IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 23 October 2023, accepted 15 November 2023, date of publication 17 November 2023,
date of current version 22 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3334212

RESEARCH ARTICLE

Network Anomaly Detection Through IP Traffic Analysis With Variable Granularity

SHOHEI KAMAMURA[1], (Member, IEEE), YUKI TAKEI[2], MASATO NISHIGUCHI[2], YUHEI HAYASHI[2], AND TAKAYUKI FUJIWARA[2]

[1]Faculty of Science and Technology, Seikei University, Tokyo 180-8633, Japan
[2]NTT Network Innovation Center, NTT Corporation, Tokyo 180-8585, Japan

Corresponding author: Shohei Kamamura (shohei-kamamura@st.seikei.ac.jp)

ABSTRACT A network anomaly detection method is proposed for large-scale, wide-range Internet Protocol (IP) networks. Because network behavior is projected onto communication traffic, anomaly detection can be achieved by properly analyzing the communication traffic flows. However, in wide-area IP networks, communication traffic flows are encapsulated by headers assigned by communication carriers and thus are observed as more macroscopic information. Therefore, accurately detecting the occurrence of anomalies in individual communication flows is difficult because the flow observation results obtained by flow measurement protocols such as IP Flow Information Export (IPFIX) are the result of superimposing various communication flows with different characteristics. In this study, we propose an anomaly-detection method based on time-series traffic flows. First, we decompose superimposed traffic flows into individual flows using our implemented system called the Fast xFlow Proxy, which can decompose traffic flows to a fine granularity. Our method detects anomalies in the decomposed flows based on a simple correlation analysis and dynamic threshold configuration. Our extensive simulation shows that, if we observe individual flows using the Fast xFlow Proxy, our method can detect anomalies caused by service failures with almost 100% accuracy. Our method can achieve an accuracy of approximately 80%–90% even in more difficult detection cases, such as small traffic fluctuations or noisy situations.

INDEX TERMS Anomaly detection, communication system traffic control, correlation, IP networks, time series analysis.

I. INTRODUCTION

To operate wide-area internet protocol (IP) networks as a social infrastructure in a sustainable and reliable manner, accurate measurement and analysis of the communication traffic over the network and detection of anomalies, such as failures or security attacks, are important. Communication traffic consists of individual flows. Generally, flows are identified using five identifiers called a 5-tuple: source IP address, destination IP address, source port number, destination port number, and protocol type. By referring to the flow information, we can observe the amount of specific applications such as web conferencing, video streaming, games, office traffic, and IoT services. These flows can be collected using flow measurement protocols called xFlow protocol such as NetFlow [1] and IPFIX [2].

Although the xFlow protocol can practically collect flow information over the Internet, it has limitations in large-scale wide-are IP networks: packet headers are encapsulated by additional headers assigned by communication carriers, and user traffic behavior is observed in a more macroscopic state. For example, encapsulation by the layer 2 tunneling protocol (L2TP) [3] is used for PPPoE user authentication, and multiple labels by segment routing using multiprotocol label switching (SR-MPLS) [4] are added as outer headers for virtual private network (VPN) users or flexible network control by traffic engineering [5]. Consequently, the xFlow protocol does not obtain detailed per-flow information but rather more macroscopic information from the outer headers.

Owing to the limitation of the xFlow protocol in large-scale, wide-area IP networks, flows are measured as

The associate editor coordinating the review of this manuscript and approving it for publication was Hosam El-Ocla.

macroinformation on which multiple services are superimposed. Although each service has different traffic characteristics depending on its usage pattern, these differences are not discernible trough this superimposition. As a result, accurately detecting communication anomalies for a specific flow from the measurement results becomes extremely difficult.

Herein, we propose a network anomaly detection method using the Fast xFlow Proxy [6], which has the capability of measuring communication flows at a fine-granular level to improve anomaly detection accuracy. The Fast xFlow Proxy can analyze complex outer headers of IP packets and perform statistical processing per service at ultrahigh speeds (100 Gbps), which can appropriately decompose macro-traffic information into individual flow behaviors. Our assumption is that if traffic is properly decomposed, we can detect anomalies quickly and accurately using a simple correlation value analysis. This is because of our empirical observation that the time-series pattern of traffic tends to fluctuate periodically while increasing monotonically unless an external factor, such as a failure, occurs. Therefore, we propose the anomaly-detection method by applying a lightweight correlation value analysis method to the decomposed communication flows.

This study makes two major contributions to the existing literature.

1. We demonstrate that, when macroscopic communication traffic information is decomposed into individual flows, anomalies can be detected clearly and numerically, even in an environment where packets are encapsulated.
2. We demonstrate the applicability of our proposed method to specific network anomalies such as service disruptions and distributed denial of service (DDoS) attacks. If we observe individual flows using the Fast xFlow Proxy, service failures, which cause communication disruptions, can be detected with almost 100% accuracy because the amount of change in the correlation values is large. However, traffic fluctuations, whose variations in correlation values are smaller than that of failures, tend to be more difficult to detect. Therefore, by dynamically adjusting the threshold value for anomaly detection, we show that our method can achieve an accuracy of approximately 80–90% in detecting small traffic fluctuations and anomalies in noisy situations.

The remainder of this paper is organized as follows: Section II discusses related works. Section III presents an overview of our Fast xFlow Proxy system and defines the problem statement. Section IV describes the proposed network anomaly-detection methods. Section V presents an evaluation of the effectiveness of the proposed method, and Section VI concludes the paper.

II. RELATED WORKS

As studies closely related to this research, we introduce studies on observing and classifying traffic [1], [2], [6], [9], [10], [11], [12], [13], [14], [15], anomaly detection based primarily on traffic analysis [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], and traffic generation [31], [32], [33], [34], [35], [36], [37], [38], [39]. The taxonomy of traffic measurements and time-series traffic analyses, which are particularly relevant to this study, is presented in Table 1.

A. TRAFFIC MEASUREMENT

In the analysis of networks, methods have been proposed for multimedia analysis at the application layer [7], as represented by web access, and for analyzing the relationship between citations of publications based on the graph theory [8]. The traffic addressed in this paper is more about lower-layer flows defined by port numbers in the transport layer and IP addresses in the IP layer.

Methods for traffic measurement can be classified into two categories: direct measurement of packets and a flow measurement-based approach that obtains meta-information of sampled packets at regular intervals. A typical example of the former is deep packet inspection (DPI) [9], which accumulates pcap format data as big data. A machine learning approach has been proposed to analyze the data obtained by DPI [10]. However, this approach is inefficient owing to the enormous amount of traffic in wide-area IP networks.

In contrast, a flow-measurement-based approach has been proposed for efficient traffic measurement [1], [2], [11]. Although various flow definitions exist, flows are identified using five identifiers called a 5-tuple: source IP address, destination IP address, source port number, destination port number, and protocol type. Netflow and IPFIX count packet appearances, whereas sFlow [11] extracts a specified number of bytes from the beginning of a packet. Each method forwards the collected data to a flow collector. Recently, efficient methods based on probabilistic data structures have been proposed, such as flow radar [12] using a Bloom filter, MV-sketch [13] using a count-min sketch structure, and HashFlow [14], which identifies elephant flows and focuses on their observations. By observing communication flows using these methods, network operators can properly assess service failures or apply traffic engineering [5] to their networks.

