicious traffic in the evaluation. [17], [19] and [20] propose the intrusion detection system for cyber security.

To operate the Internet backbone network stably, a general-purpose mechanism for detecting various anomalies in traffic information must be established. For this problem, this dissertation proposed an anomaly detection mechanism for Internet traffic named *GAMPAL (General-purpose Anomaly detection Mechanism using Prefix Aggregation without Labeled data)* [21] [22]. GAMPAL aims to detect anomalous traffic behavior in the Internet backbone network generally and efficiently and to provide early warnings to operators and subscribers. Detecting anomalies earlier is expected to identify the cause of the failure and block the traffic from the network that caused the anomalies. First, GAMPALv1 [21] is proposed, and then GAMPALv2 [22] is proposed as an improved version that solves the issues of GAMPALv1.

1.2 Requirements for Anomaly Detection

This dissertation defines four requirements for anomaly detection in the Internet backbone networks.

The first requirement is scalability. The order of the number of IP addresses in BGP full routes is enormous, $O(10^9)$ in the case of IPv4. Anomaly detection in Internet backbone networks requires analyzing a huge network space scalably.

The second requirement is to consider the locality of Internet traffic. The locality of Internet traffic is the tendency to observe more traffic destined to IP addresses that are closer in route to the observation point. This characteristic is one of the important features of Internet traffic, although it is rarely focused on anomaly detection.

The third requirement is versatility. The Internet backbone network can be affected by various anomalies. Each anomaly is different in its characteristics and scale. As described in Sect. 1.1, for the stable operation of the Internet backbone network, it is necessary to notice anomalies that may disrupt Internet operations as soon as possible. For example, failures in primary services and large-scale cyber attacks must be detected on a priority basis. However, anomalies such as spam mail and port scan attacks have a low priority for response because they are small-scale and do not have a large influence.

The fourth requirement is not to use labeled data. Labeled data has been labeled as benign or not for each packet or flow in advance and is difficult to generate. In addition, anomaly detection using labeled data may limit the detection target to anomalies defined by the data because most labeled data is specific to a particular anomaly. For example, the NSL-KDD [23] is a dataset specific to network intrusions, and the CIC-DDoS2019 [24] is a dataset specific to DDoS attacks.

1.3 Contributions

This dissertation presents four contributions based on the achievements of GAMPALv1 and GAMPALv2.

The first contribution is the introduction of prefix aggregation, which allows traffic information to be aggregated spatially and temporally. This contribution enables the scalable analysis of the enormous traffic information observed in the Internet backbone network.

The second contribution is to show that it is possible to predict the amount of traffic in a time slot based on the time feature generated from the time slot. Using semantics related to time improves prediction accuracy, and reducing the number of dimensions of the explanatory variables reduces computational complexity.

The third contribution is to show that the range of predicted traffic amounts can be defined by adopting RFR (Random Forest Regressor) as

the prediction mechanism, and the PA (Prefix Aggregate) and the period in which the anomaly occurred can be identified by comparing the observed values with the range of predicted values.

The fourth contribution is to show that GAMPAL can detect failures, event traffic, and DDoS attacks. This contribution enables the stable operation of the Internet backbone network, which can be affected by various anomalies.

1.4 Position of GAMPAL

Figure 1.1 shows the research fields related to Internet operations and the relationship among them. This dissertation identifies six research fields related to Internet operations: Traffic engineering, Internet traffic collection, Network monitoring, Internet traffic prediction, Anomaly detection, and Network management. Each research field mutually utilizes the other research fields. These research fields are closely related, and some research is multidisciplinary. For example, Internet traffic prediction can be used for long-term network management, such as network maintenance and provisioning. Among them, GAMPAL belongs to the field of anomaly detection, which leverages techniques in the field of Internet traffic prediction. Anomaly detection utilizes Internet traffic collection and network monitoring and may utilize Internet traffic prediction. It is utilized for network management, such as early response to attacks and failures, and traffic engineering, such

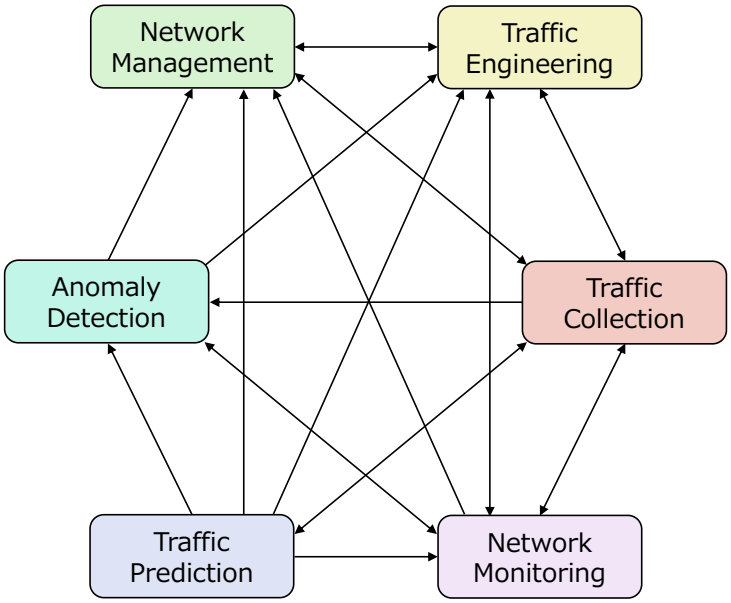


Figure 1.1: Relationship of the research fields for Internet operations.

as network configuration and redesign.

Anomaly detection utilizes Internet traffic collection and network monitoring, and may utilize traffic prediction. It is also utilized for network management, such as early response to attacks and failures, and traffic engineering, such as network configuration and redesign.

Figure 1.2 shows the position of GAMPAL and other research. In this distribution plot, the horizontal axis represents computational complexity, and the vertical axis represents versatility. Versatility in this context indicates that the method is not specific to a particular anomaly or environment. Most anomaly detection methods for the Internet can be classified into two categories: signature-based [2–9] and anomaly-based [10–20].

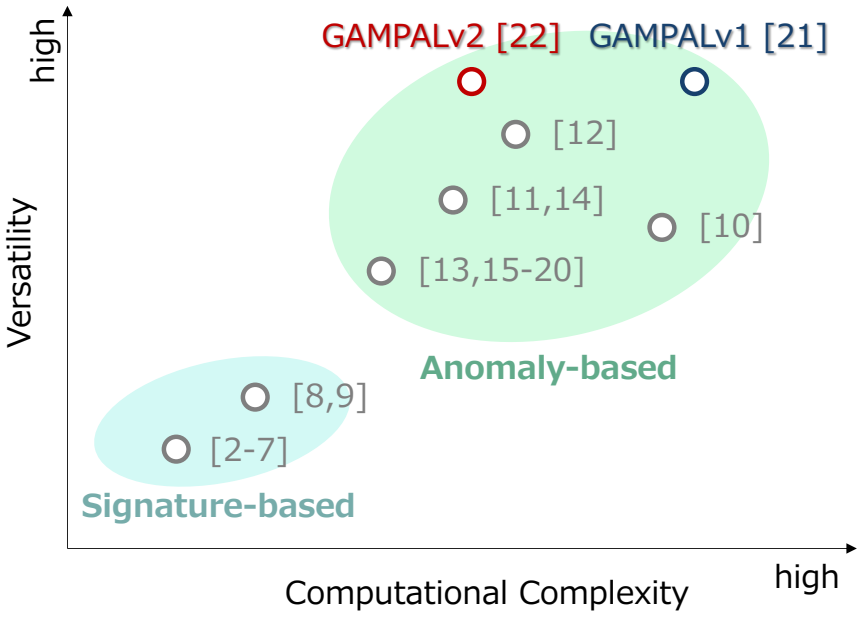


Figure 1.2: Position of GAMPAL.

As described in Sect. 1, signature-based methods are suitable for detecting known anomalies. Signature-based methods have low computational complexity and detect known anomalies. Therefore, these methods are used for cybersecurity, such as intrusion detection, and are low in versatility. Since it is common to monitor each network, signature-based methods are often used for small-scale networks. Some methods apply to middle-size networks or propose to define signatures using statistical analysis.

Anomaly-based detection methods use ML techniques. For example, classification models and clustering techniques are used to classify normal and anomalous packets [11, 12, 14, 16, 19]. In addition, regression models are used to analyze traffic metrics [10]. GAMPAL is classified into anomaly-based de-

tection methods. In general, anomaly-based detection methods can handle unknown threats better than signature-based detection methods. However, they have the disadvantages of a high false positive rate and a large computational complexity. The high false positive rate has been improved in recent years by highly accurate classification methods, combined use with signature-based methods [17, 18], and the use of multiple ML techniques [12, 13, 20]. However, the computation time must be acceptable for practical use.

Although GAMPAL belongs to the field of anomaly detection, it is also related to the field of Internet traffic prediction. Internet traffic prediction can be broadly divided into two categories depending on the time granularity of the time series data: fine-grained prediction [25–27] and coarse-grained prediction [28]. The former predicts changes in time windows of a few minutes to a few dozen minutes and can be used for anomaly detection, intrusion detection, and short-term provisioning. On the other hand, the latter predicts changes in unit hours to a day. Coarse-grained predictions are used for long-term network management and planning IT infrastructure expansions. The fine-grained method generally needs more computational complexity and is difficult to predict.

As described above, GAMPAL belongs to the field of anomaly detection, which leverages techniques in the field of Internet traffic prediction. GAMPAL detects anomalies through fine-grained traffic prediction and provides insight into the time anomalies are detected. GAMPALv1 shows high versatility, but has a high computational complexity. GAMPALv2 reduces

the computational complexity while maintaining high versatility. In addition, this dissertation verifies that GAMPAL can actually detect various anomalies.

1.5 Structure of Dissertation

This dissertation consists of nine chapters. Chapter 1 describes the background and objectives of this research and the requirements for anomaly detection in the Internet backbone networks. Chapter 2 describes related works and compares them with this research regarding the requirements. Chapter 3 describes the GAMPAL architecture common to GAMPALv1 and GAMPALv2. Chapter 4 describes the design and evaluation results of GAMPALv1. Chapter 5 describes the issues of GAMPALv1 and approaches for improvement. Chapter 6 describes the design of GAMPALv2. Chapter 7 describes the evaluation results of GAMPALv2. Chapter 8 discusses this dissertation in depth. Chapter 9 concludes this dissertation.

Chapter 2

Related Work

2.1 Preliminary Technology

2.1.1 BGP and RIB

BGP (Border Gateway Protocol) is a routing protocol in TCP/IP (Transmission Control Protocol/Internet Protocol) networks that controls the exchange of routing information between ASs (Autonomous Systems). An AS is a set of networks managed under a unified operational policy, identified by an AS number on the Internet. A TCP/IP network consists of interconnected ASs, as shown in Fig. 2.1. Connections between ASs are eBGP (external BGP) sessions, and connections within an AS are iBGP (internal BGP) sessions. An AS is connected to one or more other ASs by eBGP sessions, and routers in an AS that are connected to a neighboring AS are called edge routers of that AS.

When a router in an AS receives information on a new prefix, the

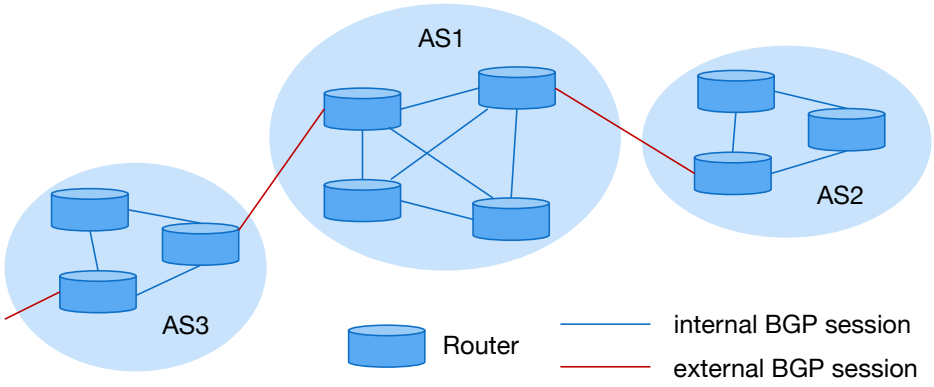


Figure 2.1: ASs and BGP sessions.

router registers the prefix in the RIB (Routing Information Base). Figure 2.2 shows an example of information in a RIB. The RIB contains an IP address prefix, the AS path, the path of traffic destined to the prefix, and other information. The AS path is the best path to reach the prefix and is determined by the algorithm in BGP. In the example of Fig. 2.2, the AS path ("ASPATH" in the figure) for the traffic destined for 1.0.0.0/24 is 4713, 2914, 13335. This example indicates that AS4713, AS2914, and AS13335 are passed through to reach this prefix.

An AS sends its own AS routing information to its neighbors by eBGP. When an AS sends routing information to its neighbors, it adds its own AS number to the AS path of each prefix. In this way, the routing information received by the neighboring AS is updated to the routing information from the viewpoint of the neighboring AS. In the AS that receives the routing information, the information is distributed by iBGP. The AS that receives the routing information sends the routing information to its neighboring ASs so

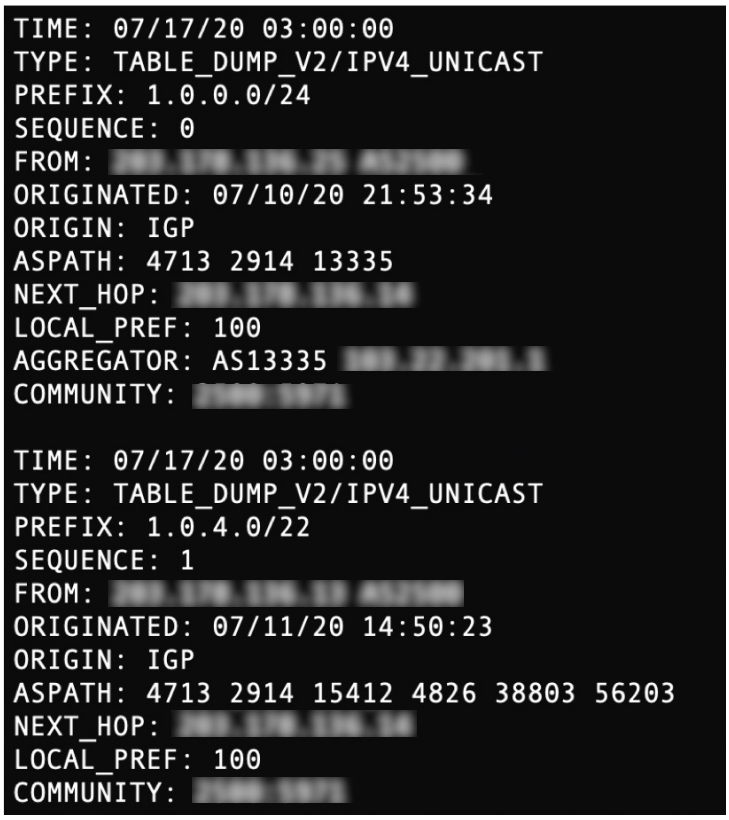
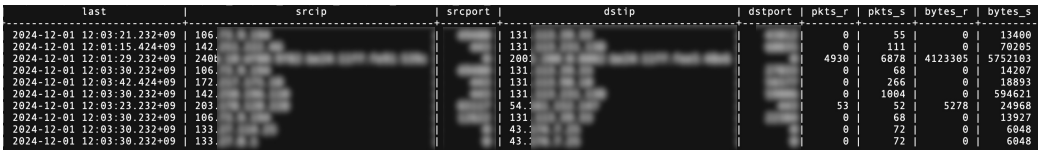


Figure 2.2: Example of the information in RIB.

that the information can be transmitted to non-neighboring ASs. Therefore, in the example in Fig. 2.2, the information on the prefix reachable from AS3 is distributed to AS2 via AS1. By repeating this process, routing information is distributed throughout the ASs that make up the Internet, and the routers in each AS can maintain the optimal routing information to reach each prefix. The number of ASs passed through to reach a destination is the number of hops. In the example of Fig. 2.2, the number of hops for 1.0.0.0/24 is three. The number of hops varies by source and destination IP addresses. If the source and destination IP addresses are far apart on the Internet, the number



last	srcip	srcport	dstip	dstport	pkts_r	pkts_s	bytes_r	bytes_s
2024-12-01 12:03:21.232+09	106.		131.		0	55	0	13400
2024-12-01 12:01:15.424+09	142.		131.		0	111	0	70205
2024-12-01 12:01:29.232+09	240		200		4930	6878	4123305	5752103
2024-12-01 12:03:30.232+09	106.		131.		0	68	0	14207
2024-12-01 12:03:42.424+09	172.		131.		0	266	0	18893
2024-12-01 12:03:30.232+09	142.		131.		0	1004	0	594621
2024-12-01 12:03:23.232+09	203.		54.3		53	52	5278	24968
2024-12-01 12:03:30.232+09	106.		131.		0	68	0	13927
2024-12-01 12:03:30.232+09	133.		43.3		0	72	0	6048
2024-12-01 12:03:30.232+09	133.		43.3		0	72	0	6048

Figure 2.3: Example of the flow information.

of hops will be higher. If the source and destination IP addresses are in a single AS, the number of hops is 0.

2.1.2 Internet Traffic Information

Data flowing over the Internet is observed and collected as packets or flows. When communicating over the Internet, data is divided into packets by IP and sent in packet units. A packet consists of an IP header and a payload. Packets move between ASs based on the destination IP address information in the IP header. A flow is a packet sequence with a common five-tuple (source IP address, destination IP address, source port, destination port, and protocol number).

Figure 2.3 shows an example of raw flow information. Flow information contains five-tuples, observed time, and data size. Flow information is generated from packets flowing through observation points using collection tools. The collected information is monitored, analyzed, and used for various purposes such as operation, management, and security.

