IEEE
Advancing Technology for Humanity

IEEE ComSoc
IEEE Communications Society

# Transaction on Advanced Communications Technology (TACT-ICACT2024)

http://www.icact.org

TACT-icact 2024

IEEE Catalog Number: CFP24561-ART
ISSN: 2288-0003

## Organizers

GIRI
Global IT Research Institute

## Sponsors

IEEE ComSoc
IEEE Communications Society

NIA NATIONAL INFORMATION SOCIETY AGENCY

ETRI

GWCVB
Gangwon Convention & Visitors Bureau

ITI
Information Technology Institute of Vietnam National University

KICS

IEEK ComSoc

Korean Institute of Information Scientists and Engineers

OSIA
Open Standards and Internet Association

Korea Institute of Information Security & Cryptology

# Table of Contents
# (Journal)

# Volume 12 Issue 1 January 2023

Page: 1475 - 1482

Title: An Adaptive User Scheduling Algorithm for 6G Massive MIMO Systems

Author : Prof. Robert Akl, Prof. Robin Chataut

Institute : Fitchburg State University

Country : USA

# Volume 12 Issue 2 March 2023

Page: 1483 - 1493

Title: A Blockchain based Security Assessment Framework

Author : Mr. NANDURI SATYANARAYANA

Institute : CDAC

Country : India

# Volume 12 Issue 3 May 2023

Page: 1494 - 1506

Title: A Horizontal Federated Learning Approach to IoT Malware Traffic Detection: An Empirical Evaluation with N-BaIoT Dataset

Author : Mr. Phuc Hao Do, Dr. Tran Duc Le, Prof. Vladimir Vishnevsky, Prof. Aleksandr Berezkin, Prof. Ruslan Kirichek

Institute : sut.ru

Country : Viet Nam

# Volume 12 Issue 4 July 2023

Page: 1507 - 1513

Title: A Deep learning Framework for Cultural Heritage Damage Detection for Preservation; Based on the case of Heunginjimun and Yeongnamnu in South Korea

Author : Dr. Sang-Yun Lee, Mr. Daekyeom Lee

Institute : ETRI

Country : Korea(South)

## Volume 12 Issue 5 September 2023

Page : 1514 - 1520

Title: A Study on Connectivity Evaluation Among Peer Groups in Pure P2P Networks

Author : Dr. Yutaka Naito, Dr. Takumi Uemura, Dr. Takashige Hoshiai

Institute : Sojo University

Country : Japan

## Volume 12 Issue 6 November 2023

Page : 1521 - 1527

Title: Quick Blocking Operation of IDS/SDN Cooperative Firewall Systems by Reducing Communication Overhead

Author : Mr. Akihiro Takai, Mr. Yusei Katsura, Prof. Nariyoshi Yamai, Dr. Rei Nakagawa, Dr. Vasaka Visoottiviseth,

Institute : Tokyo University of Agriculture and Technology

Country : Japan

Page : 1528 - 1540

Title: Automated Vulnerability Assessment Approach for Web API that Considers Requests and Responses

Author : Mr. Yuki Ishida, Prof. Masaki Hanada, Prof. Atsushi Waseda, Dr. Moo Wan Kim

Institute : Tokyo University of Information Sciences

Country : Japan

# An Adaptive User Scheduling Algorithm for 6G Massive MIMO Systems

Robin Chataut* and Robert Akl**
*School of Computing and Engineering, Quinnipiac University, USA
**Department of Computer Science, University of North Texas, USA
**robin.chataut@quinnipiac.edu, robert.akl@unt.edu**

***Abstract*—Massive MIMO (Multiple-Input Multiple-Output) is a promising wireless access technology that has emerged as a solution to the ever-increasing demand for network capacity. Massive MIMO is expected to play a crucial role in the deployment of 5G and upcoming 6G networks, enabling the realization of their full potential capacity. Despite the numerous benefits, user scheduling during downlink communication in Massive MIMO systems is a challenging task due to the large number of antenna terminals. In this paper, we propose a novel scheduling algorithm aimed at improving the area throughput, sum capacity, error performance, and ensuring fairness among all users. The proposed algorithm uses the average channel rate as the scheduling criteria, which is calculated from the channel state information obtained from the users during uplink transmission. To evaluate the performance of our proposed algorithm, we conducted simulations using Matlab. Our results demonstrate that our proposed channel rate-based scheduling algorithm is superior to conventional scheduling algorithms in terms of sumrate, throughput, and bit error performance while also ensuring fairness among all users. The proposed algorithm can address the challenge of user scheduling in Massive MIMO systems and contribute to the efficient deployment of 5G and 6G networks. The ability to improve system capacity, area throughput, and provide fairness in communication is of great importance in meeting the high demands of future wireless networks. Our approach could pave the way for further research in improving the performance of Massive MIMO systems, thereby advancing the potential of 5G and 6G networks.**

***Index Terms*—5G, 6G, Massive MIMO, User Scheduling**

## I. Introduction

THe demand for high data rates has skyrocketed due to the growing usage of mobile devices and the emergence of new applications that require high-speed data transfer. This has led to the development of next-generation wireless systems such as 5G, beyond 5G, and 6G networks, which are expected to provide high data rates, low latency, and better quality of service. Multiple-input Multiple-output (MIMO) technology has been a key factor in the development of previous generation wireless networks such as 3G and 4G, and it is expected to continue playing a critical role in future wireless systems. MIMO technology utilizes multiple antennas at both the transmitter and receiver to create multiple signal paths, which can be used to improve the robustness of the link against fading and interference [1]–[6].