Although the flow-measurement-based approach is highly scalable, IP packets in a large-scale carrier network are encapsulated with various outer headers, such as L2TP [3] or MPLS labels [4]. Ref. [15] considered a limited protocol stack, such as IPv6 over IPv4, and the maximum throughput was approximately 10 Gbps. Subsequently, we proposed the Fast xFlow Proxy [6], which can deal with various protocol stacks and has 100 Gbps performance. As presented in Table 1, our analysis method is the first one based on the Fast xFlow Proxy. The key points of the Fast xFlow Proxy are described in Section III, and our analysis method is described in Section IV.

TABLE 1. Taxonomy of traffic measurement and time-series traffic analysis. The scope of our method is also clarified.

Classification		Existing Method	Technical Aspects and Key Contribution	Our Method
Traffic Measurement	direct measurement	DPI [9]	- Detailed information by direct packet measurement - enormous amount of traffic data	-
	flow measurement	Classical xFlow technologies (NetFlow [1], IPFIX [2], sFlow [11])	- Meta-information of sampled flows - Low processing load and high scalability - Low applicability in packet encapsulated environments	-
		probabilistic data structures [12][13][14]	- Meta-information of sampled flows - More efficient measurement than classical xFlow technology - Low applicability in packet encapsulated environments	-
		Fast xFlow Proxy [6]	- Meta-information of sampled flows - Highl applicability in packet encapsulated environments (large-scale carrier network)	✔
Time-series Traffic Analysis	with model prediction	ARIMA [19][20], LSTM [21], CNN[22]	- Prediction of complex time-series variations by advanced analysis, and difference analysis from actual measurements	-
	without model prediction	Signal Analysis [23][24], Data Mining [25][26], Statistical Analysis [27][28][29]	- Similarity analysis by simple and fast autoregression without prediction	✔

DPI: Deep Packet Inspection
ARIMA: Auto Regressive Integrated Moving Average
LSTM: Long Short-Term Memory, CNN: Convolutional Neural Network

B. TIME-SERIES DATA ANALYSIS

Composite and advanced analysis methods based on multiple indicators obtained from networks [16], [17], [18] have been studied as trends in network anomaly detection. For example, methods based on deep neural evolution networks accurately estimate fault locations from large alarm information [16], predict multi-index time series variations using deep learning to detect anomalies in optical networks [17], and develop a method for anomaly detection using autoencoders, which is unsupervised learning [18]. On the other hand, simpler methods have been proposed for anomaly detection by focusing on the behavior of communication traffic, which is time-series data [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30]. Our study adopted this simple approach because our empirical observations show that the time-series pattern of traffic tends to fluctuate periodically.

Forecasting-based approaches [19], [20] that model time-series data using autoregressive analysis, such as ARIMA, and detect anomalies by comparing the values forecasted from the model with observed values have been proposed for anomaly detection. In addition, machine learning-based approaches using long short-term memory (LSTM) [21] and convolutional neural network (CNN) [22] have been proposed for forecasting. These methods assume that the behavior of time-series data is complex and fluctuates irregularly, which differs from our assumption that the shape of the communication flow is periodic.

A simple approach to anomaly detection for time series data that does not include model prediction is the unsupervised analysis of univariate data [23], [24], [25], [26], [27], [28], [29]. These methods detect anomalies by creating a low-dimensional latent space of normal time-series data and comparing it with observed values. For example, FFT [23] and SR [24] are based on signal analysis, and normA-SJ [25] and SAND [26] are based on data mining. They define their unique distance and compare the length of the distance to detect anomalies. A typical statistical approach involves computing the autocorrelation function between multiple sequences [27] and comparing probability distributions [28], [29]. A well-structured survey paper on time-series data analysis is presented in [30].

Although the proposed method described in Section IV is similar to the distance comparison approach in that it compares subsequences, it can detect anomalies quickly and accurately using a simple correlation value analysis. This is because of our empirical observation that the time-series pattern of traffic tends to fluctuate periodically while increasing monotonically unless an external factor, such as a failure, occurs. Therefore, as shown in Table 1, our analysis method adopts a simple method without model predictions. Further details are provided in Section IV.

C. TRAFFIC GENERATION

Our analysis was conducted using the traffic generator described in the appendix. This section introduces the research trends in traffic generators. Numerous traffic generators have been implemented previously [31]. They are classified as replayers and narrowly defined generators, and this section introduces the major generators.

Traffic replayers generate traffic in packet units [32], [33]. TCPivo [32] aims to replay packets accurately and quickly by tracing cap data. Tcpreplay [33] enables packet rewriting and editing and more advanced replay functions such as netmaps [34].

The most well-known traffic generators are iperf [35], which can generate traffic at arbitrary throughputs, and D-ITG [36], which can generate customized traffic with various patterns and protocols. MoonGen [37] is a logic-scriptable traffic generator, and Swing [38] is a tracing-based traffic generator that generates more realistic traffic behavior by tracing logs. Recently, NeCSTGen [39] was proposed to simulate the unpredictable behavior of various types of communication traffic using deep learning.

Many traffic generators have been developed for analyzing microscopic behavior at the packet or flow level. However, for the analysis of time-series data, which are more macroscopic and wide-ranging, we should implement a traffic generator that satisfies these requirements. Please refer to the appendix for further details.

III. PRELIMINARIES AND PROBLEM STATEMENT

A. OVERVIEW OF FAST XFLOW PROXY

As previously discussed, in large-scale wide-area IP networks, a user's packet headers are encapsulated by additional headers, such as the L2TP header or SR-MPLS labels assigned by the communication carriers. Therefore, as shown in Fig. 1(a), flow measurement protocols such as NetFlow or IPFIX collect statistics for these outer headers. Because the outer-header information consists of the IP addresses of the internal devices of the carrier network, the behaviors of individual user applications are difficult to observe.

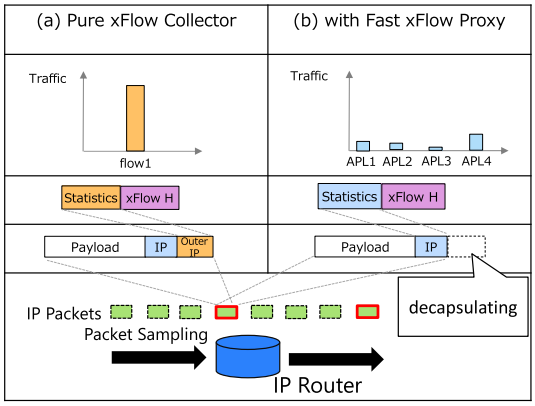


FIGURE 1. Traffic measurement using fast xFlow proxy.