2.2 Existing Research

As described in Sect. 1.1, anomaly detection mechanisms for Internet traffic are classified into two approaches: signature-based and anomaly-based methods. Signature-based methods [2–9] use a pre-defined list of signatures and are suitable for detecting known anomalies. On the other hand, anomaly-based methods [10–20] are suitable for detecting unknown anomalies. Since GAMPALv1 and GAMPALv2 belong to the anomaly-based method, this section describes the related works on the anomaly-based methods in detail.

2.2.1 Enterprise/DC-scale

Existing anomaly-based detection methods are classified into two types depending on the scale of the network: enterprise/DC (Data Center)-scale and Internet-scale. [10–12, 15–20] are for DC/Enterprise scale, and [13, 14] are for Internet scale.

For an enterprise/DC-scale network, [10] proposes a range-based anomaly detection approach. The proposed model detects both performance and security anomalies in cloud systems. This mechanism consists of a time series reconstruction model and anomaly detection modules. This research adopts LSTM (Long Short-Term Memory) encoder-decoder [29] in the reconstruction model and proposes three types of detection modules. In the performance evaluation, four different types of datasets are used. The first dataset is obtained from an experimental testbed. Besides this, Yahoo! Webscope

S5 dataset [30], Numenta Anomaly Detection Benchmark Data Corpus [31], and the CIC-DDoS2019 dataset [24] are used.

[11] proposes an intrusion detection system for SDN (Software-Defined Networking) architecture. The proposed system classifies normal and anomalous flows. This system comprises a fully connected DNN (Deep Neural Network) and GRU RNN (Gated Recurrent Unit Recurrent Neural Network) [32]. In addition, this system is designed efficiently, considering its practicality, and evaluated for its performance in terms of throughput, latency, and resource utilization. The classification model is trained on the NSL-KDD [23], a dataset for intrusion detection systems.

[12] proposes a general-purpose anomaly detection mechanism for an enterprise network. The proposed architecture consists of a clustering layer, an image generation layer, and an NN (Neural Network) analysis layer. The detection method is derived from a CNN (Convolutional Neural Network)-based classification of traffic information visualizations. The traffic information is categorized using the MCODT (Micro-Cluster Outlier Detection in Time series) clustering algorithm [33] and visualized using the SOM (Self-Organization Map) dimensionality reduction algorithm [34].

[15] proposes an approach to detect cyber attacks for a large-scale enterprise network. This research proposes clustering-based anomaly detection. The proposed approach aggregates flow information by the source IP address, and then the aggregated flows are clustered using the k-means algorithm. Finally, the flows with a large Euclidean distance from the center

of the cluster are detected as anomalies. In the case study, the CTU-13 dataset [35] is used, and the results are visualized in three dimensions by dimension reduction using PCA (Principal Component Analysis).

[16] proposes an anomaly detection methodology for SDN-based networks. This research uses flow statistics considering the application to SCADA systems. SCADA systems are used in critical infrastructures such as the electric power grid or ICS (Industrial Control Systems). The proposed architecture comprises an NN-based classifier and a DAE (Deep Autoencoder). Since labeled data for SCADA networks are unavailable, the CIC-IDS2017 [36] is used in the evaluation.

2.2.2 Internet-scale

For Internet-scale networks, [13] proposes a hybrid machine-learning model for detecting botnet attacks in network traffic. A botnet is a network of computers under the control of an attacker through malware infection. Since the botnet is used for DDoS (Distributed Denial of Service) attacks and to steal sensitive data, it is known as a major threat model on the Internet. This research focuses on anomalous DNS (Domain Name System) traffic generated during the botnet life cycle and proposes anomaly detection by DNS packet analysis. The proposed model combines the k-means algorithm, rule-based systems, and decision tree algorithms. In addition, the CTU-13, a labeled dataset with botnet traffic, is used in the evaluation.

Table 2.1: Comparison of GAMPAL and related work.

	Enterprise/DC-scale			Internet-scale		
Related work	[10]	[11]	[12]	[13]	[14]	GAMPALv2
Scalability	No	No	No	Yes	No	Yes
Locality	No	No	No	No	No	Yes
Versatility	Yes	No	Yes	No	No	Yes
Labelled data free	No	No	Yes	No	No	Yes

[14] proposes a traffic classification framework for network anomaly detection. This framework is a hybrid classification method based on NIN (Network in Network) [37] and depthwise separable convolution [38]. This method designs models so that its computational complexity reduces for practical use. [14] also uses the CICIDS2017 dataset, a labeled dataset containing multiple types of malicious attacks [36]. The proposed method classified the traffic of cyber attacks such as DDoS and port scan attacks with higher accuracy than LSTM and CNN.

Table 2.1 compares GAMPAL and the existing methods [10–14]. As described in Sect. 1.2, this dissertation defines four requirements: (i) scalability to the Internet, (ii) consideration for the locality of Internet traffic, (iii) versatility for any kind of anomaly, and (iv) not using labeled data.

Regarding scalability, [10] proposes anomaly detection for cloud systems and has no function considering scalability for BGP (Border Gateway Protocol) full routes. The system in [11] is designed to facilitate small and large-scale networks, but has no function considering scalability for BGP

full routes. The SOM used in [12] does not have an aggregation mechanism for flows because it focuses only on enterprise networks, not Internet-scale networks, and does not consider scaling. [13] proposes a botnet detection method for Internet-scale networks and uses the k-means algorithm to group Internet traffic into clusters. Although [14] does not limit the scale of the network to which it is applied, it has no function considering scalability for BGP full routes. In addition, the evaluation uses datasets generated from a simulated small-scale network.

Regarding the locality of Internet traffic, the methods in [10–14] do not consider the locality. Most open data are anonymized by modifying or deleting attributes such as IP addresses to prevent misuse and protect privacy. Therefore, anomaly detection methods that focus on the locality of Internet traffic are rare. GAMPAL proposes a unique anomaly detection mechanism by considering the locality of Internet traffic.

Regarding versatility, [10] focuses on performance and security anomalies in cloud systems. However, anomaly detection methods based on reconstruction models do not limit the detection target to a specific anomaly. In addition, [12] proposes a general-purpose anomaly detection mechanism for an enterprise network. Although [10] and [12] seem to have versatility for anomalies, they do not have enough validation to detect various anomalies. On the other hand, the approaches in [11] [13] [14] need to be sufficiently versatile for handling various anomaly types. [11] points to the original structural vulnerabilities of SDN as an issue and proposes an intrusion detection sys-

tem for SDN architectures. [13] specializes in botnet detection. [14] detects anomalies from malicious attacks in its evaluation.

Regarding the labeled data, some existing mechanisms use labeled data. [10] proposes a semi-supervised dynamic density-based anomaly detection method and uses Yahoo! Webscope S5 dataset [30], Numenta Anomaly Detection Benchmark Data Corpus [31], and the CIC-DDoS2019 dataset [24]. [11] uses the NSL-KDD [23] dataset. [13] uses DNS packets extracted from the CTU-13 dataset [35]. [14] uses the CICIDS2017 dataset [36]. [12] does not use labeled data. The detection validity is evaluated by comparing the moment the proposed method detects behavior changes and when an event occurs in the real world.

This dissertation proposes GAMPAL, an anomaly detection mechanism for the Internet backbone networks. While various anomaly detection methods have been proposed for Internet traffic, GAMPAL satisfies four requirements that consider the characteristics of the Internet backbone network, which is an advantage of GAMPAL. GAMPAL considers scalability and locality using a scalable aggregation method based on the routing table. In addition, GAMPAL aims to detect anomalies that may disrupt Internet operations, such as failures in primary services and large-scale cyber attacks. GAMPAL establishes a model to predict the volume of Internet traffic and detects anomalies by comparing predicted and observed values. Therefore, the detection target is not limited to a specific anomaly, nor does it use labeled data. The details of GAMPAL architecture are described in Sect. 3.

Chapter 3

GAMPAL Architecture

3.1 Overview of GAMPAL Architecture

Figure 3.1 shows the overview of the GAMPAL architecture. GAMPAL uses flow information and the routing table. A flow is a packet sequence with a common five-tuple (source IP address, destination IP address, source port, destination port, and protocol number). First, GAMPAL applies *prefix aggregation* to flows by referring to the routing table. Prefix aggregation is an aggregation method based on the BGP RIB (Border Gateway Protocol Routing Information Base), which aggregates flow spatially. In addition, this dissertation introduces *PA (Prefix Aggregate)*, a unit of aggregated flows based on the prefix aggregation. This aggregation method addresses the consideration of scalability and locality in the four requirements described in Sect. 2.2. Next, GAMPAL applies temporal aggregation with a five-minute time slot to flow information for each PA. Consequently, time series data

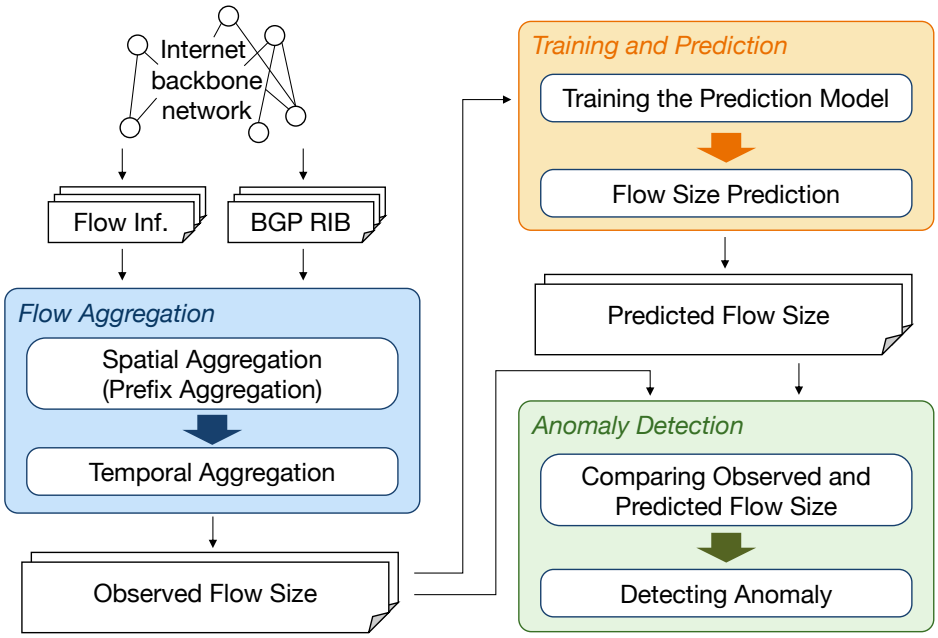


Figure 3.1: Overview of the GAMPAL archtecture.

of flow sizes for each PA are generated. The flow size is the total number of bytes of flows observed in a time slot. The more packets are observed in the time slot, the larger the flow size will be. Next, a model for each PA that predicts the flow size is established using the time series data of the flow sizes. Because the models are not classification models, GAMPAL does not use labeled data. GAMPAL compares the observed flow sizes with the predicted flow sizes and detects an anomaly if the observed flow size significantly differs from the predicted flow size. GAMPAL detects various types of anomalies by evaluating the influence of the anomaly on the flow information.

As described in Sect. 1.1, GAMPALv1 and GAMPALv2 have been

proposed. GAMPALv1 and GAMPALv2 differ in the machine learning method for the prediction model, the training approach, and the anomaly detection method. This chapter describes common features in GAMPALv1 and GAMPALv2.

3.2 Prefix Aggregation

Anomaly detection for Internet-scale networks requires a scalable analysis of network traffic, as described in Sect. 2.2. GAMPAL adopts flow information as Internet traffic. A flow is a set of packets with common attributes. Therefore, flows have fewer records per unit time and can be handled scalably. A flow is identified with a five-tuple: source and destination IP addresses, source and destination ports, and protocol.

The traffic pattern is peaky and jaggy and changes rapidly even during ordinary times. Therefore, it is necessary to aggregate flows spatially and analyze the data by its aggregation units. The order of the number of IP addresses is $O(10^9)$ in the case of IPv4. Focusing on the source or destination addresses in the five-tuple, the order of the number of flows is $O(10^9)$. Next, the addresses are grouped into the IP address prefixes in the BGP RIB. As of April 2024, the number of IPv4 BGP full routes, i.e., the number of destination address prefixes, is more than 1,100,000, i.e., $O(10^6)$, which is still too large a scale to analyze Internet traffic. In GAMPALv1, the destination address prefixes are grouped into the PAs, in each of which the first k AS

(Autonomous System) numbers in the AS path attribute are the same. To define the k of the prefix aggregation, the distribution of AS path lengths of IPv4 full routes observed in AS2500 is analyzed. Figure 3.2 shows the distribution of the AS path lengths of the IPv4 BGP full routes observed in AS2500 on July 17, 2020. Although there are few flows with the AS path length less than three, there are many flows with the AS path length of three. This result indicates that there are many flows that differ from the 3rd hop of the AS path. Therefore, the prefix aggregation with k less than 3 is insufficient to aggregate the flows. Based on this analysis, k of the prefix aggregation is defined as three, the mode value of the AS path length. If three is adopted as k, the order of the number of PAs is $O(10^4)$.

An example of the prefix aggregation is shown using Figs. 3.3 and 3.4. First, as shown in Fig. 3.3, the prefixes in the BGP RIB are aggregated by prefix aggregation. Referring to the AS path attribute in the BGP RIB, the prefixes that have the same first three AS numbers in the AS path attribute are aggregated. An identifier of a PA combines three AS numbers. Note that if the AS path length is less than three hops, the identifier combines less than three AS numbers.

Next, prefix aggregation is applied to the flow information as shown in Fig. 3.4. By matching the destination IP address of the flow information with the prefix, the PA to which each flow belongs is identified. Consequently, the flow information is divided into PAs.

At an observation point, a large number of destination addresses close

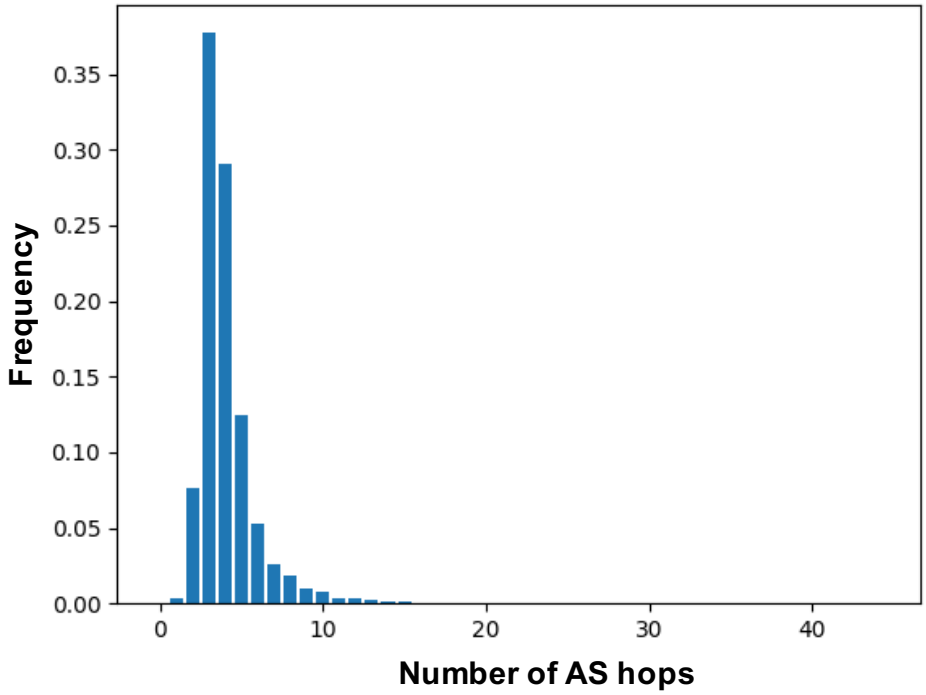


Figure 3.2: Histogram of AS path length.

to the IP address of the observation point are observed. In contrast, 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 prefix aggregation, the destination addresses close to the observation point can be grouped into fine-grained units. In contrast, the destination addresses distant from the observation point can be grouped into coarse-grained units. Therefore, prefix aggregation is a suitable aggregation method for the locality of Internet traffic. As described above, GAMPAL satisfies scalability by applying spatial aggregation over IP address space.

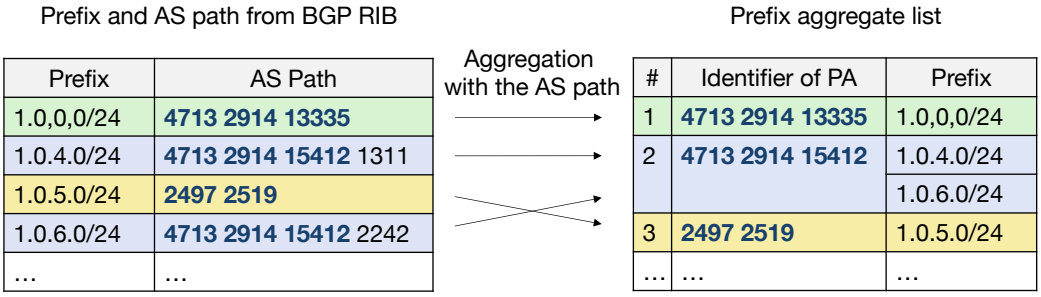


Figure 3.3: Example of prefix aggregation.

3.3 Flow Size Vector

GAMPAL uses the flow sizes as time series data. GAMPALv1 establishes a prediction model for flow sizes based on past flow sizes and uses an LSTM-RNN (Long Short-Term Memory Recurrent Neural Network) model. Before the training, to process flow information into time series data of flow size, GAMPALv1 introduces a *flow size aggregation interval* (e.g., five minutes) and aggregates flow sizes temporally. Figure 3.5 shows an example of applying temporal aggregation to flow information. During a flow size aggregation interval (e.g., from 00:00 to 00:05), the payload sizes of the observed packets are summed up. Temporal aggregation is applied to the flow information for each PA. As a result of temporal aggregation, time series data of flow sizes are generated, named *the flow size vector*.