One of the major benefits of MIMO technology is diversity gain, which refers to the improvement in signal quality due to the use of multiple signal paths. By leveraging the spatial dimension, MIMO technology can create multiple independent signal paths between the transmitter and receiver, which can help mitigate the effect of fading on the signal strength. This is particularly useful in environments with a high degree of multipath propagation, such as urban areas. Another key benefit of MIMO technology is multiplexing gain, which refers to the increase in data rate due to the use of multiple signal paths. By sending independent data streams on each signal path, MIMO technology can effectively increase the bandwidth of the link, resulting in higher data rates. This is particularly useful in applications that require high-speed data transfer, such as video streaming, online gaming, and virtual reality.

To cater to more users with better quality of service, the MIMO technique called massive MIMO plays a critical role. Massive MIMO employs hundreds of antennas at the base station, serving multiple users simultaneously. Massive MIMO is a wireless access technology operating below 6GHz, which plays a crucial role in current 5G and upcoming 6G networks by offering high spectral and energy efficiency with

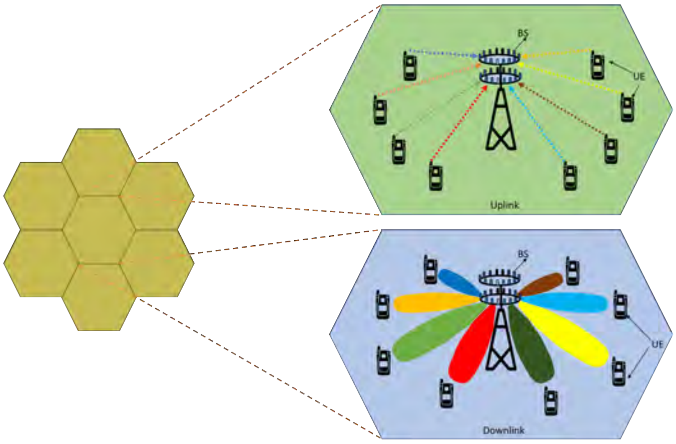


Fig. 1. Massive MIMO uplink and downlink system.

Manuscript submitted Dec 08, 2022. This is a follow-up of the invited journal to the accepted and presented paper entitled "An Adaptive User Scheduling Algorithm for 6G Massive MIMO Systems" at the 25th International Conference on Advanced Communication Technology (ICACT2023).

Robin Chataut is with the School of Computing and Engineering, Quinnipiac University, Hamden, CT, USA, 06518; (corresponding author, e-mail: Robin.Chataut@quinnipiac.edu;).

Robert Akl is with the Department of Computer Science, University of North Texas, Denton, Texas, USA 76201; e-mail: rakl@unt.edu

low latency [9]- [16]. It uses hundreds of antennas at the base station to serve tens of users simultaneously, providing high multiplexing and diversity gains while mitigating fading effects. Massive MIMO is essential to support the increasing demand for high data rates driven by applications such as blockchain, cyber-security, Smart Vehicles, the Internet of Things, augmented reality, virtual reality, and extended reality. The technique uses beamforming to direct signals towards users during the downlink, and the narrower beams resulting from more antennas improve spatial focus. Figure 1 shows a typical massive MIMO system where uplink pilot signals are transmitted by users towards the base station during uplink communication, and the downlink communication uses beamforming to direct signals towards the users. As the number of antennas increases, the beams become narrower, resulting in better spatial focus on users [17], [18].

However, with hundreds of antenna terminals, user scheduling during downlink communication is one of the major challenges in massive MIMO system deployment. A suitable user scheduling method during the downlink is necessary to enhance the throughput of massive MIMO systems when the number of active users is greater than the number of base station antenna terminals. Scheduling users with better channel conditions can improve the total area throughput. However, maintaining an adequate fairness level is equally important to ensure timely scheduling for users with weaker channel conditions. Considerable research has been conducted to develop optimal user scheduling algorithms. Greedy algorithms have been discussed in [19]- [21], which provide better fairness performance but fail to achieve optimal throughput. Traditional algorithms, Round Robin (RR), and Proportional Fair (PF) are better in terms of fairness but do not achieve optimal fairness. Linear methods like Zero Forcing (ZF) and Minimum Mean Square Error (MMSE) have been explored in [22]-[23]. The authors in [24]–[26], [28]–[31] have investigated user scheduling methods for downlink MIMO systems, but optimal performance in terms of both throughput and fairness has not been achieved. In this paper, we propose an adaptive user scheduling algorithm based on channel rate to provide the user with optimal throughput and ensure fairness among all the users.