In contrast, our proposed Fast xFlow Proxy enables the observation of flows with arbitrary granularity in large, wide-area IP networks. An example of a measurement using the Fast xFlow Proxy is illustrated in Fig. 1(b). A Fast xFlow Proxy was deployed between an IP router and commercial flow collectors [40]. If the user has given permission for detailed monitoring of their traffic, the Fast xFlow Proxy can be configured to remove outer IP headers in advance. After the Fast xFlow Proxy receives the flows, it forwards the inner header information to flow collectors. The flow collector performs statistical processing based on the inner header information observed by the IP router. The Fast xFlow Proxy is implemented using FPGA NICs [41], can be scaled out, and can process large volumes of flow information, up to 100 Gbps, in real time.

B. PROBLEM STATEMENT

Because IP traffic in large IP networks is observed as macroinformation on which various services are superimposed, accurate detection of anomalies from observation results has traditionally been difficult. However, as mentioned, the Fast xFlow Proxy can explore inside IP traffic and enable the observation of flows with arbitrary granularity, even in large-scale networks.

Therefore, the objective of this study is to establish a method for anomaly detection using a traffic measurement approach under the assumption that the Fast xFlow Proxy can observe communication traffic with arbitrary granularity. In addition, through evaluations, we clarify the guidelines for the appropriate granularity of flow observation for certain use cases.

IV. NETWORK ANOMALY DETECTION METHODS

A. NETWORK MODEL WITH FAST XFLOW PROXY

Fig. 2 presents an overview of the network anomaly-detection environment. The network is assumed to be a large-scale, wide-area IP network provided by a telecommunications carrier. In a wide-area IP network, wireless and mobile networks, such as 4G and 5G, and wired networks using optical fibers are provided as access networks for users. Traffic passing through these access networks flows into the core network through edge routers at the boundaries. The core network consists of multifunctional edge routers and high-capacity core routers connected to each other via optical devices such as reconfigurable optical add/drop multiplexers (ROADM) or optical cross-connects (OXC). Figure 3 illustrates the network topology model. A block is defined for each region, and each block consists of multiple edge routers and a core router. To establish nationwide communication across Japan, multiple blocks are defined, and a higher-level core router connects the blocks. The network consists of thousands of edge routers and hundreds of core routers. Flows are collected by the xFlow protocol only from core routers or including edge routers. In both cases, our Fast xFlow Proxy could collect the flows practically [6]. Through the core network, users can access various services, such as internet access or secure VPN connections.

In the network model, we assume that communication flows are observed at the edge routers using IPFIX. In this case, the observed communication flows are encapsulated as described above and can only be classified by geographic location, for example, "flows to IP routers connected to the Internet" or "flows to IP routers connected to VPNs." Therefore, we first remove the outer IP headers of the flows observed by IPFIX using the Fast xFlow Proxy and then send the flow information to the analysis server at a later stage.

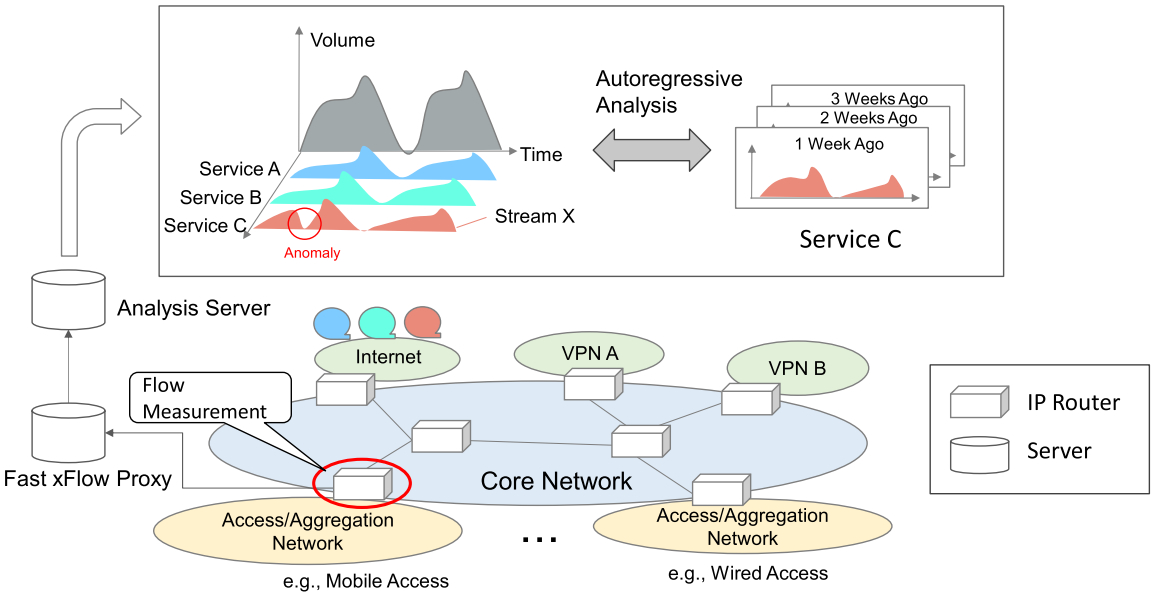


FIGURE 2. Overview of our network anomaly detection environment.

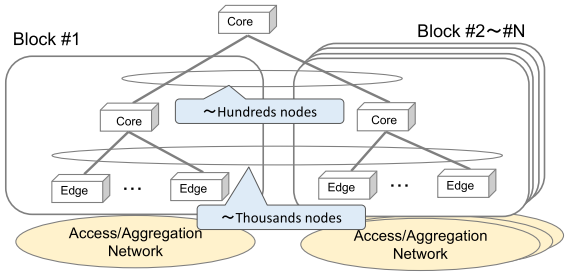


FIGURE 3. Network topology model. Only Layer 3 routers are shown, and each link consists of optical fibers and optical nodes such as OXC or ROADM.

The analysis server analyzes the detailed behavior of the traffic based on the 5-tuple of the inner IP header from the Fast xFlow Proxy output results. The analysis server records the communication flows as time-series data, representing the time variation in the volume of each flow. The specific anomaly-detection method on the analysis server is described in the next subsection.

B. ANOMALY-DETECTION METHOD

Based on our knowledge of network operations, in large-scale IP networks, even though the time-series pattern of traffic tends to increase monotonically on an annual basis, its daily variations tend to be periodic. Accordingly, we designed a simple anomaly-detection method. The proposed method detects the occurrence of an anomaly by calculating the correlation value between the shape of the target flow to be analyzed and its past shape. If no anomaly occurs, the correlation value is high; if an anomaly occurs, the correlation value is low because the correlation between the shapes is low.