If the flow size aggregation interval is longer than the duration of the anomalies, the effect on flow size is rounded off and may cause FNs (False Negatives). For example, the average duration of a DDoS (Distributed Denial of Service) attack is reported to be a little over 20 minutes [39], and most

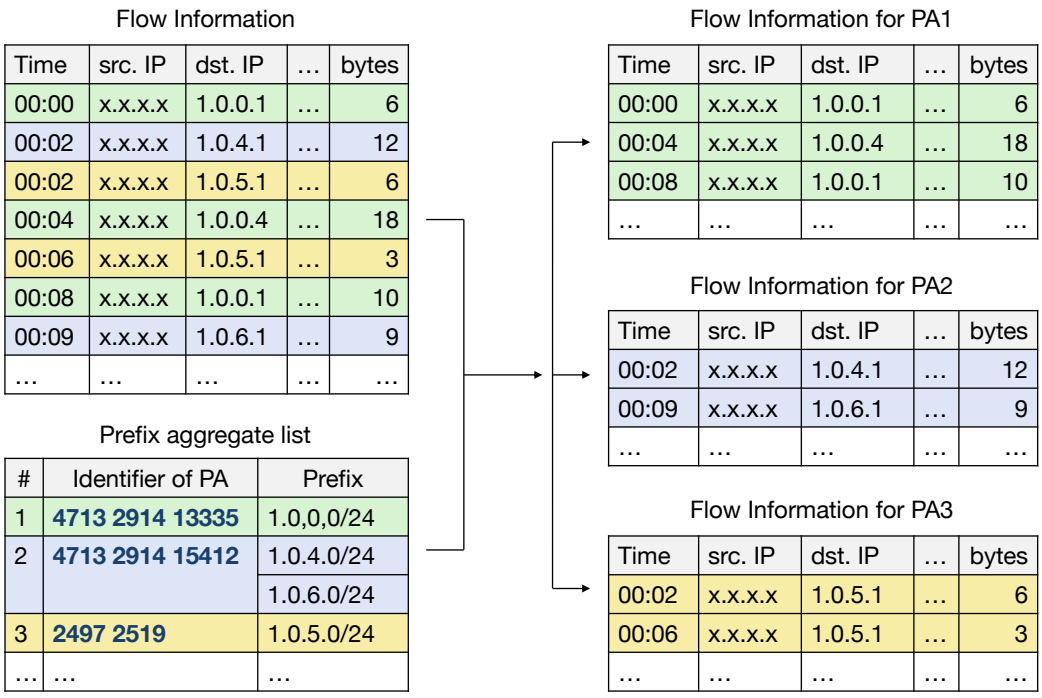


Figure 3.4: Applying prefix aggregation to flow information.

failures affecting the Internet are not resolved within a few minutes. Therefore, GAMPALv1 defines the flow size aggregation interval as five minutes, and the number of flow sizes in a flow size vector for one day is 288.

The matrix consisting of the observed flow size vectors of all PAs is named *the observed flow size matrix*. Using the observed flow size matrix, GAMPAL establishes the prediction model for each PA.

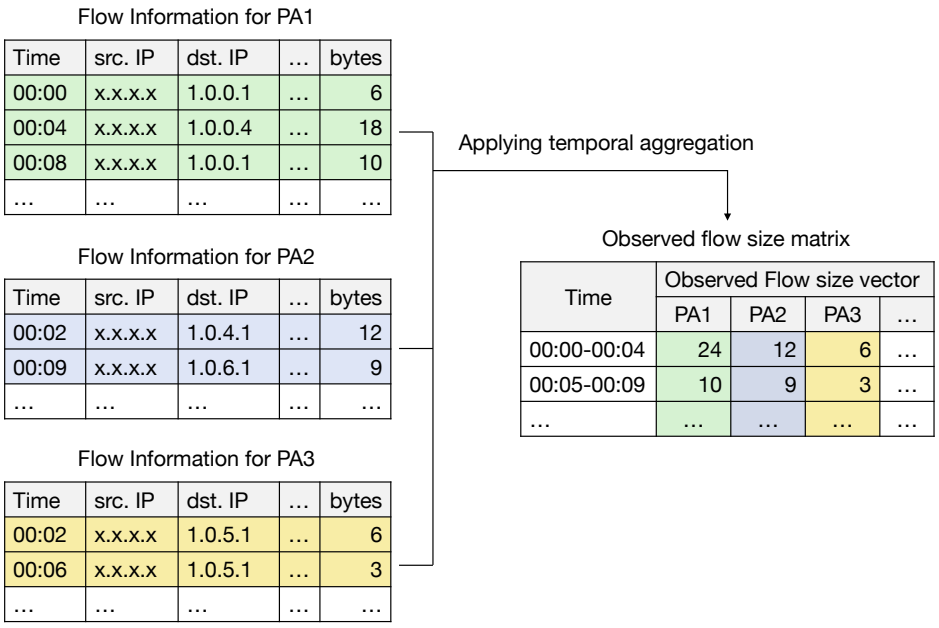


Figure 3.5: Generating flow size vector.

3.4 Anomaly Detection with Flow Size Prediction

Most existing research uses labeled data to train models on the characteristics of packets or flows with anomaly labels. The trained model detects an anomaly by evaluating the presence or possibility of an anomaly based on the characteristics of the packets or flows. GAMPAL detects anomalies by evaluating the influence of the anomaly on the flow information. The prediction model in GAMPAL predicts time series data of Internet traffic flow sizes. The model is trained on time series data of past flow sizes in which no anomalies were reported and predicts the behavior of the flow size without an anomaly. Therefore, if the observed flow size is significantly different from

the predicted flow size at the same time predicted by the model, there may have been an anomaly that affected the behavior of the flow size. In this way, GAMPAL can detect various anomalies by observing their effects on flow size. Since GAMPAL does not create a classification model, the labeled data is unnecessary.

Chapter 4

GAMPALv1

4.1 Design of GAMPALv1

Figure 4.1 shows the overview of GAMPALv1 [21]. GAMPAL applies prefix aggregation to the flow information using the routing table and then applies temporal aggregation (Fig. 4.1-(1),(2)). Consequently, time series data of flow sizes aggregated by PAs (Prefix Aggregates) are generated (Fig. 4.1-(3)). GAMPALv1 predicts the flow size based on past flow sizes using LSTM-RNN (Long Short-Term Memory Recurrent Neural Network) (Fig. 4.1-(4),(5)). It detects an anomaly by comparing the predicted flow sizes with the observed ones. GAMPAL defines the *NSD (Normalized Summation of Difference)*, which evaluates the ratio of the difference between two time series data (Fig. 4.1-(6)). In the evaluation, the NSD is calculated for each PA based on the predicted and observed flow sizes for one day.

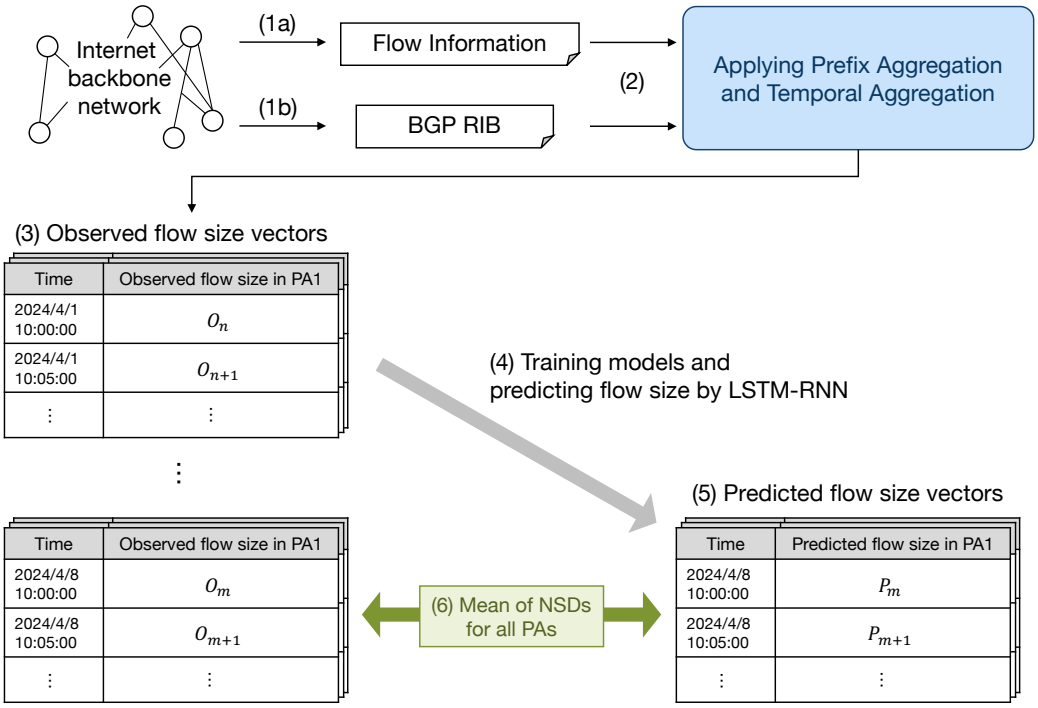


Figure 4.1: Overview of the GAMPALv1 methodology.

4.1.1 Prediction Methodology in GAMPALv1

GAMPALv1 establishes a prediction model for flow size based on past flow sizes and uses an LSTM-RNN model. As shown in Fig. 4.2, the flow size vector is the explanatory variable for the model, and the subsequent flow size is the objective variable. Therefore, the prediction model of GAMPALv1 takes a flow size vector consisting of 288 flow sizes and predicts the value of the subsequent flow size. For example, the values of the 0th to 287th flow sizes are inputted into the LSTM-RNN to generate a predicted value for the 288th flow size. This value is compared to the observed value of the 288th flow size. By sliding these variables by a single time slot along the

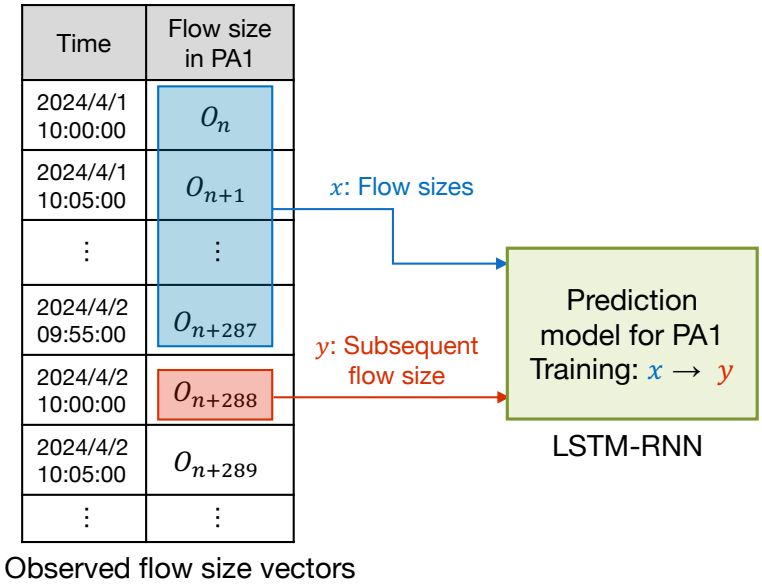


Figure 4.2: Leaning approach of GAMPALv1.

time axis over the range of the training data, the prediction model learns the behavior of the time series data of the flow size by adjusting its internal parameters so that the outputs for the explanatory variables approach the objective variables. This model learns the periodicity of Internet traffic from the flow size sequence. Internet traffic tends to exhibit a daily periodicity, with a large flow size during the daytime when there are many active users and services, and a small flow size at night. To learn the daily periodicity of Internet traffic, GAMPALv1 designed a model that takes a time series data of flow size for one day as input.

4.1.2 Detection Method in GAMPALv1

GAMPAL detects an anomaly in cases where the observed flow size differs significantly from the predicted flow size by the prediction model. The output of the model in GAMPALv1 is a scalar value in a single time slot for each PA, which means that the predicted value is one for each time slot and each PA. Therefore, GAMPALv1 calculates an indicator value that measures the difference between observed and predicted values over a certain time range and evaluates anomaly detection performance based on that indicator.

For Internet traffic, the scale of the flow size varies greatly depending on the observation environment and time. In some cases, only zero to a few bytes are observed, while in others, several hundred million bytes are observed. The scale of the flow size varies significantly for each PA, even at the same observation point at the same period. Therefore, in GAMPALv1, an indicator that can evaluate a wide range of traffic volumes on the same scale is necessary. Therefore, indicators with different scales that depend on the underlying data, such as the MSE (Mean Square Error), are unsuitable. In addition, for some PAs, the observed and predicted values may be zero. Zero means that no packets are observed during a period, and such PAs are often observed. In such cases, indicators such as the RMSPE (Root Mean Square Percentage Error) cannot be calculated because the numerator is larger than zero and the denominator is zero.

Thus, GAMPALv1 defines the NSD. 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.

$$NSD = \frac{\sum_{i=1}^{T} |m_i - p_i|}{\sum_{i=1}^{T} \max(m_i, p_i)}. \tag{4.1}$$

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 and can be calculated even if any observed or predicted value is zero. The NSD shows the difference between the two values and also indicates the validity of the prediction.

GAMPALv1 uses a prediction model to predict flow sizes on a day when an anomaly occurred and on a day when no anomaly occurred (Fig. 4.1-(5)). It then calculates the NSD using the predicted and observed flow sizes for one day. GAMPALv1 uses the mean of the NSDs for all PAs to evaluate the performance of GAMPALv1 (Fig. 4.1-(6)).

4.2 Implementation of GAMPALv1

GAMPALv1 is implemented in Python 3.7.0 on a server running Ubuntu Server 18.04.1. The LSTM-RNN model for traffic prediction is implemented with `Chainer` 5.1.0 [40], an open-source deep learning framework, and `NstepLSTM` class, a class for supporting LSTM-based learning in Chainer. `nfdump` version 1.6.17 [41] is used to convert the packet information. `bgpdump` version

1.4.99.13 [42] is used to convert the BGP RIB (Border Gateway Protocol Routing Information Base). A GPU is used for the calculations of the LSTM-RNN. The GPU platform is CUDA 9.0. The implementation is optimized to use `cuDNN` [43] library for a GPU computing environment.

As described in Sect. 4.1.1, the explanatory variable for the model is the flow size vector consisting of 288 flow sizes, and the objective variable is the subsequent flow size. In other words, the input to the model is 1×288 variables, and the output of the model is a scalar value. There is a single LSTM layer in the model. The more LSTM layers, the larger the model size and the more computational complexity. In GAMPALv1, the number of LSTM layers was determined considering the trade-off with computational complexity. The computational complexity is discussed later in Sect. 5.1.1.

4.3 Evaluation of GAMPALv1

4.3.1 Dataset

In evaluating GAMPALv1, the flow information and the BGP RIB are exported from the WIDE backbone network (AS2500) [44]. The WIDE backbone network is a nationwide Layer-2 and Layer-3 network and includes core and branch NOCs (Network Operations Centers), some of which provide connectivity to stub organizations such as universities. The flow information is raw data to which sampling is not applied. While GAMPALv1 can also work with sampling data, a large sampling rate reduces the granularity of the data

and may cause FNs (False Negatives). The flows are observed at a branch NOC accommodated in a university. The BGP RIB is observed at a route server in the WIDE backbone network.

GAMPALv1 evaluates the validation of detection for three types of anomalies: connection failures, event traffic, and cyber attacks. In the case of connection failures, the flow sizes decrease because the traffic observed during ordinary times is not generated. However, the flow size increases for the other two types of anomalies. In the case of event traffic, the flow sizes increase because the event causes traffic that is not observed during ordinary times. Event traffic is a more common anomaly than the other two types of anomalies. Since cyber attacks are major threats in the Internet backbone networks, it is necessary to evaluate the validity of cyber attack detection. In the case of cyber attacks, the flow sizes increase because the traffic for attacking is generated that is not observed during ordinary times. GAMPALv1 uses actual cases corresponding to the above three types of anomalies observed in the WIDE backbone network in the evaluation: (i) a connection failure on YouTube on October 17, 2018, (ii) a campus festival on November 22-25, 2018, and (iii) a DDoS attack between June and July 2019.

The connection failure on YouTube occurred on October 17, 2018 [45]. It is reported that the failure began around 10:00 and gradually recovered between 12:00 and 13:00 [46]. During this period, large video content flows from YouTube to end-users should not have been observed. Therefore, the

flow sizes in the PA containing the flows destined for the university network, which is the observation point, should be smaller than that of ordinary times.

The campus festival was held on November 22-25, 2018, on a university campus that is part of the WIDE backbone network. The traffic behavior observed with an event is generally called event traffic. During this period, the university canceled all classes, and many students and visitors gathered on the campus. In addition, access to the university's website from outside and the number of users of the wireless LAN (Local Area Network) in the university are expected to increase. Therefore, network traffic involving the university network, which is the observation point, may behave differently during this period.

The DDoS attack is an incident that exploited a university network that is part of the WIDE backbone network between June and July 2019 [47]. It is a UDP (User Datagram Protocol) reflection/amplification attack using ARMS (Apple Remote Management Service) [48] port (3283/udp). On July 9, the university noticed the attack and blocked the ARMS traffic [47]. Therefore, the data from July 8, just before the university had to deal with the incident, is used as the date of the DDoS attack.