### A. Contribution of the Paper

1) The user scheduling issue during the downlink massive MIMO system is investigated
2) An adaptive user scheduling scheme based on instantaneous channel rate is proposed
3) The sum rate, per-user throughput, and error performance of the proposed algorithm are accessed and compared with traditional scheduling algorithms
4) We evaluated the fairness index of the proposed algorithm. We have used Jain's fairness index to compute the fairness index.
5) The results obtained from the Matlab simulations show that the proposed algorithm is fair and performs better than the traditional user scheduling algorithm in terms of sumrate, per-user throughput, and error performance.

### B. Outline

The remainder of the paper is structured as follows: Section II defines the downlink system model for massive MIMO with $M$ antennas and $N$ users. The proposed adaptive algorithm is described in III. The simulation steps, required parameters, and algorithm analysis are presented in IV. Finally, V concludes the paper by encapsulating the major concepts of the paper.

### C. Notations

In this paper, there are specific notations and terminologies used to represent various mathematical concepts. Column vectors are denoted by lower-case letters, while matrices are denoted by upper-case letters. The inverse of a matrix is denoted by $(.)^{-1}$, and the transpose is represented by $(.)'$. The hermitian transpose is denoted by $(.)^{H}$. The circular symmetric complex Gaussian distribution with zero mean and co-variance $V$ is represented by $\mathcal{CN}(0, V)$. The space of $M$-element complex vectors is denoted by $\mathbb{C}^{M}$, where $M$ is a positive integer. The $M \times M$ identity matrix is represented by $I_M$, which is a square matrix with ones on the diagonal and zeros elsewhere.

## II. SYSTEM MODEL

In massive MIMO, the BS is equipped with numerous antennas, typically numbering in the hundreds or thousands. Downlink is the data transmission from the base station to the user equipment, such as mobile phones or laptops. The fundamental concept behind massive MIMO is to exploit the large number of antennas at the BS to concurrently communicate with multiple users utilizing the same time-frequency resource. This is accomplished by spatially combining signals from the BS's antennas to create unique signal combinations for each user. Downlink massive MIMO systems capitalize on the numerous antennas at the BS to enhance the quality and capacity of the wireless link to the users. Beamforming methods are employed by the BS to direct the transmitted signal towards each user, increasing the signal-to-noise ratio (SNR) and reducing interference from other users.

A massive MIMO downlink system is considered with M base station antenna terminals and N users. In the course of the downlink communication, the base station will send an independent and autonomous signal to each active user. If $U$ users are waiting for their turn to be scheduled, the base station selects $S$ users $(S <= U)$ according to the scheduling algorithm. The base station will apply a precoder before sending the downlink signal towards the user. The primary objective of precoding is to optimize the wireless channel between the base station and the users by modifying the phase and amplitude of the transmitted signal. The purpose of this modification is to reduce interference and enhance the quality of the signal received by the user. Precoding is a process that involves the application of a matrix operation to the data signals transmitted from the base station antennas. This operation is intended to decrease the interference between users and improve the signal-to-noise ratio (SNR) at the receiver's end. Several precoding algorithms are used in Massive MIMO systems, including zero-forcing (ZF) precoding and minimum

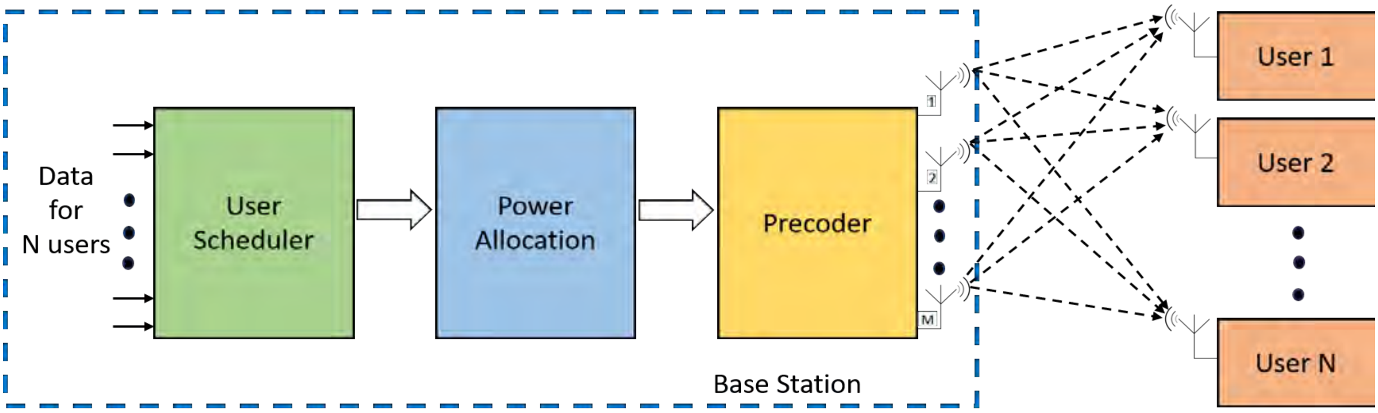


Fig. 2. System Model with $M$ base station antenna serving $N$ users.

mean squared error (MMSE) precoding. The implementation of precoding during user scheduling can lead to increased data rates, better spectral efficiency, and improved overall system performance in terms of signal quality and interference reduction. Therefore, precoding is a critical component in the design and optimization of Massive MIMO systems. The signal received by user $i$ can be represented as:

$$y_i = Hx_i + n_i \tag{1}$$

Where,

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ . \\ . \\ . \\ x_i \end{bmatrix} and \quad H_i = \begin{bmatrix} h^i_{11} & h^i_{12} & . & h^i_{1N} \\ h^i_{21} & h^i_{22} & . & h^i_{2N} \\ . & . & . & . \\ . & . & . & . \\ h^i_{N1} & h^i_{N2} & . & h^i_{NM} \end{bmatrix}$$

$y_i$ is the signal received by the $i_{th}$ user, and $x_i$ is the signal sent towards the user from the base station $i$. $H \in \mathbb{C}^{N \times M}$is the channel vector between the user terminals and the base station antenna terminals, where elements of H are independent and identically distributed. $n_i$ is the added white Gaussian noise at the $i_{th}$.