First, the analysis data for the target flow were prepared as sequence X as follows:

$$X = (x_1, x_2, \cdots, x_n) \in R^n. \tag{1}$$

Each element x_i in sequence X represents the amount of traffic per unit time. For example, if one data point is observed every hour and the analysis is performed for 1 d, n is set to 24. Our method then produces sequence Y, which is T weeks before the target date X, as follows:

$$Y = \left(y_1^T, y_2^T, \cdots, y_n^T\right) \in R^n, T = 1, 2, \cdots T_m. \tag{2}$$

Our method computes the average $\bar{Y}$ of multiple observations to suppress the effect of the singularities of previous observations.

$$\bar{Y} = (\bar{y_1}, \bar{y_2}, \cdots, \bar{y_n}) \in R^n, \tag{3}$$

where $\bar{y_l}$ is as follows:

$$\bar{y_l} = \frac{1}{T_m} \sum_{k=1}^{T_m} y_l^k. \tag{4}$$

Because the dimensions of sequences X and $\bar{Y}$ are the same, the correlation value $\rho_{X,\bar{Y}}$ can be calculated using

$$\rho_{X,\bar{Y}} = \frac{\frac{1}{n}\sum_{t=1}^{n}(x_t - \bar{x})(y_t - \bar{y})}{\sigma_x \cdot \sigma_y}, \tag{5}$$

where $\bar{x}$ and $\bar{y}$ are the means of X and $\bar{Y}$, respectively, and σ_x and σ_y are their standard deviations, respectively. $\rho_{X,\bar{Y}}$ is a standardized value that is close to $+1$ when two sequences are correlated and close to zero when they are not correlated. If $\rho_{X,\bar{Y}}$ falls below a predetermined threshold value, our method determines that an anomaly has occurred.

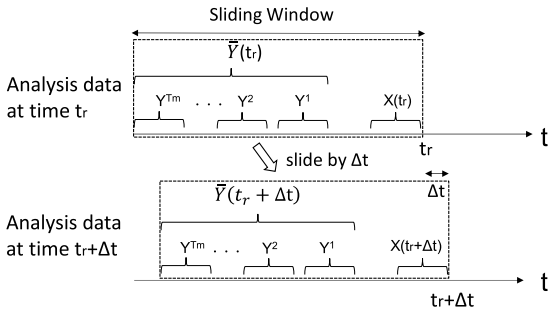


FIGURE 4. Quasi-real-time anomaly detection using sliding windows. $\mathbf{X}(t)$, $\mathbf{Y}(t)$, $\bar{\mathbf{Y}}(t)$ is based on (1), (2), (3) respectively.

Although our proposed anomaly-detection method requires sequences X and $\bar{Y}$, it can also be applied to quasi-real-time anomaly detection by employing a sliding-window mechanism. As shown in Fig. 4, sequences $X(t_r)$ and $\bar{Y}(t_r)$ are created using the abovementioned equations based on the latest traffic observation time t_r, and then $\rho_{X(t_r),\bar{Y}(t_r)}$ is computed. Next, sequences $X(t_r + \Delta t)$ and $\bar{Y}(t_r + \Delta t)$ are created at time $t_r + \Delta t$ after the observation time Δt has passed, and the correlation values $\rho_{X(t_r+\Delta t),\bar{Y}(t_r+\Delta t)}$ are computed. Thus, the correlation value is computed while sliding the window, which is the period to be analyzed, as the observation results are added. This enables quasi-real-time anomaly detection.

C. METHOD FOR DETERMINING THE THRESHOLD VALUE

In this section, we describe how the threshold value for determining an anomaly is evaluated, not only as a fixed value but also as a dynamically changing value. The dynamic threshold T_h is given by the following equation:

$$T_h = (1 - \alpha \cdot SR) \cdot \rho^{mean}, \tag{6}$$

where α is a scale factor, and ρ^{mean} is the average of the correlation values between normal periods.

Let SR represent the sampling rate obtained by the First xFlow Proxy: If F_o is the rate of the observed flow and F_e is the rate of the evented flow, where an external event such as failure is occurring, then SR is defined by the following equation:

$$SR = F_e/F_o. \tag{7}$$

For example, when $SR = 1$, the observed flow rate is equal to the flow rate of the external event occurrence, indicating that the flow for important users is monitored intensively using the Fast xFlow Proxy. Conversely, as the SR value decreases, it indicates a state in which multiple flows of the *masses* are efficiently monitored. When SR is close to one, T_h becomes smaller, and when it is close to zero, T_h increases. Our goal is to improve the anomaly-detection rate by adaptively changing T_h within a range not exceeding ρ^{mean}, in accordance with the monitoring granularity SR.

V. PERFORMANCE EVALUATION

A. AIMS AND CONDITIONS

We evaluated our method by computer simulations to clarify whether it could detect anomalies, even in an environment where packets are encapsulated. In addition, we analyzed its applicability to two anormal traffic fluctuation scenarios. The first is a decrease in traffic due to service failures on the Internet. The second is an increase in traffic due to bandwidth-consuming security attacks such as DDoS attacks.

The traffic was generated as a one-month time series from April 1, 2022, to April 30, 2022, using our traffic generator described in the Appendix. Time-series data were generated by superimposing 10 different communication flows with approximately the same traffic rate. In the simulation, an external event was intentionally generated on one service on a specific date and time that reduced the traffic rate to zero owing to a failure between 1 and 5 h (Tuesday, April 26) and increased the traffic rate between 10% and 100% owing to a security attack between 1 and 5 h (Monday, April 25).

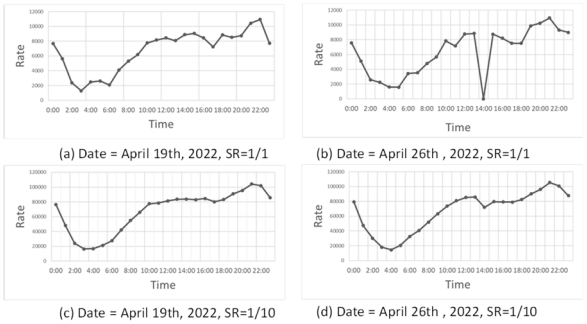


FIGURE 5. Example of generated flows. April 16, 2022 ((a) and (c)) represents normal conditions and April 26, 2022 ((b) and (d)) represents the date of the disruption. SR means the sampling rate. Noise ratio is set as 0.1.

Fig. 5 shows an example of the generated traffic. Fig. 5(a) and 5(b) show the daily traffic variation for the normal and 1-hour failure cases observed with SR = 1/1, whereas Fig. 5(c) and 5(d) show the traffic variation for the normal and 1-hour failure cases observed with SR = 1/10. In the case of SR = 1/1, shown in Fig. 5(b), the decrease in traffic is evident, whereas in the case of SR = 1/10 in Fig. 5(d), the anomaly is difficult to detect because it occurs only in a portion of the observed traffic. Using this time-series traffic data as input, the correlation was computed as an evaluation index using (5) in Section IV-B.

B. RESULTS FOR FAILURE EVENTS

Fig. 6 shows the correlation between the date of the failure event (Tuesday, April 26) and the average value on the same day of the week (Tuesday) for the previous three weeks. The X-axis represents the failure time, and the Y-axis represents the SR, which varies from 1/1 to 1/10. Darker-colored areas on the heat map indicate that the correlation values decreased significantly. Additionally, Figs. 6(a)–(c) indicate that random values were added as noise components

FIGURE 6. Correlation values when the sampling rate (SR) is changed at the failure event. Darker colored areas on the heat map indicate a significant decrease in correlation values. The correlation value was computed using (5).