4.3.2 Validation of Anomaly Detection

Figure 4.3 shows the NSDs for the normal day and the days of the connection failure, the campus festival, and the DDoS attack. The NSD for the normal

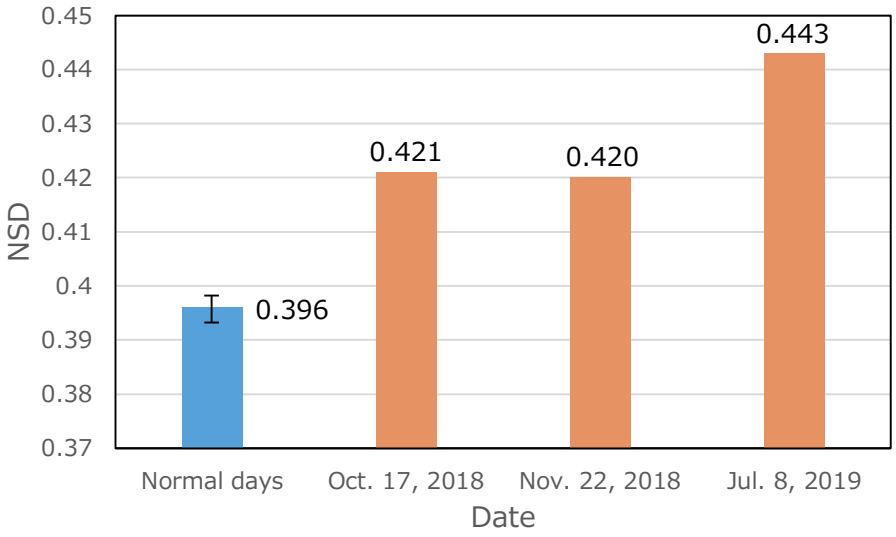


Figure 4.3: Evaluation result of GAMPALv1.

days is the mean value and standard deviation of the NSDs calculated for the days on which no anomalies were reported. The dates on which no anomalies were reported are selected from those that are close to the date on each anomaly. The mean and standard deviation of the NSDs for the normal day are 0.396 and 0.0023, respectively. Note that the NSDs for the normal days are all less than 0.4, and the maximum value is 0.399. On the other hand, the NSDs for all the days with anomalies exceed 0.42. These results indicate that the difference between the predicted and observed flow size is large when an anomaly occurs. The NSD for the days with the DDoS attacks is particularly large, indicating the influence of the attacks on the network traffic. As described above, it is confirmed that GAMPALv1 can detect anomalies based on the difference between the predicted and observed flow sizes on the days with the anomalies.

4.4 Contributions in GAMPALv1

GAMPALv1 shows that the ratio of the difference between the predicted and observed flow size vectors increases on the days when an anomaly occurs. This result shows that GAMPALv1 can detect various anomalies such as connection failures, event traffic, and cyber attacks.

GAMPAL is the anomaly detection mechanism for the Internet backbone network and satisfies the requirements considering the characteristics of the Internet backbone networks, which is the advantage of this research. In existing research, signature-based anomaly detection with the predefined rules and anomaly detection based on classification models that learn the characteristics of anomalies have been proposed. On the other hand, GAMPAL focuses on the byte count of observed flows and detects anomalies based on the influence of the anomaly on the flow sizes. GAMPAL is different from existing anomaly detection in these respects.

Thus, GAMPAL may fit into the defense-in-depth strategy, an information security concept for threat prevention. The defense-in-depth is a strategy that uses multiple security measures to ensure protection against threats. In the case of anomaly detection, anomalies that cannot be detected by one detection method are detected by another to prevent FNs. Since GAMPAL has different characteristics from other research, it is expected to be used with signature-based methods or anomaly detection methods based on classification models in the defense-in-depth strategy.

Chapter 5

Issues of GAMPALv1

5.1 Issues of Prediction Model

5.1.1 Enormous Computation Complexity

GAMPALv1 requires significant computational time, particularly for training the prediction model and outputting predicted values. GAMPALv1 takes about four days to train a model and to output predicted values on all the PAs for one day using a GPU. This fact is a severe issue for the practical use of GAMPAL. The objective of anomaly detection is to ensure the stable operation of the network. However, excessive computational complexity may increase the load on the machines that comprise the network. Therefore, the computation for anomaly detection should be performed quickly, even under standard computing environments such as a CPU.

The reason for the enormous computational complexity is that the model is based on the LSTM-RNN (Long Short-Term Memory Recurrent

Neural Network). GAMPALv1 established the prediction model based on the LSTM-RNN for learning the periodicity of network traffic. LSTM-RNN is known to have an enormous computational complexity, which makes it unsuitable for systems with time constraints. In addition, the number of prediction models is the same as the number of PAs (Prefix Aggregates) ($O(10^4)$).

The other reason is that the input data has a large number of dimensions. The computational complexity of the model depends on this number, both during training and prediction. As described in Sect. 4.1.1, GAMPALv1 established a model in which the flow size vector is the explanatory variable and the subsequent flow size is the objective variable. Therefore, the prediction model for each PA takes a 288-dimensional input and outputs a scalar value. In the context of neural networks, the size of the model and the computational complexity are proportional. Although the prediction model implemented in GAMPALv1 is relatively simple, comprising a single LSTM layer, it requires four days to process. Consequently, the issue of computational complexity cannot be resolved by merely redesigning the prediction models based on LSTM-RNNs.

5.1.2 Using Semantics related to Time

The prediction model in GAMPALv1 not only has enormous computational complexity but is also unsuitable for predicting flow sizes. The model learns the characteristics of flow size behavior from the flow size sequence. However,

flow size is complexly affected by the hourly or daily period and features such as the day of the week, whether it is a business day, and so on. In GAMPALv1, semantics related to time, such as date, time, and day of the week, are not used in the input data to the model. To predict flow size accurately, it is necessary to learn the relationship between the semantics related to time and flow size.

5.2 Issues of Detection Method

5.2.1 Validity of Prediction

The output of the model in GAMPALv1 is a scalar value for each PA, which means that the predicted values are unique for each time slot and each PA. To evaluate the ratio of the difference between the predicted and observed flow size vector, GAMPALv1 defined NSD (Normalized Summation of Difference). However, the NSD varies significantly for each condition, such as PA, period, and season in GAMPALv1. This problem makes it difficult to define criteria for anomaly detection in GAMPALv1. The cause of this problem is the validity of the predictions.

In cases in which the validity of the prediction is low, the range of possible values for the flow size is wide. Since the output of the prediction model in GAMPALv1 is a scalar value, the NSD tends to be large in such a case, even if an anomaly has not occurred. On the other hand, in cases in which the prediction is highly valid, the range of possible values for the flow

size is narrow. In such a case, the NSD tends to be small.

This detection method causes FNs (False Negatives) and FPs (False Positives). In GAMPALv1, the detection accuracy is evaluated using labeled data [49], and the mean recall is 81.8% [21]. Here, the recall for the detection evaluation is shown in Equation 5.1 below, using the numbers of TPs (True Positives) and FNs.

$$Recall = \frac{TP}{TP + FN}. \tag{5.1}$$

These results are lower than those of related research. Note that the dataset used for the accuracy evaluation in GAMPALv1 is partially anonymized. In addition, since the periods without anomalous flows are extremely short, the training data for the model is insufficient. From these points of view, the dataset used in GAMPALv1 is unsuitable for accuracy evaluation. Therefore, GAMPALv2 evaluates detection accuracy using a new dataset.

5.2.2 Identification of PA and Period with Anomalies

The method method in GAMPALv1 cannot identify the PA and period in which the anomaly occurred. This detection method makes it difficult to define standard criteria of the NSD for anomaly detection for all PAs. For this reason, GAMPALv1 calculates the mean of the NSDs for all PAs on the day for evaluation and the days without an anomaly. Since the mean of the

NSD is larger on the day with the anomaly, GAMPALv1 concluded that it can detect days with anomalies. Identifying the PA and period in which the anomaly occurred is necessary for the practical use of GAMPAL.

5.3 Approach to GAMPALv2

To solve the issues of GAMPALv1 and improve its practicality, GAMPALv2 defines four requirements. For the issue described in Sect. 5.1, GAMPALv2 requires (a) reducing the computational complexity in training and prediction and (b) using the semantics related to time, such as date, time, and day of the week, to train the model. Reducing the number of dimensions of the inputs to the model reduces the computational complexity required for training and prediction, thus reducing processing time. In addition, by using the semantics related to time for prediction, complex periodicity and characteristics of network traffic can be learned.

For the issue described in Sect. 5.2, GAMPALv2 requires (c) consideration of the range of possible values for the flow size and (d) detection of when and in which PAs an anomaly occurred. Considering the range of predicted values, the issue of the difference in the validity of the prediction for each condition can be solved. In addition, anomaly detection for each PA can provide the insight necessary for the response after the detection.

Chapter 6

GAMPALv2

6.1 Overview of GAMPALv2

Figure 6.1 shows the overview of the GAMPALv2 methodology. For scalability and locality, GAMPALv2 uses the prefix aggregation introduced in GAMPALv1 (Figs. 6.1-(1)∼(3)). GAMPALv2 establishes a model that predicts the flow size in a time slot from the time feature which is extracted from the time attributes of that time slot (Fig. 6.1-(4)). The time feature is a variable consisting of seven items: month, day, hour, minute, day of the week, week of the month, and whether it is a business day. Further details of each item are described later in Sect. 6.4. GAMPALv2 predicts flow size by focusing on the relationship between the time feature and flow size. In addition, GAMPALv2 adopts RFR (Random Forest Regressor) [50] as the prediction model. Random Forest is an ensemble learning model consisting of multiple decision trees. It is known that RFR is suitable for modeling the influences

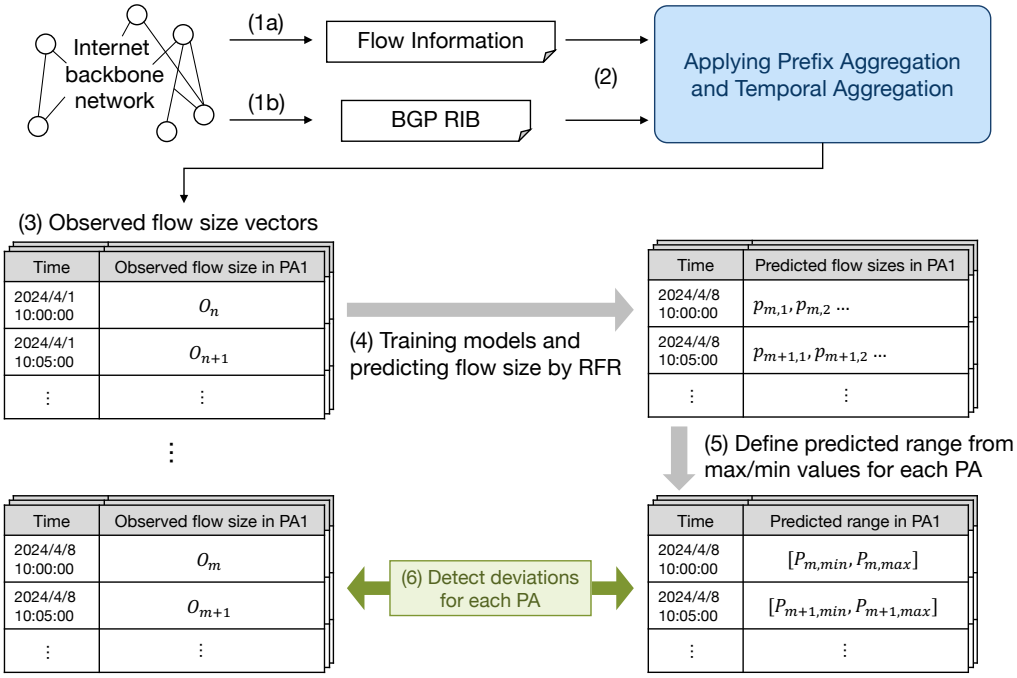


Figure 6.1: Overview of the GAMPALv2 methodology.

of multiple features in explanatory variables on the objective variable. RFR contributes to reducing the processing time of GAMPAL since its computational complexity is smaller than that of LSTM-RNN (Long Short-Term Memory Recurrent Neural Network).

In the detection method, GAMPALv2 calculates a predicted range of possible values for the flow size, and detects anomalies for each PA (Prefix Aggregate) by comparing the predicted range with the observed value (Figs. 6.1-(5),(6)). The number of values output by the RFR equals the number of decision trees that comprise the RFR. Therefore, the detection method of GAMPALv2 defines a predicted range based on multiple pre-

dicted values from the RFR. Comparing the predicted range and observed flow size for each PA, GAMPALv2 detects anomalies. This improvement enables anomaly detection for each PA.

6.2 Prediction Methodology

6.2.1 Learning Approach in GAMPALv2

To address the requirements of reducing the computational complexity in training and prediction (Sect. 5.3-(a)), and using the semantics related to time (Sect. 5.3-(b)), GAMPALv2 defines the time feature. Note that since the time slot of the flow size vector is a five-minute time interval (e.g., 0:00 to 0:05), as in GAMPALv1, the time feature is generated from the starting time of each time slot. As shown in Fig. 6.2, the model predicts the flow size from the time feature of each time slot by modeling the relationship between the time feature and flow size. This method reduces the dimension of variables input to the model from 288 to 7, which is expected to reduce the computational complexity.

GAMPALv2 adopts RFR as the prediction model. As described in Sect. 6.1, it is known that RFR is suitable for modeling the influences of multiple features in explanatory variables on the objective variable. In addition, the time feature combines correlated variables, such as date and day of the week or day of the week and business day. This condition is called multicollinearity, in which combinations of correlated variables exist in the

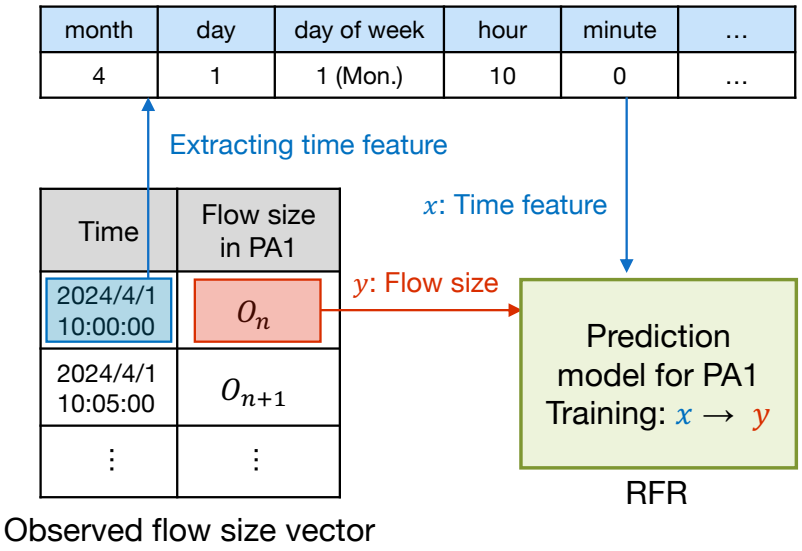


Figure 6.2: Leaning approach in GAMPALv2.

explanatory variables, and it can make regression modeling difficult. RFR is known to be less affected by multicollinearity and shows high prediction accuracy in the preliminary experiments described in Sect. 6.2.2.

This learning approach also minimizes the impact of missing data. In GAMPALv1, time series data consisting of continuous flow sizes is used as input, so even partial data loss prevents accurate learning and prediction. Therefore, if data for a short period is missing, continuous time series data of the flow size, including that period, cannot be used as training data or as input for prediction. In GAMPALv2, the data around missing data can be used as training data because the data are input at each time slot. In addition, since the input is not the flow size vector but the time feature, it is possible to predict after the period of missing data. Furthermore, it is

possible to exclude the period when an anomaly is reported from the training data. In this way, GAMPALv2 mitigates the restrictions on the training and input data for the prediction.

6.2.2 Preliminary Experiments

In this dissertation, a preliminary experiment is conducted to establish the most suitable prediction model for the proposed learning method. This experiment implements three types of prediction models based on NN (Neural Network)-based method, tree-based ML (Machine Learning) method, and kernel method, and compares them with the model in GAMPALv1. As the NN-based method, LSTM-RNN [51] used in GAMPALv1 is adopted. As the tree-based ML method, RFR [50] is adopted. The background to the adoption of RFR as a tree-based ML method is described in Sect. 8.2 later. Random Forest is an ensemble learning model consisting of multiple decision trees. The ensemble learning models achieve advanced prediction by combining many weak learners. Therefore, a random forest consisting of a small number of decision trees may have insufficient prediction accuracy. On the other hand, a model with too many decision trees may be influenced by outliers and lead to enormous computational complexity. Based on these perspectives, this experiment implements an RFR model consisting of 200 decision trees and defines the mean value of 200 output values as the predicted value. As the kernel method, SVR (Support Vector Regression) [52] is adopted. SVR is a regression model based on SVM (Support Vector Ma-

chine). SVM is a standard algorithm in ML analysis that is known as a simple and powerful model.

This preliminary experiment uses observed flows and BGP RIB (Border Gateway Protocol Routing Information Base) obtained from the WIDE backbone network [44]. Each of the three prediction models predicts the flow size on October 10, 2018, which is used in the evaluation of GAMPALv1 as the date when anomalies were not reported. These models are trained with the data for ten days (September 24 - October 3), during which anomalies were not reported. In addition, to compare with GAMPALv1, a prediction model based on LSTM-RNN, which uses a flow size vector as its input, is also implemented. Note that this model contains two LSTM layers, and the model size is larger than that of GAMPALv1. This model is trained with data from the same dates, and the time series data of the flow size on October 9 is used as input for prediction. This experiment uses a single time series of flow without prefix aggregation. The evaluation of this experiment uses the NSD described in Sect. 4.1.2 to compare the prediction accuracy with GAMPALv1. Although the NSD is unsuitable for anomaly detection as described in Sect. 6.3.1 later, it is suitable for evaluating prediction accuracy. The NSD represents the ratio of the difference between the predicted and observed flow size vectors; a smaller NSD indicates a smaller difference.

Table 6.1 shows the results of the preliminary experiment. Among the three models, the model based on RFR is the best, with an NSD of 0.335. Compared to GAMPALv1, the NSD is reduced by 47.7%. This result shows

Table 6.1: Result of preliminary experiments.

Method	GAMPALv1	GAMPALv2		
Model	LSTM	LSTM	RFR	SVR
NSD	0.640	0.765	0.335	0.656

that the RFR is the most suitable for the learning method for GAMPALv2. As described in Sect. 6.2, RFR is suitable for modeling the influences of multiple features in explanatory variables on the objective variable and is less affected by multicollinearity. On the other hand, the result of LSTM-RNN with the learning method in GAMPALv2 shows a larger difference from observed values than in GAMPALv1. The reason for this is that the input in the proposed method is not time series data. Based on the results of this experiment, GAMPALv2 adopts a model based on the RFR.