We do the precoding before scheduling the user to minimize multi-user interference. We get the matrix for precoding by stacking the beamforming vectors and user signals.

$$y_i = HWx_i + n_i \tag{2}$$

Where,

$$W = \begin{bmatrix} p_1 & p_2 & . & . & . & p_j \end{bmatrix}$$

$W \in \mathbb{C}$ is the precoding matrix, which contains a set of precoders. $p_j$ is the vector used for precoding the $j_{th}$ user. For our simulations, we have applied two simple linear precoders, ZF and MMSE [34]:

$$W_{ZF} = H^H(HH^H)^-1 \tag{3}$$

$$W_{MMSE} = H^H(HH^H + \sigma^2 I)^-1 \tag{4}$$

We compute the sumrate by considering the uniform power allocation among each user as [32]:

$$Sumrate = \sum_{i=1}^{N} \log_2 \left( 1 + \frac{|b_i h_i|^2}{1 + \sum_{j=1, j \neq i}^{N} |b_i h_j|^2} \right) \tag{5}$$

where, $b_k$ is the $k_{th}$ row of precoding matrix B and $h_k$ is the $k_{th}$ row of the channel matrix H.

## III. PROPOSED ALGORITHM FOR DOWNLINK USER SCHEDULING

The proposed algorithm is summarized in 1. We initialize the active users set $U$, including N active users. The set of selected users is $S$, which is null initially as non of the users are scheduled. Then we calculate the instantaneous channel rate for each user:

$$C_j = log_2 \left( 1 + \sqrt{\sum_{j=1}^{N} |h_j|^2} \right) \tag{6}$$

The mean channel rate is computed based on the active users waiting to be scheduled. The calculated mean channel rate will also be the selection criteria for the proposed algorithm.

$$\bar{C} = \frac{\sum C_j}{N} \tag{7}$$

The user with an instantaneous channel rate closest to the mean channel rate is selected first. Once the selected user is scheduled, we update the set containing the remaining active and selected users.

$$\pi(j) = argmin|||A_j| \tag{8}$$

$$S = S\,U\,\pi(j) \tag{9}$$

$$U = U - S \tag{10}$$

The process of user selection is repeated until all the active users are scheduled. Then, the mean channel rate is re-evaluated for the next set of active users.

$$U \neq \{\phi\} \tag{11}$$

**Algorithm 1** Proposed Algorithm for Massive MIMO Downlink Scheduling

```
Initialization:
1. $U = \{1, 2, 3, 4, ....N\}$
2. $S = \{\phi\}$
3. $j = 0$
Channel Rate Calculation:
4. $C_j = log_2\left(1 + \sqrt{\sum_{j=1}^{N} |h_j|^2}\right)$
5. $\bar{C} = \frac{\sum C_j}{N}$

Selection Criteria:
6. $A_j = |C_j - \bar{C}|$

Algorithm iteration:
do
    7. $\pi(j) = argmin|||A_j|$
    8. $S = S \, U \, \pi(j)$
    9. $U = U - S$
    10. $i = i + 1$
While $U \neq \{\phi\}$
```

## IV. SIMULATION RESULTS AND ANALYSIS

In this section, we analyze the results obtained from the Matlab simulations. For simulations, we set up a massive MIMO base station with many antenna terminals (16 to 512). We assume that all the antenna terminals are communicating with 128 single active users simultaneously. We have considered various antenna configurations with different modulation techniques (QPSK, 16QAM, 16QAM) for conducting the simulations. The system's bandwidth is set to 20 MHz, whereas a carrier frequency of 2.5 GHz is used. A perfect channel state information (CSI) is assumed between the user and the base station, and the Rayleigh fading channel model is used for simulations. We have compared our proposed algorithm with traditional schedulers like Proportional Fair (PF) and Round Robin (RR) algorithms for analysis. In addition, we have used ZF and MMSE precoding to reduce the effect of multi-user interference and to simplify the processing required at the receiver. The simulation parameters used are shown in I.