FIGURE 7. Correlation values when the sampling rate (SR) is changed at the traffic increase event. Darker colored areas on the heat map indicate a significant decrease in correlation values. The correlation value was computed using (5).

at the rates of 1%, 5%, and 30%, respectively. This assumes that the shape of the communication flow is periodic but fluctuates periodically, and the fluctuations are small and irregular.

First, we focus on Fig. 6(a), which shows almost no noise components. As the SR value increases, the correlation value decreases. Increasing the SR value means that the failed flow is monitored more intensively, enabling easier numerical identification of external events such as service failures. However, even if the SR is not set as high as 1/1, a decrease in the correlation value can be visually confirmed in the range of 1/3 to 1/4. In addition, the correlation decreased as the failure time increased, even when the SR value was as low as 1/10. In other words, although the detection of failures is easy if the SR is set to a high value, anomalies can be detected even with a smaller SR, depending on the threshold value of the correlation. This also indicates that monitoring a single service intensively is not necessary and that multiple services can be efficiently monitored.

Next, we focus on the cases in Fig. 6(b) and (c), where the noise component is large. In the presence of a certain amount of noise, the correlation values tended to decrease as the SR value increased, independent of the effect of failure. For example, in the case of zero failure time and SR = 1/1, the correlation value for 10% noise was 0.9771 and that for 30% noise was 0.7389, indicating a decrease. This is because the number of observed flows decreased as the SR value increased, which increased the correlation value's susceptibility to noise components. In the case of a low SR, for example, the correlation value was 0.9829 when the failure time was zero, SR was 0.1, and noise was 30%, whereas the correlation value was 0.9573 when the failure time was 1 h under the same conditions. In other words, although the correlation value analysis was affected by the noise component, it tended to decrease as the failure time increased. Therefore, an anomaly can be detected by determining whether the correlation value decreases, compared to the normal condition.

C. RESULTS FOR TRAFFIC INCREASE EVENTS

Fig. 7 shows the correlation values between the date of the traffic increase event (Monday, April 25) and the average values for the same day of the week (Monday) for the previous three weeks. The X-axis shows the time of traffic increase, and the Y-axis shows the SR, which varies from 1/1 to 1/10. As in Fig. 6, the darker colors on the heatmap indicate that the correlation values decreased significantly. In addition, Fig. 7(a)–(c) indicate that traffic increased by 10%, 50%, and 100%, respectively. For example, a 100% increase in Fig. 7(c) indicates that the original traffic volume has doubled.

In contrast to Fig. 6, in the case of increased traffic, the correlation value does not decrease as much as it does in the case of failure. For example, in the case of a 10% traffic increase in Fig. 7(a), the decrease in the correlation values cannot be visually read from the heatmap. However, this

case is not without a trend either. For example, in the top row of the heatmap in Fig. 7(a), where the SR value is 1/1, the correlation value gradually decreases as the traffic time increases.

Next, in the case of 50% traffic in Fig. 7(b), the correlation value decreases significantly if the SR value is not less than 1/3 and the time increase is not less than 2 h. This trend is more pronounced in Fig. 7(c). In the case of a 100% increase in traffic, as shown in Fig. 7(c), the correlation value tended to decrease not only when the SR value was large but also when the SR value was 1/10. Although the decrease in the correlation value is evident when the SR is large, the correlation value tends to decrease even when the SR is as low as 1/10, as shown in the bottom row of the heatmap in Fig. 7(c).

TABLE 2. Analysis of anomaly detection based on accuracy indicators with moderate threshold (Th = 0.999).

		Accuracy	Precision	Recall	Specificity
Failure	noise ratio =1%	100%	100%	100%	100%
	noise ratio =10%	50%	50%	100%	0%
	noise ratio =30%	50%	50%	100%	0%
Traffic Increase	increase ratio =10%	58%	100%	16%	100%
	increase ratio =50%	95%	100%	90%	100%
	increase ratio =100%	100%	100%	100%	100%

TABLE 3. Analysis of anomaly detection based on accuracy indicators with sensitive threshold (Th = 0.9999).

		Accuracy	Precision	Recall	Specificity
Failure	noise ratio =1%	79%	70%	100%	57%
	noise ratio =10%	50%	50%	100%	0%
	noise ratio =30%	50%	50%	100%	0%
Traffic Increase	increase ratio =10%	71%	68%	78%	64%
	increase ratio =50%	83%	74%	100%	65%
	increase ratio =100%	81%	72%	100%	62%

D. RESULTS FOR THRESHOLD CONFIGURATION

Tables 2, 3, and 4 present the accuracy, precision, recall, and specificity for each condition when the threshold value for judging anomaly detection was changed. Each metric was defined using true positive (TP), true negative (TN), false positive (FP), and false negative (FN) as follows. Accuracy is the ratio of correct predictions to all predictions, formulated as (TP + TN)/(TP + TN + FP + FN). Precision is the ratio of outcomes predicted to be positive to those that are actually positive and is formulated as TP/(TP + FP). Recall is the ratio of outcomes that are positive to those that could be correctly predicted to be positive, formulated as TP/(TP +FN). Specificity is the ratio of outcomes predicted to be negative to those that are negative and is formulated as TN/(TN +FP).

In this evaluation, a positive value indicates that failures or traffic increases have occurred, whereas a negative value indicates that no traffic fluctuations have occurred. For evaluating the effect of noise, ten different data sets were prepared and measured under each of the conditions in Figs. 6 and 7. In the evaluation results that follow, the fixed thresholds (0.999 or 0.9999) are heuristically determined from the results of Figs. 6 and 7. The dynamic thresholds are calculated based on (6) defined in Section IV-C.

TABLE 4. Analysis of anomaly detection based on accuracy indicators with dynamic threshold. α is set to 0.01. The dynamic thresholds are calculated based on (6).

		Accuracy	Precision	Recall	Specificity
Failure	noise ratio =1%	100%	100%	100%	100%
	noise ratio =10%	90%	83%	100%	80%
	noise ratio =30%	79%	70%	100%	58%
Traffic Increase	increase ratio =10%	97%	100%	94%	100%
	increase ratio =50%	100%	100%	100%	100%
	increase ratio =100%	100%	100%	100%	100%

Table 2 shows the results for relatively loose and *moderate* threshold settings. Failures with small noise (e.g., 1%) and traffic increases with large changes (e.g., IR = 50% or 100%) can be detected with good accuracy. However, accuracy is low when noise is high or traffic changes are small. In the former case with high noise, the specificity is zero because all normal states are regarded as false positives, whereas in the latter case with low traffic fluctuation, many false negatives occur, except in areas with high SR, resulting in a small recall.

Table 3 lists the results for the strict and sensitive thresholds. Although the accuracy of the cases with low traffic change is higher (71%) than that in Table 2 (58%), the accuracy decreased for the cases with higher traffic increases (IR = 50% or 100%) and noise = 1%, which are areas with high accuracy in Table 2. These are the results of judging false positive predictions that could have been judged as true negatives owing to the strict threshold value. Moreover, the specificity cannot be increased when the noise is high (10%–30%); in other words, false-positive judgments cannot be suppressed.