6.3 Detection Method

6.3.1 Predicted Range

To address the requirements of consideration of the range of possible values for the flow size(Sect. 5.3-(c)), and detection of when and in which PAs an anomaly occurred (Sect. 5.3-(d)), GAMPALv2 defines *predicted range*. The predicted range indicates the range of possible values for the flow size and consists of an upper and a lower limit, so that the observed values that do not include anomalies take values within the range. As described in Sect. 6.2.2, the RFR comprises multiple decision trees. Therefore, the number of values

output from RFR equals the number of decision trees that comprise RFR. GAMPALv2 uses the cluster of outputs from the RFR to define a predicted range, which is the range of possible values for the flow size.

In GAMPALv2, several methods for calculating the predicted range from the clusters of outputs from RFR are considered. These include setting the lower and upper limits of the predicted range to the values calculated from the mean and standard deviation of the cluster of outputs or to the values of percentiles for the cluster of outputs. Among the various methods, GAMPALv2 adopted the 0th percentile of the cluster of output values, i.e., the minimum and maximum values are set as the lower and upper limits of the predicted range, respectively. GAMPALv2 adopts the minimum and maximum values in the cluster of outputs for each timestamp as the lower and upper limits of the predicted range, respectively.

Figure 6.3 shows the concept of the predicted range in GAMPALv2. Anomalies are detected by comparing the predicted range with the observed flow size at each time slot. In the case that a cluster of outputs is distributed over a wide range, the validity of the flow size prediction at that time is low because the range of possible values for the flow size is wide. On the other hand, in the case that the cluster of outputs is distributed over a narrow range, the validity of the flow size prediction at that time is high because the range of possible values for the flow size is narrow. This detection method considers the validity of prediction depending on the PA and period.

GAMPALv2 detects anomalies by comparing the observed flow size

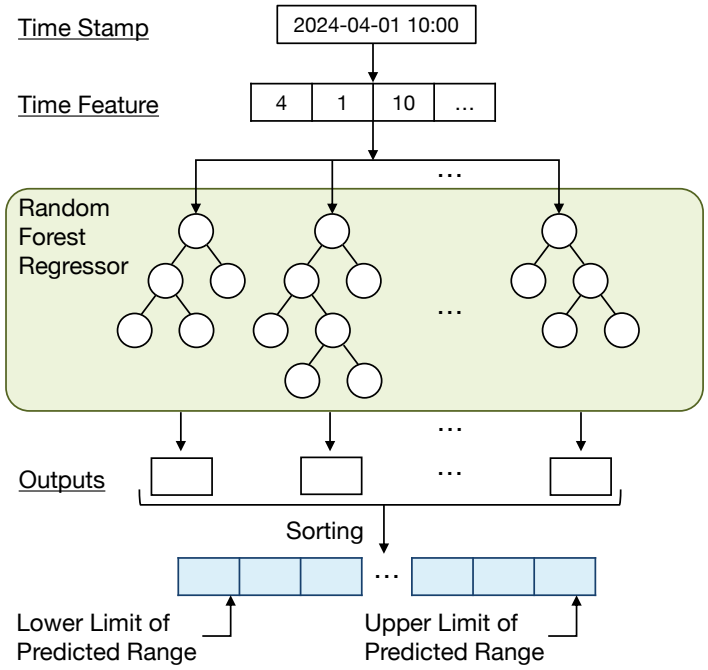


Figure 6.3: Definition of predicted range.

with the predicted range. Figure 6.4 shows an example of the observed flow size and the predicted range. The red line indicates the observed flow size, and the blue area indicates the range of the predicted flow size. All observed values in Fig. 6.4 are within the predicted range. It indicates that the flow sizes on this day follow the prediction based on the past behavior of the flow size. Figure 6.5 shows an example of the observed flow size and the predicted range in the case of an anomaly. The observed flow size in Fig. 6.5 exceeds the upper limit of the predicted range significantly for most of the period. It indicates that the flow size on this day is much larger than the prediction based on the past behavior of the flow size. This example shows that GAMPALv2 detects an anomaly when the observed values continuously

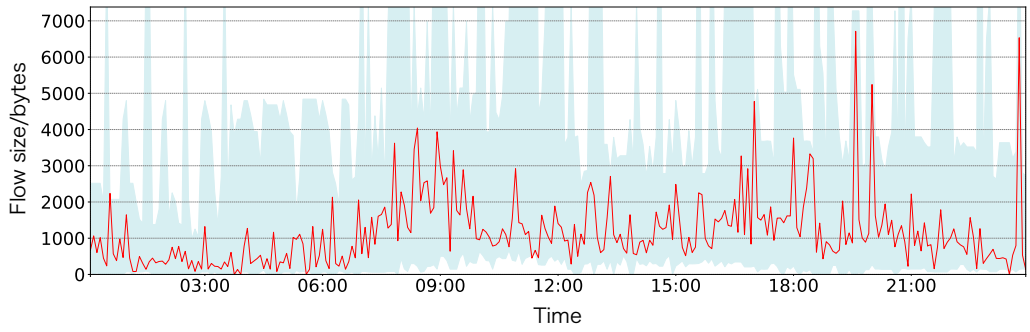


Figure 6.4: Example of observed values within predicted range.

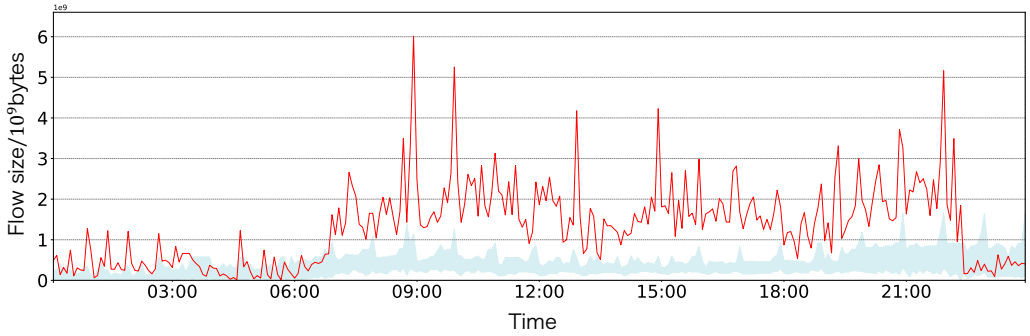


Figure 6.5: Example of observed values exceeding predicted range.

deviate from the predicted range. However, since the pattern of flow size is peaky and jaggy, the observed values may temporarily be out of the predicted range. If such behavior is detected as an anomaly, FPs (False Positives) will occur frequently. Therefore, the detection method should be filtered based on the duration of the out-of-range period and the amount of differences from the lower and upper limits. The details of this filtering are described in Sect. 6.3.2.

6.3.2 Filters for Preventing False Positives

To prevent FPs, GAMPALv2 defines a filter for the duration of out-of-range periods and a filter for the difference values.

First, the difference value at each time slot is calculated. This step compares the observed value at each time slot with the predicted range. If the observed value is within the predicted range, the difference value at the time slot is zero. If the observed value is above the upper limit of the predicted range, the difference value at this time is the difference between the observed value and the upper limit of the predicted range. If the observed value is below the lower limit of the predicted range, the difference value at this time is the difference between the observed value and the lower limit of the predicted range. The calculated difference values are time series data with the same length as the observed and predicted flow sizes.

Next, the difference values are smoothed by a moving average. Since the pattern of flow sizes is peaky and jaggy, the pattern of difference values is also peaky and jaggy. Therefore, the simple moving average is used to smooth the short-term peaks and to reduce FPs. The window length of the simple moving average is defined as a coefficient w. Let x_t be the value of the t-th slot of a time series data and the window of moving average $w = 2v + 1$, the value of the t-th slot of the time series data applied to the moving average is shown by the following equation.

$$a_i = \frac{1}{w} \sum_{t=i-v}^{i+v} x_t. \tag{6.1}$$

Next, if the difference values exceed the threshold, the out-of-range period is detected as an anomaly. Since a period with a wide predicted range means that the validity of the prediction is low, it is appropriate to increase the threshold for the difference values. Therefore, the threshold is defined as the width of the predicted range for each time slot (i.e., the differences between the maximum and minimum values of the predicted range) multiplied by a coefficient α. This dissertation introduces this threshold as the adjusted predicted range. Let x_{max} be the maximum value and x_{min} be the minimum value of the predicted range in a given time slot, the adjusted predicted range in the time slot is $\alpha(x_{max} - x_{min})$.

Finally, out-of-range periods with less than T minutes are filtered out. As with the smoothing, this process filters out short out-of-range periods and reduces FPs. After these processes, the out-of-range periods not filtered out are detected as anomalies.

6.4 Implementation of GAMPALv2

Figure 6.6 shows the overall procedure of GAMPALv2. GAMPALv2 consists of three major components: preprocessing (Fig. 6.6-(1)), training and prediction (Fig. 6.6-(2)), and anomaly detection (Fig. 6.6-(3)). Preprocessing shown in Fig. 6.6-(1) is the same as GAMPALv1, and GAMPALv2 defines

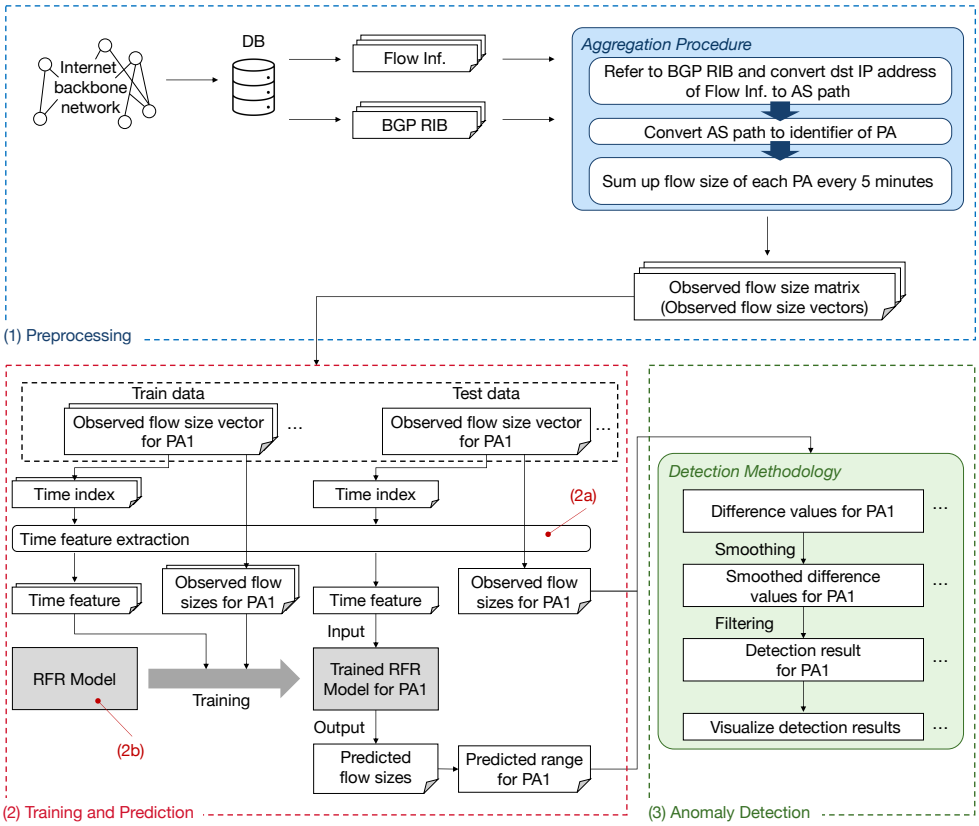


Figure 6.6: Overall procedures of GAMPALv2.

the variable k of prefix aggregation as three, as described in Sect. 3.2.

Training and prediction shown in Fig. 6.6-(2) and anomaly detection shown in Fig. 6.6-(3) are the proposed methods in GAMPALv2. The component of training and prediction is implemented according to the method described in Sect. 6.2. Figure 6.6-(2a) shows the extraction of the time feature. As described in Sect. 6.1, the time feature, which is the explanatory variable of the prediction model, consists of seven items: month, day, hour, minute, day of the week, week of the month, and whether it is a business day.

The attribute of the day of the week is an integer from "0" for Monday to "6" for Sunday. It is generated using the `datetime` module [53] in Python. The attribute of the week of the month is an integer greater than or equal to "1" and indicates which week of the month the date on the timestamp. The attribute of the business day represents whether the date is a business day or a holiday, with "0" for Saturdays, Sundays, and national holidays in Japan and "1" for others. Figure 6.6-(2b) shows the RFR model before the training. As in the preliminary experiment in Sect. 6.2.2, the RFR model consists of 200 decision trees.

The anomaly detection component is implemented according to the method described in Sect. 6.3. The values of the variables for smoothing and filtering are described in Sect. 7.2.

GAMPALv2 is implemented on a server running Ubuntu LTS Server 20.04.6. The `nfdump` command version 1.6.17 [41] is used to convert the binary flow information into a readable format, and the `bgpdump` command version 1.4.99.13 [42] is used to convert the binary BGP RIB into a readable format. A dataset is created from the flow information and the BGP RIB using PostgreSQL 12.9. The flow aggregation, prediction, and evaluation programs are implemented in Python 3.7.8. The prediction models are implemented with the `scikit-learn` libraries [54].

Chapter 7

Evaluation of GAMPALv2

7.1 Dataset

In evaluating GAMPALv2, the flow information and the BGP RIB (Border Gateway Protocol Routing Information Base) are exported from the WIDE backbone network (AS2500) [44] as in evaluating GAMPALv1 described in Sect. 4.3.

This evaluation uses the following three incidents as well as the evaluation of GAMPALv1: (i) a connection failure on YouTube on October 17, 2018, (ii) a campus festival on November 22-25, 2018, and (iii) a DDoS (Distributed Denial of Service) attack between June and July 2019.

7.2 Validation of Anomaly Detection

In the detection method shown in Sect. 6.3, there are three coefficients: w, the window length of the moving average, T, the coefficient in the filtering by the duration of out-of-range periods, and α, the coefficient in the filtering by the difference values of out-of-range periods. In the evaluation, the w is defined as 3, and the T is defined as 30 minutes, considering the trade-off between FPs (False Positives) and FNs (False Negatives). If w is too large, it may round off changes other than short-term peaks and miss the influence of the anomaly on the flow size, causing FN. In addition, if the value of the short-term peak of the flow size is too large, it may become indistinguishable from the long-term deviation after smoothing is applied, causing FPs. In this perspective, w is defined as 3 in this evaluation. If T is too long, it may cause FNs. On the other hand, if T is too short, deviations associated with short-term peaks during ordinary times may be detected and cause FPs. As described in Sect. 3.3, the average duration of DDoS failures is over 20 minutes. In addition, according to a guideline [55] published by MIC (Ministry of Internal Affairs and Communications) in Japan, telecommunication providers should, in principle, announce the failure within 30 minutes after it occurs. Therefore, anomalous behavior in Internet traffic that continues for more than 30 minutes can be regarded as a critical incident. Since smoothing by moving averages increases the duration of out-of-range periods, this evaluation defines the T as 30 minutes. Unlike w and T, it is difficult to define an appropriate value for α by an empirical rule. If α is too large, anomalies

are less likely to be detected, and FNs will increase. On the other hand, if α is too small, FPs may increase. Therefore, this evaluation compares the results of anomaly detection by applying the filter by the difference values of out-of-range periods with $\alpha = 0.05$, 0.15, and 0.25.

7.2.1 Connection Failure on YouTube

The details of the connection failure on YouTube are described in Sect. 4.3.1. While the connection failure on YouTube was occurring, the flow size in the PA (Prefix Aggregate) containing the flows destined for the university network, which is the observation point, should be smaller than that of the ordinary times. Unlike the other two incidents, the period at which the anomaly occurred is known for this incident.

Figure 7.1 shows the detection results of connection failure on YouTube. Figs. 7.1 (1)-(4) are the results of October 10, 2018, exactly one week before the connection failure, and the graphs shown in Figs. 7.1 (5)-(8) are the results of October 17, 2018, when the connection failure occurred.

In Figs. 7.1 (1) and (5), the blue areas show the predicted range and the red lines show the observed flow size in the PA containing flows destined for the university network. In Figs. 7.1 (2) to (4) and (6) to (8), the green areas show the adjusted predicted range ($\alpha = 0.05$, 0.15, or 0.25), and the purple lines show the difference values. The periods colored orange are out-of-range periods for longer than $T = 30$ minutes and are detected as anomalies.