TABLE I
SIMULATION PARAMETERS

| Parameter | Value |
|---|---|
| Base Station Antenna Terminal | 16 to 512 |
| Number of Users | 128 |
| Carrier Frequency | 2.5 GHz |
| Bandwidth | 20 MHz |
| Coherence Internal | 200 Symbols |
| Channel Model | Uncorrelated Rayleigh Fading |
| Signal Variance | 2 |
| SNR | 0 dB - 25dB |
| Modulation | QPSK, 16QAM, 64QAM |

Fig. 3 depicts the error performance of the proposed algorithm with 16 users, 16 base station antenna terminals, 16QAM modulation, and MMSE precoding. The proposed algorithm exhibits better BER performance than the traditional algorithm across the entire range of user SNR in the simulation. For instance, at a BER of $10^{-2}$, the proposed algorithm achieves a 6dB gain over the RR algorithm and a 4dB gain over the PF algorithm. Similarly, conducting the same experiment with 16 users, 16 base station antenna terminals, and MMSE precoding, but with 64QAM modulation, results in degraded error performance for all algorithms, as shown in Fig. 4. At BER $10^{-1}$, the proposed algorithm achieves almost 3 dB gain over the PF algorithm and 4dB gain over the RR algorithm. Nonetheless, the per-user throughput increases for all algorithms with higher modulation order. Additionally, the simulation with comparable parameters and QPSK modulation, depicted in Fig. 5, results in improved error performance for all algorithms. This improvement stems from QPSK being less susceptible to interference and noise in comparison to higher modulation orders such as 16QAM and 64QAM used in our experiments.

Fig.6 illustrates the simulation outcomes when 16 users,

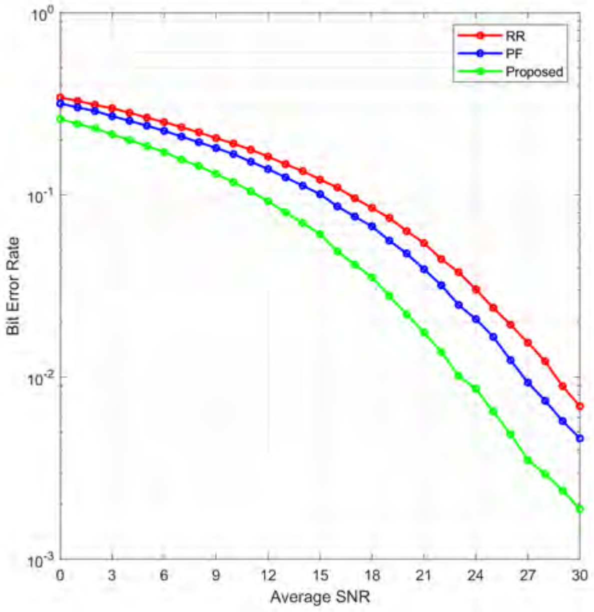


Fig. 3. BER vs. SNR performance with 16 users, 16 base station antennas, 16QAM modulation, and MMSE precoding.

16 base station antenna terminals, and 16QAM modulation are employed, but with Zero Forcing (ZF) precoding. The performance trend follows the same pattern as the previous experiment; however, the overall performance of all algorithms has decreased. Specifically, at a BER of $10^{-1}$, the proposed algorithm outperforms the RR algorithm by 5 dB and PF by 3.5 dB, underscoring the superiority of the proposed algorithm's BER performance compared to conventional algorithms. In Fig.7, QPSK modulation is employed, and all algorithms demonstrate enhanced error performance. This improvement can be attributed to the lower susceptibility of lower modulation orders, such as QPSK, to noise and interference. In contrast, higher modulation orders, such as 16QAM, are more susceptible to noise and interference, resulting in degraded error performance.

Fig.8 illustrates the analysis of the sumrate performance of the proposed algorithm. This simulation involved 16 base

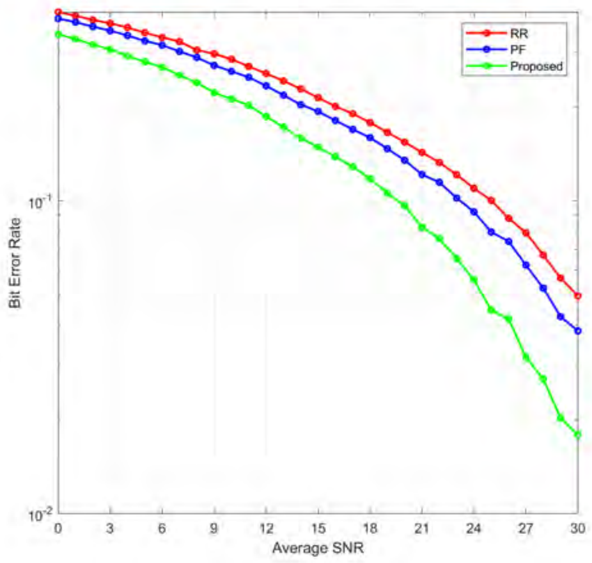


Fig. 4. BER vs. SNR performance with 16 users, 16 base station antennas, 64QAM modulation, and MMSE precoding.