Table 4 shows the results of the proposed dynamic threshold. While our proposed method inherits the good characteristics of the moderated threshold (0.999), excelling in the case of low noise (1%) and large traffic fluctuations, it can also achieve a high accuracy of 97% for low traffic fluctuations and a relatively high accuracy (79%–90%) even under high noise conditions. However, in the case of high noise, the value of ρ^{mean}, which is used as the reference, varies. This causes a false-positive misjudgment of the normal state as an anomaly. Consequently, the accuracy (specificity) tended to decrease. However, a relatively high accuracy of over 79% can be achieved even in a noisy environment (30%), and our observations suggest that this noise is approximately the maximum.

In summary, the data under analysis had the characteristic of decreasing correlation values with a slope corresponding to the SR, and a dynamic threshold value using (6) could follow the characteristic well. The evaluation results showed that,

TABLE 5. Difference of metrics between no-changed condition and changed condition. At 0%, there is no difference and therefore no anomaly detection is possible; as the percentage approaches 100%, detection becomes easier. The noise ratio is 0.010 and failure time is 5 h. Correlation is computed using (5), Cosine Similarity and FFT-based Similarity are computed using (8), and SBD Similarity are computed using (9).

		Correlation	Cosine Similarity	FFT-based Similarity	SBD Similarity
with Fast xFlow Proxy	SR=1/1	51%	0%	11%	27%
	SR=1/10	1%	0%	0%	7%
	SR=1/100	0%	0%	0%	0%
Conventional Approach	SR=1/1000	0%	0%	0%	0%
	SR=1/10000	0%	0%	0%	0%

SR: Sampling Rate FFT: Fast Fourier Transform SBD: Shape-Based Distance

even with a fixed threshold value, the overall accuracy was relatively good. This is because the analysis was performed at high resolution based on our Fast xFlow Proxy. In other words, an SR of 1/10, which is treated as a low resolution in this evaluation, is also a high resolution in the real world, and we believe that such good accuracy can only be achieved using the Fast xFlow Proxy.

E. COMPARISON WITH EXISTING METHODS

Finally, the effectiveness of the proposed method over existing methods is discussed based on Table 5. In Table 5, in addition to the previously mentioned correlation values, we use cosine similarity [42], Fourier Transform-based similarity [23], [24] and k-shape similarity [26] as evaluation metrics, both with and without our Fast xFlow Proxy implementation. The cosine similarity is given by the following equation using sequence X and $\bar{Y}$ defined in section IV-B.

$$S^{Cos}_{X,\bar{Y}} = \frac{\sum_{t=1}^{n} x_t \cdot y_t}{\sqrt{\sum_{t=1}^{n} x_t^2}\sqrt{\sum_{t=1}^{n} y_t^2}}. \tag{8}$$

The similarity by the shape-based distance (SBD) based on the k-shape method is calculated as follows:

$$S^{SBD}_{X,\bar{Y}} = 1 - \frac{\sum_{(i,j)\in P} D_{ij}}{\max\left(\left|x'_t\right|, \left|y'_t\right|\right)}, \tag{9}$$

where x'_t and y'_t represent the elements of Z-normalised sequence X and $\bar{Y}$, respectively. D_{ij} is the distance matrix by their Euclidean distance. P is the optimal pair with the minimum distance calculated by the Hungarian algorithm. In FFT-based similarity, the cosine similarity was calculated after transforming the data into frequency components using the FFT and removing the small noise components.

To compare these different metrics, the difference between the metric value when no change occurred and the metric value when the change occurred was calculated. For example, in SBD similarity, when the sequence in no-changed conditions is given as X_1 and $\bar{Y}_1$ and the sequence in changed conditions is given as X_2 and $\bar{Y}_2$, the difference is simply computed as follows:

$$Diff = |S^{SVD}_{X_1,\bar{Y}_1} - S^{SVD}_{X_2,\bar{Y}_2}|. \tag{10}$$

In other words, at 0%, there is no difference and therefore no anomaly detection is possible; as the percentage approaches 100%, detection becomes easier.

The most important result in Table 5 is that without our Fast xFlow Proxy implementation, no clear change can be detected using any analysis method owing to the low resolution of the observations. The results also suggest that the Fast xFlow Proxy may also be applicable to SBD while it has low applicability to Cosine Similarity or FFT-based algorithms. In this study, we adopted a correlation analysis as a simple and effective approach, but in future work, we would also like to analyze the applicability of different algorithms.

VI. CONCLUSION

In this study, we propose a network anomaly-detection method for large-scale wide-area IP networks using a powerful system called the Fast xFlow Proxy, which can measure IP traffic at a fine-grained resolution. The anomaly-detection method is based on an analysis of correlation values with past time-series patterns because the time-series patterns of individual flows tend to fluctuate periodically. In conclusion, the higher the resolution of the measurement using the Fast xFlow Proxy, the lower the correlation value. This facilitates anomaly detection. Although anomaly detection tends to be difficult under large noise or small traffic changes, its accuracy can be improved using the dynamic thresholding method.

Finally, we discuss the limitations of this study and its future directions. The first issue concerns the analysis methods: although this is the first study to utilize the Fast xFlow Proxy, the basic strategy is limited to a simple correlation analysis. As discussed in Section V-E, therefore, we will analyze the applicability of different algorithms such as SBD. Second issue is that the proposed method can detect the occurrence of failures, but not their location. In this study, we focus on service failures on the Internet. However, failures can be occurred not only on the Internet but also in the core

network or user-side equipment, such as campus networks, access networks, and base station areas. In future work, we will extend the anomaly-detection method to detect both the occurrence and location of failures. The key concept is the spatial analysis of multiple flows. This implies that multiple flows are monitored, and their correlation is analyzed to identify the fault location. The third issue is that this study did not use actual traffic, but used pseudo-traffic generated by our traffic generator, as shown in appendix A. Because the effectiveness of the proposed method has been demonstrated in this study, in future works, we will implement the proposed system and demonstrate its feasibility for anomaly detection in actual traffic.

APPENDIX

Although communication flows can be measured using flow measurement protocols such as NetFlow and IPFIX, they cannot be flexibly used because they must be managed in a manner that keeps personal information confidential. Therefore, we implemented a time-series traffic generator that generates periodic traffic patterns and simulates various external events.

The generator produces communication flows superimposed on a specific communication path as time-series data. Communication flow is defined between a region as the origin and a service as the destination. Regions and services have specific attributes. The region identifier, name, number of users, connected services, service utilization rate (service_rate), and status of the region can be defined. For a service, the service identifier, name, traffic model, related parameters, base rate (base_rate), and status of the service can be defined. For region i and service j, the communication flow $F(i, j)$ between (i, j) is defined by

$$F\,(i, j) = user(i) \times service\_rate(i) \times base\_rate(j). \quad (11)$$

For example, if a time-series traffic model of daily variation is given as a simple sine curve, the flow occurring at time t between (i, j) is defined by the following equation:

$$F\,(i, j, t) = \frac{F\,(i, j)}{2} sin\,(\omega t + \phi_0) + F_0. \quad (12)$$

For representing the 24-hour daily variation, the frequency ω is set to $\pi/12$, initial phase ϕ_0 is set to 2.8, and offset F_0 is set to $F(i, j)/2$. In addition to a sine curve, traffic models can be defined arbitrarily. In this study, we implemented a night peak model, which has a peak at around 10:00 p.m., similar to video distribution, and a day peak model, which has a peak during the daytime, similar to web conferencing. These models can be developed using approximate equations, for example, by performing polynomial fitting to the data obtained from actual measurements. An example of the night-peak model is shown in Fig. 5.