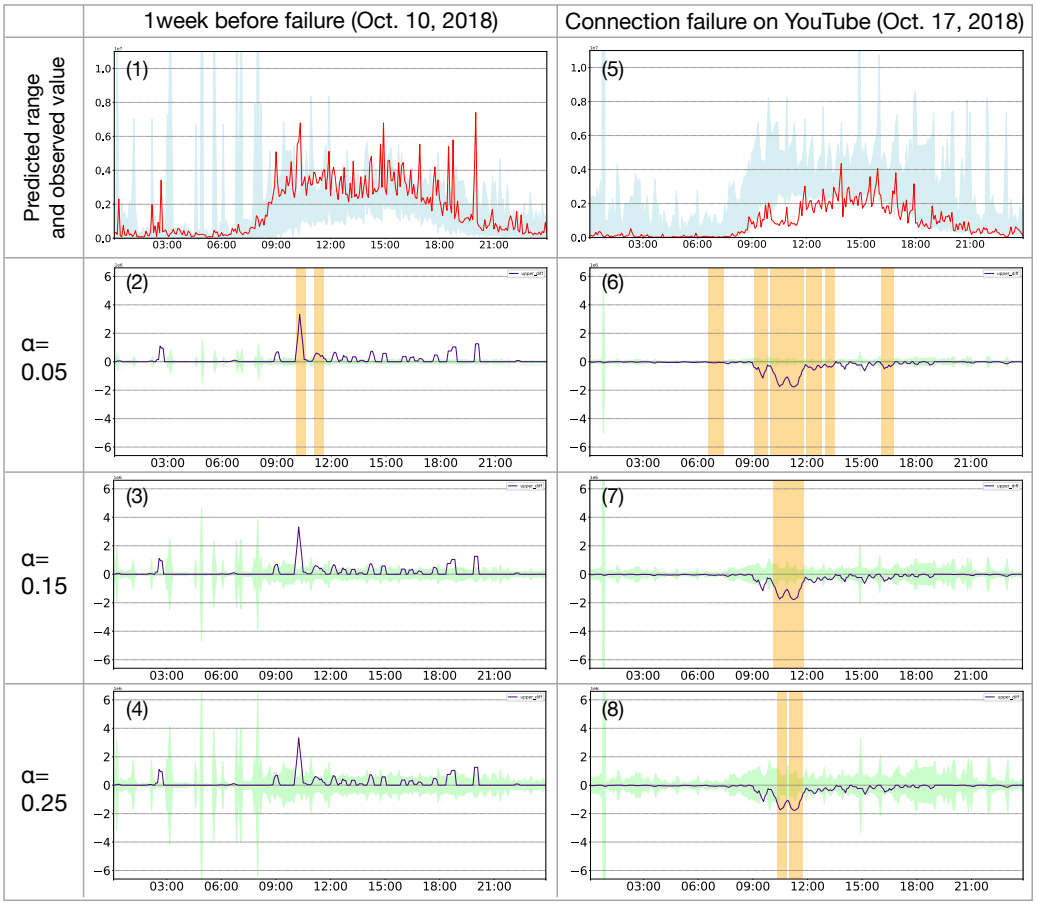


Figure 7.1: Detection result of connection failure on YouTube.

Note that the model that predicts the flow on October 10 is trained using the data for ten days from September 24 to October 3, 2018, and the model that predicts the flow on October 17 is trained using the data for ten days from October 1-10, 2018.

First, Fig. 7.1 (1) shows that the observed flow sizes remain within the predicted range for most of October 10. In Fig. 7.1 (2) where $\alpha = 0.05$, there are two FPs longer than $T = 30$ minutes and are detected as anomalies,

although the anomalies are not reported. When α is increased to 0.15, as shown in Fig. 7.1 (3), the duration of FPs is less than 30 minutes, and they are not detected as anomalies. On the other hand, as shown in Fig. 7.1 (5), the observed values fall below the lower limit of the predicted range in the morning. In particular, anomalies are detected between 10:00 and 12:00 in Figs. 7.1 (7) and (8). During this period, the connection failure on YouTube occurred. Therefore, the anomaly is detected correctly when α is larger than 0.15. For $\alpha = 0.05$, the adjusted predicted range is considered too small because anomalies are detected even when failures are not reported. These results indicate that α of 0.15 or 0.25 is appropriate for detecting this connection failure, while α of 0.05 may cause FPs. Since too large α may increase FNs, $\alpha = 0.15$ is considered appropriate for this case.

7.2.2 Event Traffic associated with Campus Festival

The details of the campus festival are described in Sect. 4.3.1. During the campus festival, network traffic may behave differently. Therefore, this section describes detection results for the PA containing the flows destined for the university network. Note that, unlike the case of connection failure on YouTube, it is difficult to estimate when this event affects the behavior of flow sizes.

Figure 7.2 shows the detection results of event traffic. Figures 7.2 (1) to (4) are the results from November 15 to 18, 2018, exactly one week before the campus festival, and the graphs shown in Figs. 7.2 (5) to (8) are the

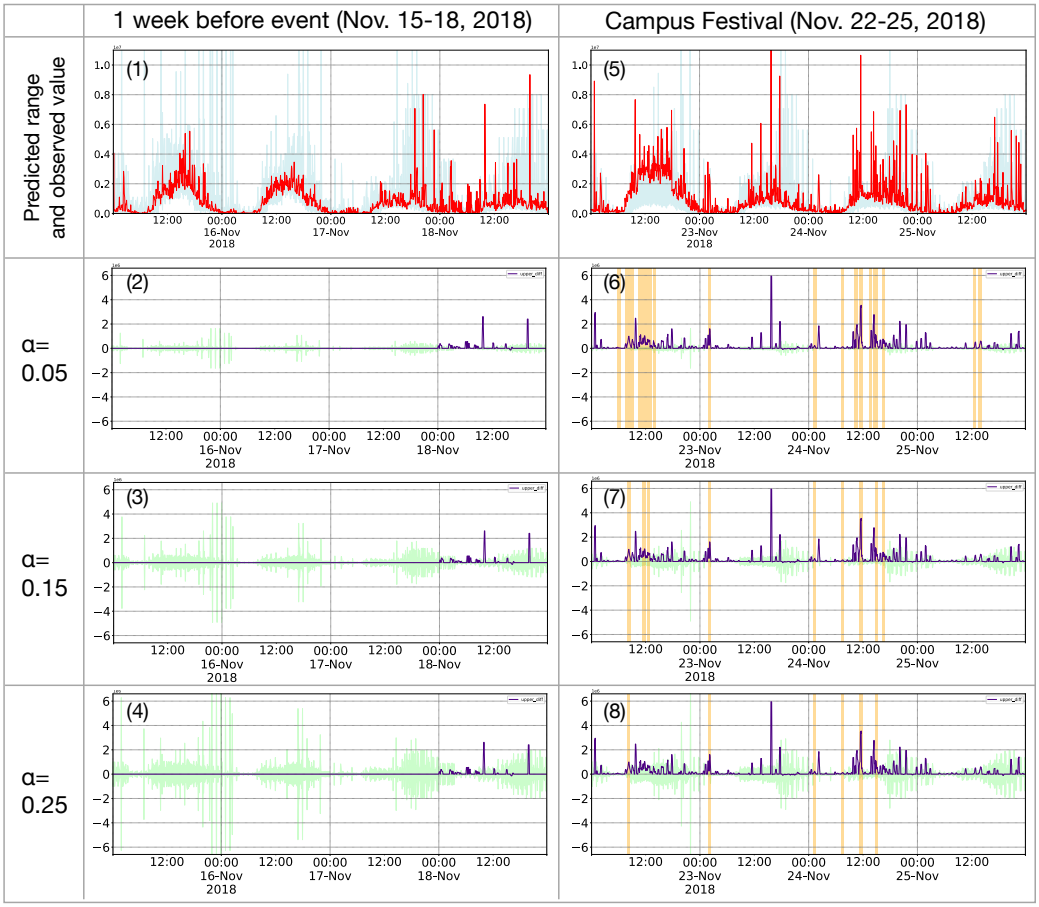


Figure 7.2: Detection result of event traffic.

results from November 22 to 25, 2018, when the campus festival was held.

As shown in Figs. 7.2 (2) to (4), any anomaly is not detected one week before the campus festival. On the other hand, Figs. 7.2 (6) to (8) show that GAMPALv2 detects many anomalies during the campus festival. This result indicates that GAMPALv2 detects the event traffic associated with the campus festival. As α increases, the number of periods detected as anomalies decreases. In addition, looking at the results for each day, anomaly

detection due to increasing flow size is frequent during the daytime on the first and third days. However, note that there is no information to confirm these results, such as the number of visitors on each day of the event.

7.2.3 DDoS Attack

The details of the DDoS attack are described in Sect. 4.3.1. As the case in GAMPALv1, the data from July 8, just before the university had to deal with the incident, is used as the date of the DDoS attack. Because of the long-term nature of this attack, May 20, exactly seven weeks before the DDoS, is chosen as the date when any anomaly is not reported. Through checking the ARMS (Apple Remote Management Service) flow, it is confirmed that the attack did not occur on this date. Since this attack exploited the university network as a reflector, many flows with spoofed source IP addresses destined for the university network should have been observed. Therefore, more flows than ordinary times should be observed in the PA, which contains the flows destined for the university network.

Figure 7.3 shows the detection results of the DDoS attack. Figures 7.3 (1) to (4) are the results on May 20, 2018, exactly seven weeks before the date of the DDoS attack, and Figs. 7.3 (5) to (8) are the results on July 8, 2019, the date of the DDoS attack.

As shown in Figs. 7.3 (2) to (4), no anomaly is detected seven weeks before the date of the DDoS attack. On the other hand, Figs. 7.3 (6) to

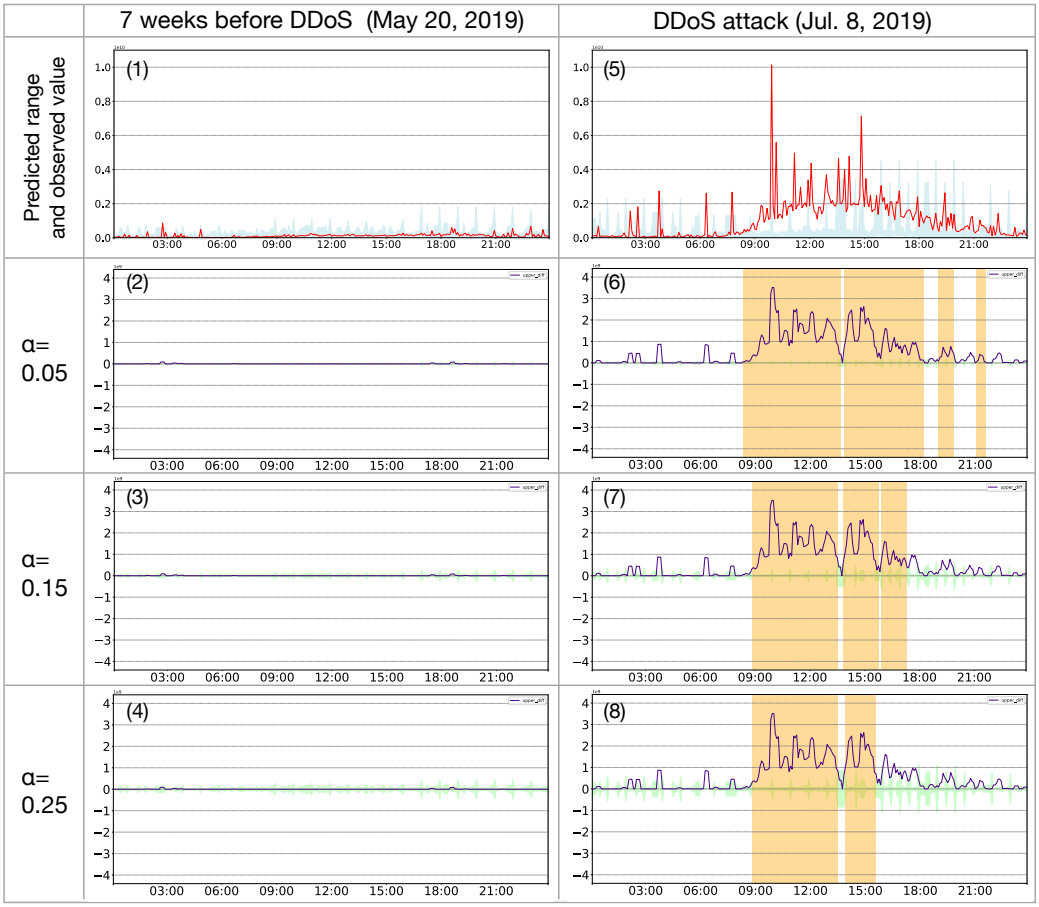


Figure 7.3: Detection result of DDoS attack.

(8) show that GAMPALv2 detects some anomalies on the date of the DDoS attack. The difference values are much larger than the adjusted predicted range. These results indicate that GAMPALv2 detects the DDoS attack. The huge observed flow size indicates the scale of this attack well.

Since this case is a DDoS attack using the university network as a reflector, amplified flows are sent from the university network in various directions. GAMPALv2 detects anomalies in many PAs at various times,

Table 7.1: Variable importance in the prediction model.

Feature	month	day	hour	min
Importance	0.011	0.083	0.432	0.023

Feature	day of the week	week of the month	business day
Importance	0.340	0.009	0.102

not only one destined for the university network.

7.2.4 Variable Importance in Prediction Model

The RFR (Random Forest Regressor) used as the prediction model in GAMPALv2 can evaluate the "Variable importance," which expresses the degree of influence of each explanatory variable on the objective variable. This section shows the variable importance of the prediction model trained on data from the WIDE backbone network. Table 7.1 shows the result of variable importance of the prediction model described in Sect. 6.2.2.

Among the seven items in the time feature, the top three that most influence the flow size are the hour, the day of the week, and whether it is a business day. In other words, these items particularly influence the flow size. A discussion of the evaluation results is described in Sect. 8.4.2.

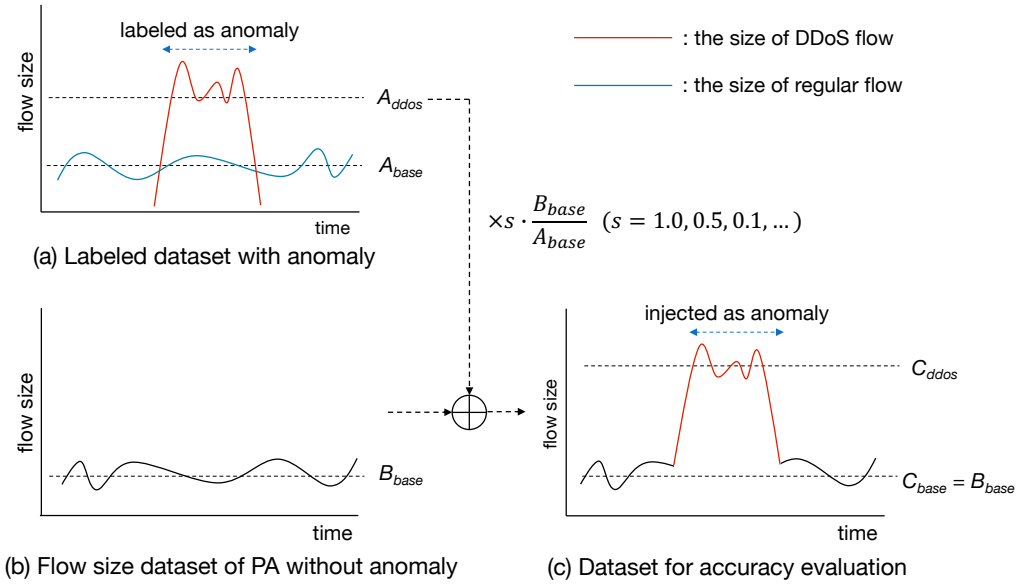


Figure 7.4: Procedures for generating dataset for evaluating anomaly detection accuracy.

7.3 Accuracy Evaluation

7.3.1 Dataset for Accuracy Evaluation

Section 7.2 describes the evaluation of detectability of the anomalies that occurred in the real world using the data from the WIDE backbone network. However, this dataset is not labeled data, and it does not necessarily mean that there is any anomaly other than the known one. Even if GAMPALv2 detects an anomaly on a date and time when no anomaly is reported, the detection cannot be confirmed as an FP. Therefore, labeled data is necessary to evaluate anomaly detection accuracy.

Although the evaluation of GAMPALv2 requires long-term labeled

data for a few days to 10 days or more, most labeled data is only a few minutes to a few hours. Therefore, this dissertation generates a dataset for the accuracy evaluation of GAMPALv2. Figure 7.4 shows the procedures for generating a dataset for evaluating anomaly detection accuracy. This dataset is generated by injecting a labeled dataset with anomalies (Fig. 7.4-(a)) into a flow size dataset of a PA without anomaly (Fig. 7.4-(b)). In the dataset (a), let A_{base} be the average size of regular flows, and A_{ddos} be the average size of DDoS flows. In the dataset (b), let B_{base} be the average size of flows. In this case, the sizes of DDoS flows in the dataset (a) are multiplied by $s \cdot B_{base}/A_{base}$ (s = 1.0, 0.5, 0.1 ...) and they are added to the dataset (b). Consequently, a pseudo-labeled DDoS dataset for accuracy evaluation (Fig. 7.4-(c)) is generated. s is a variable that adjusts the scale of the anomalous flows to be injected. By reducing the value of s, a dataset with little changes in flow size due to anomalies can be generated.

This dissertation adopts one hour of the data from CIC-DDoS2019 [24], the labeled dataset consisting of regular and DDoS flows as (a). In addition, (b) is generated from the data retrieved from the WIDE backbone network. The data from May 20, 2019, described in Sect. 7.2.3, is used for (b). First, GAMPALv2 is applied to this data. The flows observed on May 20 are classified into 14,539 PAs, and anomalies are detected in 296 PAs. These anomalies cannot be determined as TPs (True Positives) or FPs. On the other hand, no anomaly is detected in the remaining 14,243 PAs. Into these flow sizes, (b) is injected. In the evaluation, anomaly flows are injected at five points

at 6, 9, 12, 15, and 18 o'clock in each PA, and the anomaly detection accuracy is evaluated. Therefore, this evaluation verifies how many anomalies out of 71,215 anomalies (14,243 PAs $\times$ five anomalies) can be detected by GAMPALv2. If an anomaly is detected at the point where it is injected, it is a TP. If an anomaly is not detected at the point where it is injected, it is a FN. This detection evaluation counts the number of TPs and FNs and calculates the recall, which indicates the percentage of detectable anomalies out of all anomalies. Note that since FPs cannot occur in this evaluation, the precision, which is the ratio of the number of TPs to the total number of TPs and FPs, is not calculated, and detectability is evaluated with recall.

7.3.2 Result of Accuracy Evaluation

Table 7.2 shows the evaluation results of the detection accuracy of GAMPALv2. Each column shows the detection results for each scale of injected DDoS flows. The top three rows represent TP, FN, and recall, respectively, and the bottom six rows show the distribution of the number of TPs per PA. For example, $TP = 5$ indicates the number of PAs for which all five anomalies are detected.