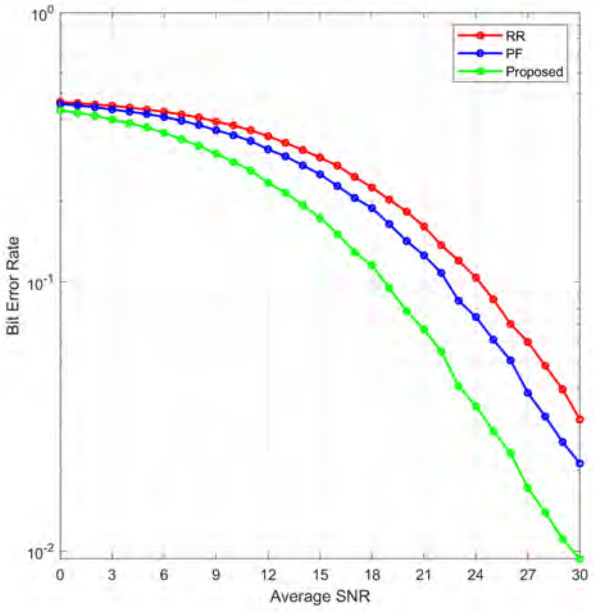


Fig. 6. BER vs. SNR performance with 16 users, 16 base station antennas, 16QAM modulation, and ZF precoding.

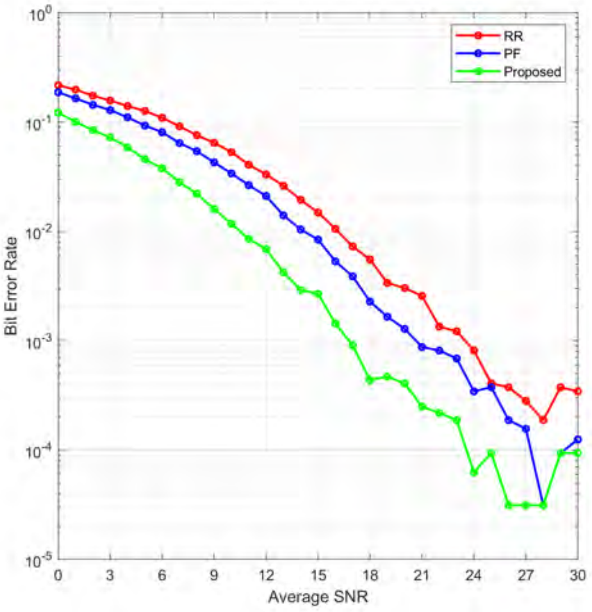


Fig. 5. BER vs. SNR performance with 16 users, 16 base station antennas, QPSK modulation, and MMSE precoding.

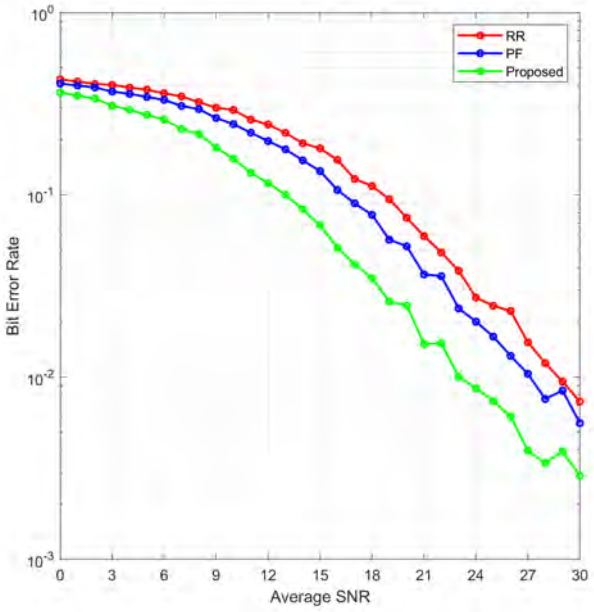


Fig. 7. BER vs. SNR performance with 16 users, 16 base station antennas, QPSK modulation, and ZF precoding.

ISSN: 2288-0003 (Online)

station antenna terminals communicating with 16 users using 16QAM modulation and MMSE precoding. The simulation results indicate that the proposed algorithm outperforms the traditional algorithms. For instance, at an SNR of 21dB, the proposed algorithm achieves a sum rate of 60 bits/s/Hz, whereas the PF algorithm attains a sum rate of 43 bits/s/Hz, and the RR algorithm exhibits the poorest performance, with a sum rate of 38 bits/s/Hz. The high sum rate is primarily attributed to the increased number of antenna terminals. Nevertheless, as the number of active users in a cell grows, the sum rate will eventually reach a saturation point.

We conducted a similar experiment using ZF precoding, and the sumrate performance was comparable to that of MMSE precoding, as demonstrated in Fig.9. With ZF precoding at an SNR of 21dB, the proposed algorithm attained a sumrate of 60 bits/s/Hz, while the PF and RR algorithms recorded a sum rate of 42 bits/s/Hz and 37 bits/s/Hz, respectively.

We then considered the performance of our proposed algorithm with several modulation techniques. This simulation was administered with 16 base station antenna terminals communicating with 16 users using 16QAM modulation and MMSE precoding. As shown Fig.10, QPSK exhibited the best error performance across a range of SNRs, while 64QAM displayed the best performance due to its ability to transmit more data per symbol. However, higher modulation orders are more susceptible to noise and interference, leading to higher error rates. Therefore, the optimal modulation order depends on the application and the user's requirements. Furthermore, we performed a simulation with ZF precoding using 16 base station antenna terminals communicating with 16 users via 16QAM modulation, as shown in Fig.11. We observed that the performance was nearly identical to that of MMSE precoding.