Because multiple flows are superimposed on a communication path, the flow $F_o(t)$ observed at time t in the IP router is given by the following equation, where the region and service sets are represented by R and S, respectively:

$$F_o\,(t) = \sum\nolimits_{i \in R, j \in S} F\,(i, j, t). \quad (13)$$

Using (13), F_o, which is the result of traffic observations over a set period, is generated. This generator can change the state of arbitrary regions and services or increase traffic between them over an arbitrary period of time. By generating these external events, the evented flow F_e can be defined.

REFERENCES

[1] B. Claise, *Cisco Systems NetFlow Services Export Version 9*, Standard RFC3954, IETF, Oct. 2004.

[2] B. Claise, B. Trammell, and P. Aitken, *Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information*, Standard RFC7011, IETF, Sep. 2013.

[3] W. T. A. Rubens, G. Pall, G. Zorn, and B. Palter, *Layer Two tunneling Protocol 'L2TP'*, Standard RFC2661, IETF, Aug., 1999.

[4] C. Filsfils, S. Previdi, L. Ginsberg, B. Decraene, S. Litkowski, and R. Shakir, *Segment Routing Architecture*, Standard RFC8402, IETF, Jul. 2018.

[5] S. Kamamura, "Dynamic traffic engineering considering service grade in integrated service network," *IEEE Access*, vol. 10, pp. 79021–79028, 2022, doi: 10.1109/ACCESS.2022.3194115.

[6] S. Kamamura, Y. Hayashi, Y. Miyoshi, T. Nishioka, C. Morioka, and H. Ohnishi, "Fast xFlow proxy: Exploring and visualizing deep inside of carrier traffic," *IEICE Trans. Commun.*, vol. 105, no. 5, pp. 512–521, 2022, doi: 10.1587/transcom.2021ebp3086.

[7] K. Abbas, M. K. Hasan, A. Abbasi, U. A. Mokhtar, A. Khan, S. N. H. S. Abdullah, S. Dong, S. Islam, D. Alboaneen, and F. R. A. Ahmed, "Predicting the future popularity of academic publications using deep learning by considering it as temporal citation networks," *IEEE Access*, vol. 11, pp. 83052–83068, 2023, doi: 10.1109/ACCESS.2023.3290906.

[8] A. Khan, J. P. Li, N. Ahmad, S. Sethi, A. U. Haq, S. H. Patel, and S. Rahim, "Predicting emerging trends on social media by modeling it as temporal bipartite networks," *IEEE Access*, vol. 8, pp. 39635–39646, 2020, doi: 10.1109/ACCESS.2020.2976134.

[9] T. Bujlow, V. Carela-Español, and P. Barlet-Ros, "Independent comparison of popular DPI tools for traffic classification," *Comput. Netw.*, vol. 76, pp. 75–89, Jan. 2015, doi: 10.1016/j.comnet.2014.11.001.

[10] M. Lotfollahi, M. J. Siavoshani, R. S. H. Zade, and M. Saberian, "Deep packet: A novel approach for encrypted traffic classification using deep learning," *Soft Comput.*, vol. 24, no. 3, pp. 1999–2012, Feb. 2020, doi: 10.1007/s00500-019-04030-2.

[11] *sFlow*. Accessed: Mar. 2023. [Online]. Available: http://www.sflow.org/

[12] Y. Li, R. Miao, C. Kim, and M. Yu, "FlowRadar: A better NetFlow for data centers," in *Proc. NSDI*, 2016, pp. 311–324.

[13] L. Tang, Q. Huang, and P. P. C. Lee, "A fast and compact invertible sketch for network-wide heavy flow detection," *IEEE/ACM Trans. Netw.*, vol. 28, no. 5, pp. 2350–2363, Oct. 2020, doi: 10.1109/TNET.2020.3011798.

[14] Z. Zhao, X. Shi, X. Yin, Z. Wang, and Q. Li, "HashFlow for better flow record collection," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2019, pp. 1416–1425, doi: 10.1109/ICDCS.2019.00141.

[15] M. Elich, M. Grégr, and P. Čeleda, "Monitoring of tunneled IPv6 traffic using packet decapsulation and IPFIX," in *Proc. Int. Workshop Traffic Monitor. Anal.*, 2011, pp. 64–71.

[16] H. Yang, X. Zhao, Q. Yao, A. Yu, J. Zhang, and Y. Ji, "Accurate fault location using deep neural evolution network in cloud data center interconnection," *IEEE Trans. Cloud Comput.*, vol. 10, no. 2, pp. 1402–1412, Apr. 2022, doi: 10.1109/TCC.2020.2974466.

[17] H. Yang, Y. Wan, Q. Yao, B. Bao, C. Li, Z. Sun, H. Wang, J. Zhang, and M. Cheriet, "Anomaly prediction with hybrid supervised/unsupervised deep learning for elastic optical networks: A multi-index correlative approach," *J. Lightw. Technol.*, vol. 40, no. 14, pp. 4502–4513, Jul. 15, 2022, doi: 10.1109/JLT.2022.3168594.

[18] Y. Ikeda, K. Ishibashi, Y. Nakano, K. Watanabe, and R. Kawahara, "Anomaly detection and interpretation using multimodal autoencoder and sparse optimization," 2018, *arXiv:1812.07136*.

[19] F. Wang, M. Li, Y. Mei, and W. Li, "Time series data mining: A case study with big data analytics approach," *IEEE Access*, vol. 8, pp. 14322–14328, 2020, doi: 10.1109/ACCESS.2020.2966553.