The results show that when injecting DDoS flows with $s = 1.0$, there are 66,322 TPs, and GAMPALv2 can detect 93.1% of the anomalies. Although not precisely comparable due to the different calculation methods, the average recall in GAMPALv1 described in Sect. 5.2.1 is 81.8%. All five anomalies are detected in 11,378 PAs or 80% of all PAs. When $s = 0.5$, the

Table 7.2: Result of accuracy evaluation.

Scale		×1.0	×0.5	×0.1	×0.05	×0.01
Number of TPs		66,322	62,535	42,680	32,816	23,238
Number of FNs		4,893	8,680	28,535	38,399	47.977
Recall		0.931	0.878	0.599	0.461	0.326
Number of PAs with	TP=0	115	220	1,681	3,522	6,763
	TP=1	115	271	1,747	2,386	1,672
	TP=2	268	586	2,045	1,951	1,224
	TP=3	687	1,188	2,296	1,867	1,224
	TP=4	1,680	2,362	2,415	1,890	1,354
	TP=5	11,378	9,616	4,059	2,685	2,006

number of detected anomalies is reduced to 62,535, while the recall is still 87.8%. In addition, four or more anomalies are detected in 84.1% of all PAs. The smaller the scale of the injected anomaly, the more difficult it is to detect, so the number of TPs and the recalls decrease. In the case of $s = 0.01$, the recall is 32.6%, and all five anomalies are detected in 47.5% of all PAs. Although the scale of anomaly flows is so tiny that the anomalies could not be detected in almost all PAs, all five anomalies are detected in 2,006 PAs. On the other hand, when $s = 1.0$, all five anomalies are detected in 80% of all PAs, while the anomalies are not detected in 115 PAs. Therefore, to identify future works of GAMPALv2, Sect. 7.3.3 analyzes the features of PAs that are difficult to detect anomalies.

7.3.3 Cause Analysis of FNs

The fact that GAMPALv2 fails to detect some anomalies in Sect. 7.3.2 means that the flow sizes injected with DDoS flows are within the predicted range in the long term. In other words, the predicted range of such a PA is too wide for the size of the DDoS flows. The width of the predicted range depends on the behavior of the training data of the model. Therefore, this section shows examples of the training data and prediction results for a PA with $TP = 0$ in Figs. 7.5 and 7.6.

In each figure, (1) shows the training data for ten days, and (2) shows the resulting predicted range and observed values. Since the observed values are small compared to the upper limit of the predicted range in Fig. 7.5 (2), the vertical axis scale in (2) is scaled down and is shown in Fig. 7.5 (3).

In Fig. 7.5 (1), flows are seldom observed during many periods, but some flows are observed irregularly. In such a case, it is difficult for the model to predict when a flow is observed, and the predicted range becomes too wide, as shown in Figs. 7.5 (2) and (3). As a result, the upper limit of the predicted range exceeds the injected DDoS flow sizes, and the anomalies are not detected. In Fig. 7.6 (1), the size of observed flows varies greatly depending on the day. In such a case, it is difficult to predict whether a large number of flows will be observed on a given day. As a result, the predicted range becomes too wide, and the anomalies are not detected, as shown in Fig. 7.6 (2). As shown in these examples, if the flow size in the

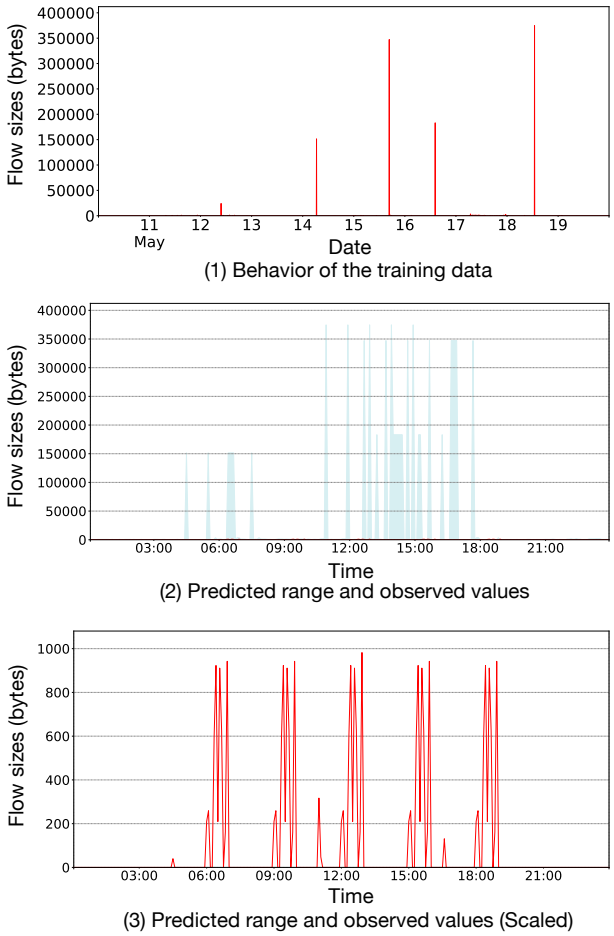


Figure 7.5: Example 1 of PA that failed to detect.

training data is irregular or the differences between the peaks and bottoms are significant, the predicted range tends to become too wide, and FNs occur. This problem is caused by the bias of the flow size for each PA. The future work corresponding to this problem is described later in Sect.8.4.1.

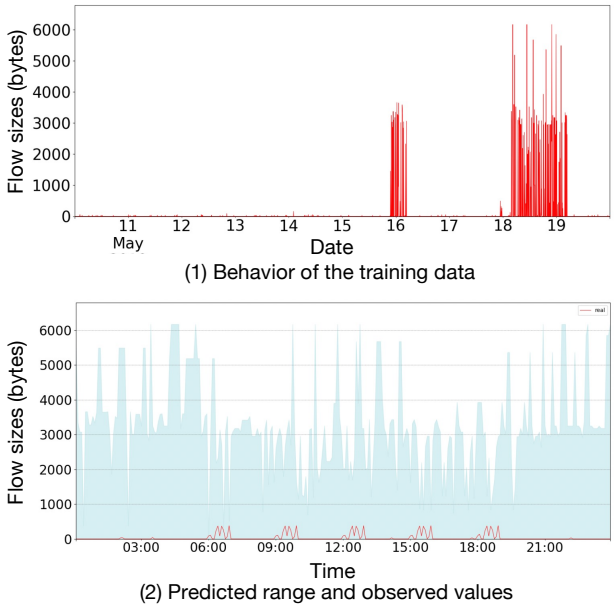


Figure 7.6: Example 2 of PA that failed to detect.

7.4 Computational Complexity

As described in Sect. 5.1.1, GAMPALv1 requires approximately four days on a GPU to train the model and calculate the predicted flow sizes for one day. On the other hand, GAMPALv2, based on RFR, takes about 23 minutes using an 8-core CPU. The reduction in processing time is attributed to changing the machine learning method of the prediction model from LSTM-RNN (Long Short-Term Memory Recurrent Neural Network) to RFR and reducing the number of dimensions of the data input to the model. As shown above, GAMPALv2 significantly reduces computational complexity compared to GAMPALv1, contributing to its improvements for practical use.

Chapter 8

Discussion

8.1 Discussion on Advantages of GAMPAL

The most important feature of GAMPAL is that it is an anomaly detection mechanism for the Internet backbone networks. For this reason, GAMPAL satisfies the requirements defined by considering the characteristics of the Internet backbone network. Table. 8.1 shows the comparison results with the accuracy evaluations of the existing research described in Sect. 2.2. Note that the scores in the table are cited from the evaluation results of each research [10,11,13,14]. Therefore, the evaluation methods and datasets differ from each other.

Table 8.1: Comparison with the results of accuracy evaluations.

	[10]	[11]	[13]	[14]	GAMPALv1	GAMPALv2
Recall	98.0%	89%	98.9%	99.4%	81.8%	93.1%

Among the existing research, [10], [11], [13], and [14] show recalls in the accuracy evaluations. Note that [12] does not describe the accuracy evaluation. The methods proposed in [10], [13], and [14] show high detection accuracy, with recalls above 98% for all of them. As described in Sect. 7.3.2, the recall of GAMPALv2 is 93.1%. GAMPALv2 does not show an advantage in accuracy evaluation relative to the existing methods. Based on the above, GAMPAL has room for improvement in terms of accuracy. However, the methods proposed in [10], [11], [13], and [14] do not have versatility to anomalies. Thus, GAMPAL shows a high recall comparable to the other existing methods while showing versatility.

Furthermore, GAMPAL does not use labeled data and detects anomalies by focusing on the changes of the flow size, which is the number of bytes of flow information. This is a unique feature of GAMPAL. As mentioned in Sect. 4.4, the defense-in-depth is a key strategy in cybersecurity. Since GAMPAL has different characteristics from the other existing methods, it is expected to be used with signature-based methods or anomaly detection methods with classification models in the defense-in-depth strategy. For the use case of GAMPAL combined with other anomaly detection methods, it is suitable for GAMPAL to have a smaller computational complexity. In this context, the reduction of computational complexity in GAMPALv2 is a major contribution to its practical use.

8.2 Discussion on Machine Learning Models in GAMPALv2

Sect. 6.2.2 describes the preliminary experiments to establish the prediction model for GAMPALv2. In this preliminary experiment, several different types of machine learning methods are compared. Among them, RFR (Random Forest Regressor) is selected as the tree-based model. However, besides RFR, Light-GBM is also well known as a tree-based model [56]. This section compares RFR and Light-GBM and discusses the reasons for choosing RFR.

First, the prediction accuracies of RFR and Light-GBM are compared in the same way as the preliminary experiments. For RFR, the mean and median values of the cluster of output values are calculated, and the difference between each and the observed values is evaluated. Light-GBM is implemented using the `LGBMRegressor` [57] from the Light-GBM library in Python [58]. For Light-GBM, the default output value is used as the predicted value. Light-GBM outputs the cumulative predicted value. Let $\hat{y}$ be the final prediction value, $\hat{y}_0$ be the initial prediction, N be the total number of trees, η be the learning rate, and $f_n(x)$ be the prediction for the nth tree, the cumulative output value is shown in the following equation [59].

$$\hat{y} = \hat{y}_0 + \sum_{n=1}^{N} \eta \cdot f_n(x). \tag{8.1}$$

Table 8.2 shows the results of the accuracy evaluation. The accuracy of the Light-GBM is better than that of the RFR with the median value, but

Table 8.2: Comparison results of prediction accuracy of tree-based models.

Model	RFR (mean)	RFR (midian)	Light-GBM
NSD	0.335	0.350	0.348

not better than that of the RFR with the mean value. Although the results may vary depending on the parameter tuning, the prediction accuracies of RFR and Light-GBM are close to each other in predicting the flow size vector. Some research has shown that Light-GBM has higher prediction accuracy than RFR for many datasets [60]. However, in the prediction task in GAMPAL, the number of dimensions of the explanatory variables is small, and it is thought that the advantage of Light-GBM is not demonstrated.

One of the reasons for selecting RFR in this dissertation is its simplicity. RFR and Light-GBM are tree-based models with many common parameters, such as the "estimators," which is the number of decision trees. However, Light-GBM generates the decision trees that comprise the model based on the gradient boosting method. Therefore, Light-GBM has the "learning rate," an important parameter related to gradient boosting. RFR does not have this parameter. In addition, while the trees that comprise RFR are independent of each other, the trees that comprise Light-GBM are generated by propagating the error of the previous decision tree by gradient boosting. Thus, Light-GBM is a more complex model than RFR. Furthermore, GAMPALv2 defines the predicted range using the cluster of output values from RFR. However, Light-GBM generates trees cumulatively by the gradient boosting, so the output values are more complex than those of RFR. For

these reasons, this dissertation selects RFR as the model in GAMPALv2.

Although Light-GBM is not used in this dissertation, it is a powerful and fast model. In addition, while RFR reduces variance through Bagging (Bootstrap Aggregating), Light-GBM reduces bias through Boosting. Based on these features, Light-GBM may be helpful in future work.

8.3 Discussion on Metrics for Internet traffic

As described in Sect. 3.1, GAMPAL employs the number of bytes in a flow as a metric. The sum of the bytes of the flows observed in each time slot is defined as the flow size, and the time series data of the flow size is defined as the flow size vector. The model predicts the flow size and detects anomalies by comparing the predicted flow size vector with the observed flow size vector. This section considers GAMPAL in terms of some metrics that represent the volume of Internet traffic.

Besides the number of bytes, the number of packets and sessions are also metrics that represent the volume of Internet traffic, and the influence of an anomaly is different for each metric. Major threat models in the Internet backbone networks include cyber attacks that occupy bandwidth. Since such anomalies are reflected in the bytes well, GAMPAL adopts the number of bytes as a metric. As a result, GAMPAL can detect traffic decreases due to connection failures and increases due to event traffic and DDoS (Distributed Denial of Service) attacks based on the number of bytes. However,

not only bytes but also packets and session counts may be effective metrics for anomaly detection. For example, spam mail and port scan attacks are difficult to detect with the bytes because they are insensitive to the number of bytes. This is a limitation that GAMPAL faces. However, it may be more effective to use the number of packets instead of bytes as a metric for GAMPAL. Thus, there is room for further consideration of anomaly detection using other metrics besides the bytes. In addition, since GAMPALv2 reduces computational complexity, a method for performing anomaly detection using multiple metrics in parallel may also contribute to improving the accuracy and versatility of GAMPAL.

8.4 Contributions by Proposed Methods

8.4.1 Contribution by Prefix Aggregation

One of the most important points of GAMPAL is the definition of prefix aggregation. Prefix aggregation is suitable for spatial aggregation of Internet traffic. As described in Sect. 3.2, this is because of the locality of Internet traffic.

This section describes the advantages and future work of prefix aggregation in more detail. By aggregating Internet traffic, Internet traffic is divided into several classes. If a large amount of traffic belongs to a class due to an aggregation method, anomalous traffic in the PA may be buried by other regular traffic. Therefore, the aggregation method must be used,

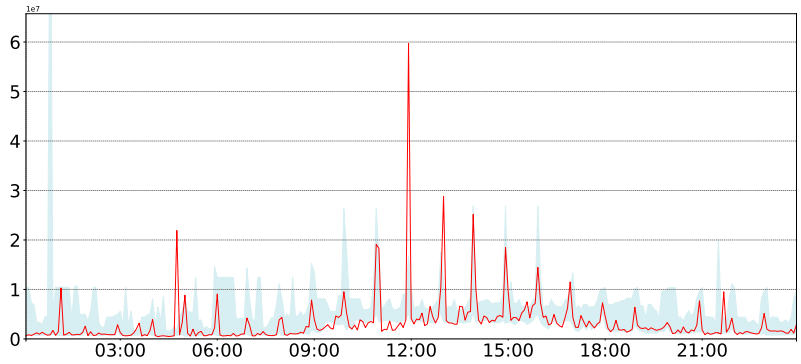


Figure 8.1: Prediction result for connection failure with a single class.

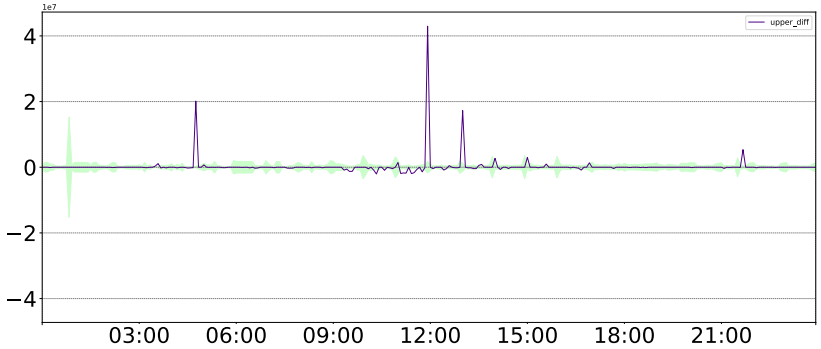


Figure 8.2: Detection result for connection failure with a single class.

in which traffic frequently observed is divided into fine-grained classes, and traffic not frequently observed is divided into coarser-grained classes. Due to the locality of Internet traffic, flows destined to IP addresses close to the observation point are more frequently observed. Since prefix aggregation can divide such flows into fine-grained PAs (Prefix Aggregates), it is a suitable aggregation method for the locality of Internet traffic.

To confirm the contributions of prefix aggregation, the anomaly detection results without prefix aggregation are shown in Figs 8.1-8.4. Figure 8.1

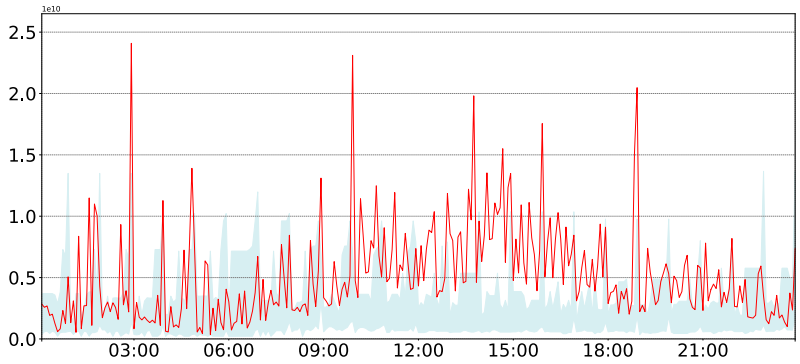


Figure 8.3: Prediction result for DDoS with a single class.