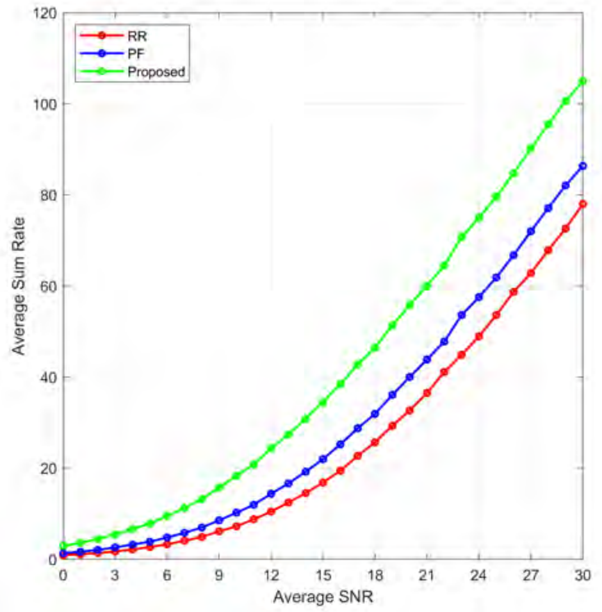


Fig. 8. Sumrate vs. BER performance with 16 users, 16 base station antennas, 16QAM modulation, and MMSE precoding.

We evaluated the proposed algorithm's average throughput per user performance. This simulation was administered with 16 base station antenna terminals communicating with 16 users using 16QAM modulation and MMSE precoding. As shown in Fig. 12, the average per-user throughput for the proposed algorithm was best among the compared algorithms. Our algorithm achieved a per-user throughput of 3.14 Mbps, whereas, for RR and PF algorithms, it was found to be 2.33 Mbps and 2.53 Mbps, respectively.

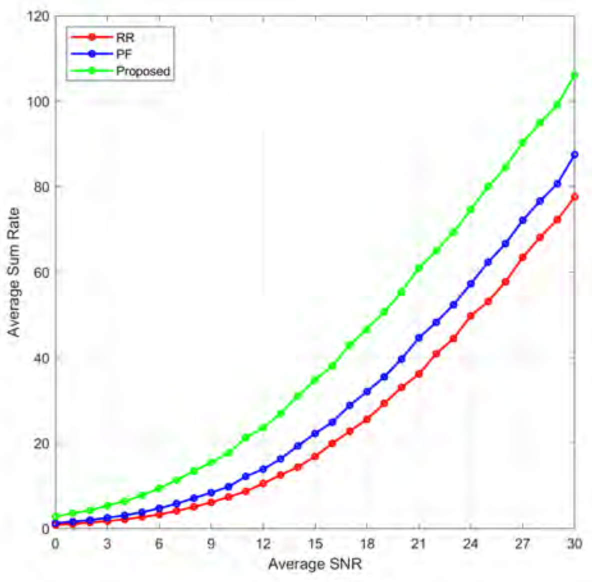


Fig. 9. Sumrate vs. BER performance with 16 users, 16 base station antennas, 16QAM modulation, and ZF precoding.

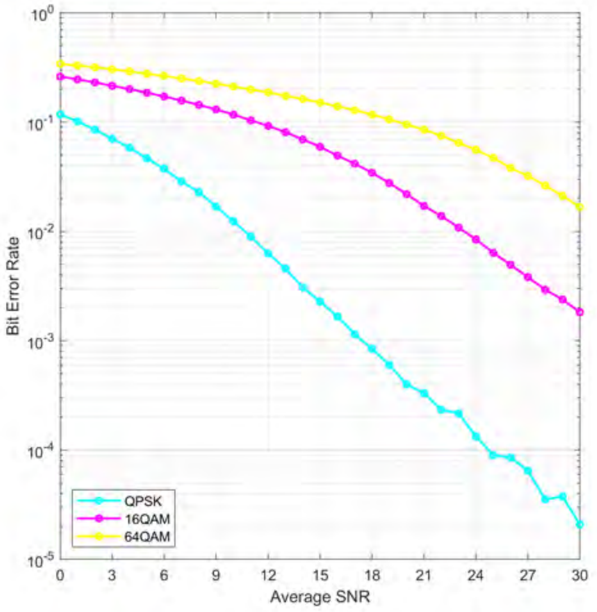


Fig. 10. BER performance of the proposed algorithm with several modulation schemes with 16 users, 16 base station antennas, and MMSE precoding

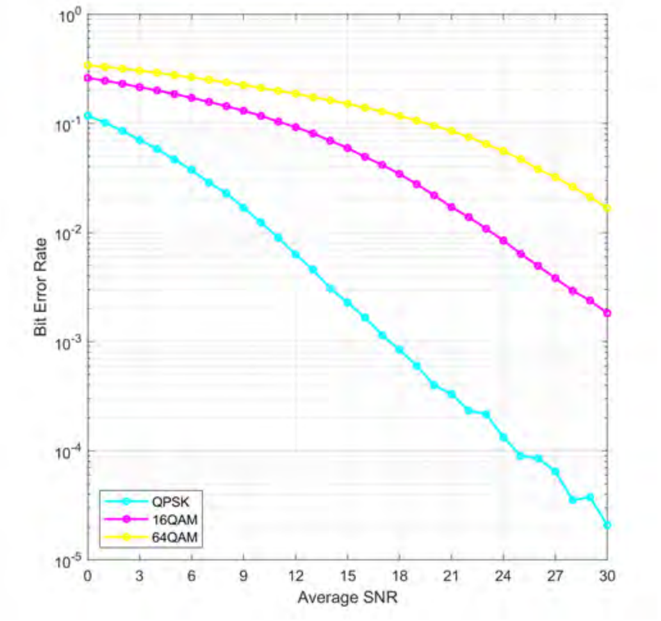


Fig. 11. BER performance of the proposed algorithm with several modulation schemes with 16 users, 16 base station antennas, and ZF precoding