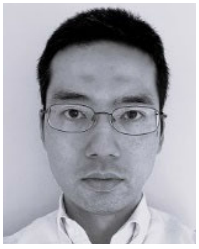
[20] S. Saha, A. Haque, and G. Sidebottom, "An empirical study on internet traffic prediction using statistical rolling model," in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*, May 2022, pp. 1058–1063, doi: 10.1109/IWCMC55113.2022.9825059.
[21] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long short term memory networks for anomaly detection in time series," in *Proc. Eur. Symp. Artif. Neural Netw., Comput. Intell. Mach. Learn. (ESANN)*, Apr. 2015, pp. 89–94.
[22] M. Munir, S. A. Siddiqui, A. Dengel, and S. Ahmed, "DeepAnT: A deep learning approach for unsupervised anomaly detection in time series," *IEEE Access*, vol. 7, pp. 1991–2005, 2019, doi: 10.1109/ACCESS.2018.2886457.
[23] F. Rasheed, P. Peng, R. Alhajj, and J. Rokne, "Fourier transform based spatial outlier mining," in *Proc. Int. Conf. Intell. Data Eng. Automated Learn. (IDEAL)*, 2009, pp. 317–324, doi: 10.1007/978-3-642-04394-9_39.
[24] H. Ren, B. Xu, Y. Wang, C. Yi, C. Huang, X. Kou, T. Xing, M. Yang, J. Tong, and Q. Zhang, "Time-series anomaly detection service at Microsoft," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 3009–3017, doi: 10.1145/3292500.3330680.
[25] P. Boniol, M. Linardi, F. Roncallo, T. Palpanas, M. Meftah, and E. Remy, "Unsupervised and scalable subsequence anomaly detection in large data series," *VLDB J.*, vol. 30, no. 6, pp. 909–931, Nov. 2021, doi: 10.1007/s00778-021-00655-8.
[26] P. Boniol, J. Paparrizos, T. Palpanas, and M. J. Franklin, "SAND: Streaming subsequence anomaly detection," *Proc. VLDB Endowment*, vol. 14, no. 10, pp. 1717–1729, Jun. 2021, doi: 10.14778/3467861.3467863.
[27] M. Toledano, I. Cohen, Y. Ben-Simhon, and I. Tadeski, "Real-time anomaly detection system for time series at scale," in *Proc. KDD, Workshop Anomaly Detection Finance*, 2018, pp. 56–65.
[28] J. Hochenbaum, O. S. Vallis, and A. Kejariwal, "Automatic anomaly detection in the cloud via statistical learning," 2017, *arXiv:1704.07706*.
[29] A. Siffer, P.-A. Fouque, A. Termier, and C. Largouet, "Anomaly detection in streams with extreme value theory," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 1067–1075, doi: 10.1145/3097983.3098144.
[30] S. Schmidl, P. Wenig, and T. Papenbrock, "Anomaly detection in time series: A comprehensive evaluation," *Proc. VLDB Endowment*, vol. 15, no. 9, pp. 1779–1797, May 2022, doi: 10.14778/3538598.3538602.
[31] O. A. Adeleke, N. Bastin, and D. Gurkan, "Network traffic generation: A survey and methodology," *ACM Comput. Surv.*, vol. 55, no. 2, pp. 1–23, Feb. 2023, doi: 10.1145/3488375.
[32] W.-C. Feng, A. Goel, A. Bezzaz, W.-C. Feng, and J. Walpole, "TCPivo: A high-performance packet replay engine," in *Proc. ACM SIGCOMM Workshop Models, Methods Tools Reproducible Netw. Res.* New York, NY, USA, Aug. 2003, pp. 57–64, doi: 10.1145/944773.944783.
[33] F. Klassen. (2013). *Tcpreplay: Pcap Editing and Replaying Utilities*. AppNeta. [Online]. Available: http://tcpreplay.appneta.com
[34] *Netmap—The Fast Packet I/O Framework*. Accessed: May 2023. [Online]. Available: http://info.iet.unipi.it/ luigi/netmap/
[35] *iPerf—The Ultimate Speed Test Tool for TCP, UDP and SCTP*. Accessed: May 2023. [Online]. Available: https://iperf.fr/
[36] S. Avallone, S. Guadagno, D. Emma, A. Pescape, and G. Ventre, "D-ITG distributed internet traffic generator," in *Proc. 1st Int. Conf. Quant. Eval. Syst.*, 2004, pp. 316–317, doi: 10.1109/qest.2004.1348045.
[37] P. Emmerich, S. Gallenmüller, D. Raumer, F. Wohlfart, and G. Carle, "MoonGen: A scriptable high-speed packet generator," in *Proc. Internet Meas. Conf.*, Oct. 2015, pp. 275–287, doi: 10.1145/2815675.2815692.
[38] K. V. Vishwanath and A. Vahdat, "Swing: Realistic and responsive network traffic generation," *IEEE/ACM Trans. Netw.*, vol. 17, no. 3, pp. 712–725, Jun. 2009, doi: 10.1109/tnet.2009.2020830.
[39] F. Meslet-Millet, S. Mouysset, and E. Chaput, "NeCSTGen: An approach for realistic network traffic generation using deep learning," in *Proc. IEEE Global Commun. Conf.*, Rio de Janeiro, Brazil, Dec. 2022, pp. 3108–3113, doi: 10.1109/GLOBECOM48099.2022.10000731.
[40] *Flowmon Collector*. Accessed: Apr. 2023. [Online]. Available: https://www.flowmon.com/en/products/appliances/netflow-collector
[41] *Intel FPGA Programmable Acceleration Card N3000 Data Sheet*. Accessed: May 2023. [Online]. Available: https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ds/ds-pac-n3000.pdf
[42] Y. Du, J. Wang, W. Feng, S. Pan, T. Qin, R. Xu, and C. Wang, "AdaRNN: Adaptive learning and forecasting of time series," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2021, pp. 402–411, doi: 10.1145/3459637.3482315.

SHOHEI KAMAMURA (Member, IEEE) received the B.E., M.E., and Ph.D. degrees from Waseda University, Tokyo, in 2004, 2006, and 2013, respectively. He joined the Nippon Telegraph and Telephone Corporation (NTT), in 2006. From 2017 to 2020, he was engaged in the commercial development of the SDN controller and speech recognition for web conferencing with NTT Communications. Since 2021, he has been an Associate Professor with the Department of Computer and Information Science, Seikei University. His current research interests include IP and optical network design, control, optimization, and network data mining. He is a member of the Institute of Electronics, Information, and Communication Engineers (IEICE), Japan. He received the Young Engineer Award from IEICE, in 2010; the ICM English Session Encouragement Award from IEICE, in 2010; the Best Paper Award of ICC from IEEE, in 2014; the TELECOM System Technology Award from the Telecommunication Advancement Foundation, in 2017; and the Distinguished Contributions Award from IEICE, in 2017.

YUKI TAKEI received the B.E. and M.E. degrees in electrical and electronic engineering from Waseda University, in 2014 and 2016, respectively. He joined the Nippon Telegraph and Telephone Corporation (NTT) in 2016. He is currently an Engineer with the Photonic Transport Control Systems Group, Photonic Transport Network Systems Project, Network Innovation Center. His current research interests include network visualization, hardware offloading, and network controlling.

MASATO NISHIGUCHI received the B.E. and M.E. degrees from the University of Tsukuba, Ibaraki, in 2016 and 2018, respectively. He joined the Nippon Telegraph and Telephone Corporation (NTT), in 2018. Currently, he is a Researcher with NTT. His current research interests include broadband network gateway (BNG) disaggregation and IP over optical network operations and management. He is a member of IEICE.

YUHEI HAYASHI received the B.E. and M.E. degrees from the Tokyo Institute of Technology, Tokyo, in 2012 and 2014, respectively. He joined the Nippon Telegraph and Telephone Corporation (NTT), in 2014. Currently, he is a Researcher with NTT. His current research interests include network security, network visualization, and optimization.



TAKAYUKI FUJIWARA received the M.E. degree from Osaka University, in 2006. He joined the Nippon Telegraph and Telephone Corporation (NTT), in 2006. His current research interests include solving network design and flow optimization with mathematical methods.