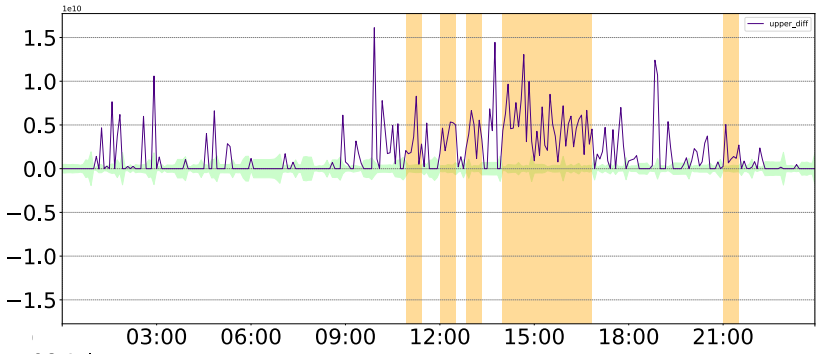


Figure 8.4: Detection result for DDoS with a single class.

shows the prediction results, and Fig. 8.2 shows the detection results, without prefix aggregation on the day of the YouTube connection failure. From 10:00 to 12:00, when the connection failure occurred, the observed flow size remained close to the lower limit of the predicted range, but an anomaly could not be detected. This is because the traffic reduction caused by the connection failure is buried by the other traffic in the case of aggregation into a single class.

Figure 8.3 shows the prediction results, and Fig. 8.4 shows the de-

tection results, without prefix aggregation on the day of the DDoS attack. Since this DDoS attack is large-scale, anomalies are detected even when aggregating into a single class. However, the influence of the DDoS attack is not apparent in the case without prefix aggregation compared to the case with prefix aggregation. In the case with prefix aggregation, the observed flow size is continuously out of range for most of the daytime. On the other hand, in the case without prefix aggregation, the out-of-range period of the observed flow size continues for a short time.

The evaluation results show that applying prefix aggregation makes the influence of anomalies on the flow sizes clear. This is because prefix aggregation is a flow aggregation method considering the locality of Internet traffic.

However, even when the prefix aggregation is applied, traffic may be biased to a particular PA. This is because the number and payload of traffic destined for IP addresses associated with primary services or networks to which an observation point belongs are enormous. This problem cannot be solved by increasing the number of ASs that define prefix aggregation. As future work, the detection accuracy of GAMPAL may be improved by further classification of these flows in more detail.

8.4.2 Contribution of Defining Time Feature

One of the differences between GAMPALv1 and GAMPALv2 is the definition of the time feature, which is a measure to address the issue that GAMPALv1 does not use the semantics related to time to train the model. This section considers the contribution of using the semantics related to time.

In Sect. 7.2.4, the variable importance of the prediction model trained on the time feature is evaluated, and the top three that most influence the flow size are the hour, the day of the week, and whether it is a business day. In general, the flow size tends to be larger during the daytime because there are a large number of users. On the other hand, the flow size tends to be smaller during the nighttime because there are a small number of users. In addition, there are times during the daytime when the use of the network is concentrated. Therefore, the influence of hour on flow size is large. Furthermore, it is known that Internet traffic patterns in Japan tend to show a weekly periodicity [61]. Therefore, the day of the week and business day also greatly influence the flow size. The results of the variable importance are consistent with the characteristics of Internet traffic, which indicates that the prediction model learns these characteristics.

Thus, the evaluation results in Sect. 7.2.4 are consistent with the characteristics of Internet traffic. In other words, the learning method in GAMPALv2, which models the relationship between the time feature and the flow size, is effective in learning the behavior of flow size. In future work,

changing the variables that comprise the time feature based on evaluating the variable importance may improve the accuracy of prediction and anomaly detection.

8.5 Considerations on Definition of Predicted Range

Considerations on the definition of predicted range as described in Sect. 6.3.1, GAMPALv2 sets the lower and upper limits of the predicted range to the minimum and maximum values of the cluster of output values from the RFR. In addition, the filter has three different coefficients to prevent FPs (False Positives), described in Sect. 6.3.2. In Sect. 7.2, $w = 3$, $T = 30min$, $\alpha = 0.05$, 0.15, and 0.25 are used to evaluate the validation of anomaly detection. However, the definition of the predicted range and the values for the filters may not be suitable for all data in any environment. These conditions should be adjusted and determined based on the background, such as the environment in which the data is collected, anomalies to be detected, and use cases. For example, if the detection target is limited to large-scale attacks, it may be suitable to increase the threshold of the filter or to define the predicted range so that its width becomes larger. In addition, a use case that defines multiple predicted ranges and evaluates the degree of risk according to the predicted range from which the observed value deviates could be considered. Similarly, a use case could be to define multiple filters with different parameters. In these use cases, it is possible to take countermeasures against anomalies

according to the results of the risk assessment. As described above, future work on the definition of predicted ranges and filters is possible, and these approaches may further improve GAMPAL.

Chapter 9

Conclusion

The Internet backbone network contains a large amount of traffic originating from various kinds of users and services. In addition, the Internet backbone network encounters anomalies such as network facility failures, disturbances from social phenomena, and cyber attacks. To operate the Internet backbone network stably, a general-purpose mechanism for detecting various anomalies in traffic information must be established.

This dissertation proposes GAMPAL, an anomaly detection mechanism for Internet traffic. For scalability to the number of entries in the BGP RIB (Border Gateway Protocol Routing Information Base), GAMPALv1 introduces prefix aggregation and aggregate flows spatially. In addition, GAMPAL establishes a prediction model for flow sizes based on past flow sizes. By comparing predicted and observed flow sizes, GAMPAL detects anomalies. GAMPAL has four characteristics as follows: (i) scalability to the Internet, (ii) consideration for the locality of Internet traffic, (iii) versatility for any

kind of anomaly, and (iv) unnecessity of labeled data.

First, GAMPALv1 [21] is proposed, and then GAMPALv2 [22] is proposed as an improved version that solves the issues of GAMPALv1. GAMPALv1 establishes a prediction model based on LSTM-RNN (Long Short-Term Memory Recurrent Neural Network) and proposes NSD (Normalized Summation of Difference) as an evaluation indicator. However, GAMPALv1 requires significant computational time, particularly for training the prediction model and outputting predicted values. Moreover, GAMPALv1 has difficulty defining standard criteria of the NSD for anomaly detection for all PAs and cannot identify the PA (Prefix Aggregate) and the period in which the anomaly occurred.

To solve these issues, GAMPALv2 is proposed as an improved anomaly detection mechanism. GAMPALv2 defines four requirements as follows: (a) reducing the computational complexity in training and prediction, (b) using the semantics related to time, such as date, time, and day of the week, to train the model, (c) consideration of the range of possible values for the flow size, and (d) detection of when and in which PAs an anomaly occurred. To address the requirements (a) and (b), GAMPALv2 adopts RFR (Random Forest Regressor) as the prediction model and defines the time feature. To address the requirements (c) and (d), GAMPALv2 defines the predicted range based on the predicted values generated by decision trees in the RFR and detects anomalies, such as whether the predicted values are within the predicted range.

This dissertation presents that GAMPALv2 can detect service failures, event traffic, and DDoS (Distributed Denial of Service) attacks. In addition, this dissertation confirms that the computational time for GAMPALv2 is reduced by 99.6% compared to GAMPALv1 by improving the prediction model. Prediction accuracy is also improved by using semantics such as date, time, and day of the week to predict the flow size. Furthermore, through improvements in the detection method, GAMPALv2 can detect when and at which PA anomalies occur. The proposals in this dissertation have contributed to the practical use of GAMPAL.

As future work for GAMPAL, the anomaly detection using metrics other than bytes should be tested. As described in Sect. 8.3, other metrics, such as the number of packets and sessions, also represent the data size of Internet traffic. Using metrics other than bytes may be effective when detecting anomalies such as spam mail or port scans that do not influence bytes well. GAMPAL using multiple metrics may improve accuracy and versatility. When predicting flow size, the hour, the day of the week, and whether it is a business day are effective variables among the time feature, as described in Sect. 7.2.4. On the other hand, the effective variables for predicting other metrics, such as packets or session counts, may be different from those for bytes. In addition, as described in Sect. 8.4.1, further classification of PAs to which many flows belong and aggregation of some PAs to which a small number of flows belong should be considered. Resolving the bias of flow size in PAs may reduce FNs (False Negatives) and FPs (False Positives). Moreover,

as described in Section 8.5, there are several types of possible definitions of the predicted range and parameters of the filter. The definitions of the predicted range and parameters of the filter should be adjusted according to the environment in which the data are collected and the purpose. Furthermore, anomaly detection using multiple predicted ranges and filters with different parameters may be effective.

References

[1] C. Fraleigh, F. Tobagi, and C. Diot. Provisioning IP backbone networks to support latency sensitive traffic. In *Proc. of IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No.03CH37428)*, volume 1, pages 375–385 vol.1, 2003.

[2] M. Roesch. Snort - Lightweight Intrusion Detection for Networks. In *Proc. of the 13th USENIX Conference on System Administration*, LISA '99, page 229–238. USENIX Association, 1999.

[3] A. Bakshi and Y. B. Dujodwala. Securing Cloud from DDOS Attacks Using Intrusion Detection System in Virtual Machine. In *Proc. of 2010 Second International Conference on Communication Software and Networks (ICCSN)*, pages 260–264, 2010.

[4] S. VivinSandar and S. Shenai. Economic denial of sustainability (EDoS) in cloud services using HTTP and XML based DDoS attacks. *International Journal of Computer Applications*, 41:11–16, 03 2012.

[5] C. Mazzariello, R. Bifulco, and R. Canonico. Integrating a network IDS into an open source Cloud Computing environment. In *Proc. of 2010 Sixth International Conference on Information Assurance and Security*, pages 265–270, 2010.

[6] Z. Liang and R. Sekar. Fast and automated generation of attack signatures: a basis for building self-protecting servers. In *Proc. of the 12th ACM Conference on Computer and Communications Security*, CCS '05, page 213–222. Association for Computing Machinery, 2005.

[7] C. Kreibich and J. Crowcroft. Honeycomb: creating intrusion detection signatures using honeypots. *SIGCOMM Comput. Commun. Rev.*, 34(1):51–56, January 2004.

[8] R. Kumar and Sharma D. HyINT: Signature-Anomaly Intrusion Detection System. In *Proc. of 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pages 1–7, 2018.

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

[10] P. K. Deka., Y. Verma, A.B. Bhutto, E. Elmroth, and M. Bhuyan. Semi-Supervised Range-Based Anomaly Detection for Cloud Systems. *IEEE Transactions on Network and Service Management*, 20(2):1290–1304, 2023.

[11] T.A.Tang, L. Mhamdi, D. McLernon, S.A.R. Zaidi, M. Ghogho, and F. El Moussa. DeepIDS: Deep Learning Approach for Intrusion Detection in Software Defined Networking. *Electronics*, 9(9), 2020.

[12] 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 16th IEEE Annual Consumer Communications Networking Conference (CCNC) 2019*, pages 1–7, 2019.

[13] A. Zaheer, S. Tahir, M. F. Almufareh, and B. Hamid. A Hybrid Model for Botnet Detection using Machine Learning. In *Proc. of International Conference on Business Analytics for Technology and Security (ICBATS) 2023*, pages 1–8, 2023.

[14] 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 *2022 7th International Conference on Signal and Image Processing (ICSIP)*, pages 390–394, 2022.

[15] D. S. Terzi, R. Terzi, and S. Sagiroglu. Big data analytics for network anomaly detection from netflow data. In *Proc. of 2nd International Conference on Computer Science and Engineering (UBMK) 2017*, pages 592–597, 2017.

[16] L. Reuter, O. Jung, and J. Magin. Neural network based anomaly detection for SCADA systems. In *Proc. of 23rd Conference on Innovation*

in Clouds, Internet and Networks and Workshops (ICIN) 2020, pages 194–201, 2020.

[17] C. N. Modi and D. Patel. A novel hybrid-network intrusion detection system (H-NIDS) in cloud computing. In *2013 IEEE Symposium on Computational Intelligence in Cyber Security (CICS)*, pages 23–30, 2013.

[18] M. M. Shurman, R. M. Khrais, and A. A. Yateem. IoT Denial-of-Service Attack Detection and Prevention Using Hybrid IDS. In *Proc. of International Arab Conference on Information Technology (ACIT) 2019*, pages 252–254, 2019.

[19] E. Salma, F. Alberto, B. Abdullah, A. Saleh, and H. Francisco. On the combination of genetic fuzzy systems and pairwise learning for improving detection rates on Intrusion Detection Systems. *Expert Systems with Applications*, 42(1):193–202, 2015.

[20] N. T. Pham, E. Foo, S. Suriadi, H. Jeffrey, and H. F. M Lahza. Improving performance of intrusion detection system using ensemble methods and feature selection. In *Proc. of the Australasian Computer Science Week Multiconference*, ACSW '18, New York, NY, USA, 2018. Association for Computing Machinery.

[21] 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*, 77(5-6):437–454, June 2022.

[22] T. Wakui, F. Teraoka, and T. Kondo. GAMPALv2: An Anomaly Detection Mechanism for Internet Traffic by Predicting Flow Size Range from Time Features. *IEICE Transactions on Information and Systems*, E108-D(6), January 2025 (accepted).

[23] NSL-KDD dataset. `https://www.unb.ca/cic/datasets/nsl.html` (Last accessed 16 Feb 2024).

[24] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani. Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy. In *Proc. of the 2019 International Carnahan Conference on Security Technology (ICCST)*, pages 1–8, 2019.

[25] A. Lazaris and V. K. Prasanna. An LSTM Framework For Modeling Network Traffic. In *Proc. of 4th IFIP/IEEE Symposium on Integrated Network and Service Management*, pages 19–24, 2019.

[26] Z. Wang, X. Su, and Z. Ding. Long-Term Traffic Prediction Based on LSTM Encoder-Decoder Architecture. *IEEE Transactions on Intelligent Transportation Systems*, 22(10):6561–6571, 2021.

[27] B. Sun, W. Cheng, P. Goswami, and G. Bai. Short-term traffic forecasting using self-adjusting k-nearest neighbours. *IET Intelligent Transport Systems*, 12(1):41–48, 2018.

[28] T. Sebastian, A. Rodolfo, Z. Youduo, M. Guido, and P. Achille. Deep Learning-Based Traffic Prediction for Network Optimization. In *Proc.*

of 2018 20th International Conference on Transparent Optical Networks (ICTON), pages 1–4, 2018.

[29] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. M. Shroff. LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection. *ArXiv*, abs/1607.00148, 2016.

[30] N. Laptev and S. Amizadeh. S5 - A Labeled Anomaly Detection Dataset, version 1.0. `https://webscope.sandbox.yahoo.com/` (Last accessed 20 Dec 2024).

[31] S. Ahmad, A. Lavin, S. Purdy, and Z. Agha. Unsupervised real-time anomaly detection for streaming data. *Neurocomputing*, 262:134–147, 2017.

[32] K. Cho, B. V. Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation, 2014.

[33] K. Flanagan, E. Fallon, A. Awad, and P. Connolly. Self-configuring NetFlow anomaly detection using cluster density analysis. In *Proc. of 19th International Conference on Advanced Communication Technology (ICACT) 2017*, pages 421–427, 2017.

[34] T. Kohonen. The self-organizing map. *Proc. of the IEEE*, 78(9):1464–1480, 1990.

[35] S. García, M. Grill, J. Stiborek, and A. Zunino. An empirical comparison of botnet detection methods. *Computers & Security*, 45:100–123, 2014.

[36] I. Sharafaldin, A. Habibi Lashkari, and A.A. Ghorbani. Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSP*, 1:108–116, 2018.

[37] M. Lin, Q. Chen, and S. Yan. Network In Network, 2013.

[38] X. Ren, H. Gu, and W. Wei. Tree-RNN: Tree structural recurrent neural network for network traffic classification. *Expert Systems with Applications*, 167:114363, 2021.

[39] S. H. Dummer and S. Rath. A Retrospective on DDoS Trends in 2023 and Actionable Strategies for 2024, 2024. `https://www.akamai.com/blog/security/a-retrospective-on-ddos-trends-in-2023` (Last accessed 1 September 2024).

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

[41] nfdump. `http://nfdump.sourceforge.net` (Last accessed 28 May 2024).

[42] bgpdump. `https://github.com/RIPE-NCC/bgpdump`(Last accessed 28 May 2024).

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

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

[45] First Post of failure by TeamYouTube. https://x.com/TeamYouTube/status/1052373937839980544 (Last accessed 28 May 2024).

[46] Post of Failure Resolution by TeamYouTube. https://twitter.com/TeamYouTube/status/1052393799815589889 (Last accessed 28 May 2024).

[47] Regarding UDP Reflection/Amplification attacks using Apple Remote Management Service. https://www.sfc.itc.keio.ac.jp/ja/news_20190708_cns_filter_ard_sfc.html (Last accessed 28 May 2024).

[48] NETSCOUT. https://www.netscout.com/blog/asert/call-arms-apple-remote-management-service-udp (Last accessed 16 Feb 2024).

[49] 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*, 73, 2017.

[50] RandomForestRegressor from Scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html (Last accessed 28 May 2024).

[51] LSTM Layer from Keras. https://keras.io/api/layers/recurrent_layers/lstm/ (Last accessed 28 May 2024).

[52] SVR from Scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html (Last accessed 28 May 2024).

[53] datetime — Basic date and time types. https://docs.python.org/ja/3/library/datetime.html (Last accessed 28 May 2024).

[54] Scikit-learn. https://scikit-learn.org/stable/index.html (Last accessed 28 May 2024).

[55] Ministry of Internal Affairs and Communications. *Guidelines for Public Notification and Public Relations in Telecommunications Services Failures*. Ministry of Internal Affairs and Communications, 3 2023.

[56] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T. Liu. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Proc. of Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[57] LightGBM:Python-package Introduction. https://lightgbm.readthedocs.io/en/latest/pythonapi/lightgbm.LGBMRegressor.html (Last accessed 20 Dec 2024).

[58] LightGBM:Python-package Introduction. https://lightgbm.readthedocs.io/en/latest/Python-API.html (Last accessed 20 Dec 2024).

[59] T. Pedro. Unlocking the power of gradient-boosted trees using LightGBM. In *PyData London 2022*, 2022.

[60] MLJAR. LightGBM vs Random Forest. `https://mljar.com/machine-learning/lightgbm-vs-random-forest/` (Last accessed 20 Dec 2024).

[61] 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 the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (ACM SIGCOMM '06)*, pages 207–218, 2006.