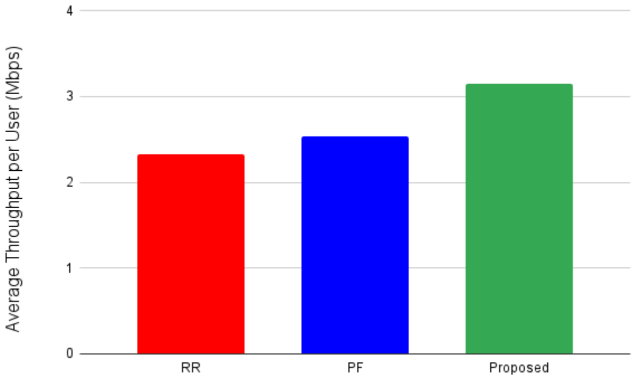


Fig. 12. Average throughput per user with 16 users, 16 base station antennas, 16QAM modulation, and MMSE precoding.

We use Jain's fairness index to evaluate the performance of the proposed algorithm. Jain's fairness index is a widely-used metric for assessing fairness in the distribution of limited resources, especially in the context of networking and telecommunications. This metric is particularly useful when multiple users or applications are competing for a finite amount of resources like CPU time or wireless bandwidth. It offers an objective way of quantifying the degree of fairness in resource allocation and comparing different allocation schemes. The fairness of resource allocation is critical in networking to prevent congestion, service degradation, or even network failure. Jain's fairness index allows for the comparison of resource allocation schemes by measuring how equitably resources are distributed among users or applications. An index value close to 1 suggests that resources are being allocated fairly to all users or applications, whereas a value closer to 0 indicates an unfair distribution, with some users or applications receiving more than their fair share of resources.

We measured Jain's fairness index for all the algorithms [33].

$$\mathcal{F}(X) = \frac{\left(\sum_{i=1}^{N} x_i\right)^2}{\sum_{i=1}^{N} x_i^2} \tag{12}$$

Where $\mathcal{F}$ is the fairness index whose values are between 0 and 11, and $x_j$ is throughput for $i$th user. As shown in II, simulation results show that the fairness provided by the proposed algorithm is similar to that of the traditional algorithms.

TABLE II
FAIRNESS INDEX COMPARISON

| Scheduling Algorithm | Fairness Index |
|---|---|
| Round Robin | 0.973 |
| Proportional Fair | 0.983 |
| Proposed | 0.999 |

## V. CONCLUSION

In conclusion, this paper addressed the issue of user scheduling during downlink signaling in a massive MIMO system. The proposed algorithm takes into account the instantaneous channel rate, which enables it to adaptively schedule users based on their current channel conditions. This results in a significant improvement in the sum rate and per-user throughput, as well as providing better error performance and fairness among all users. The simulation results also showed that the performance of the proposed algorithm varied with different modulation techniques. Specifically, 64QAM provided the best data rate, while QPSK provided the best error rate. This indicates that the choice of modulation technique can significantly affect the performance of the user scheduling algorithm, and it is essential to choose an appropriate modulation technique that suits the requirements of the system.

Furthermore, the fairness of the proposed algorithm was assessed using Jain's fairness index, which is a commonly used metric for measuring fairness in communication systems. The fairness index of 0.99 obtained from the proposed algorithm indicates that the algorithm ensures fairness among all users, which is an essential requirement for any scheduling algorithm. The proposed adaptive user scheduling algorithm based on instantaneous channel rate is a suitable candidate for downlink user scheduling in a massive MIMO system with a large number of antennas. The algorithm provides improved performance in terms of sum rate, per-user throughput, error performance, and fairness, and can be adapted to different modulation techniques to suit the requirements of the system.

## REFERENCES

[1] M. Giordani, M. Polese, M. Mezzavilla, S. Rangan, and M. Zorzi, ”Toward 6G Networks: Use Cases and Technologies,” *IEEE Commun. Mag.*, vol. 58, pp. 55-61, 2020.

[2] A. Kurve, ”Multi-user MIMO systems: The future in the making,” *IEEE Potentials*, vol. 28, pp. 37-42, 2009.

[3] G. J. Foschini and M. J. Gans, ”On Limits of Wireless Communications in a Fading Environment When Using Multiple Antennas,” *Wireless Pers. Commun.*, vol. 6, pp. 311-335, 1998.

[4] Q. H. Spencer, C. B. Peel, A. L. Swindlehurst, and M. Haardt, ”An introduction to the multi-user MIMO downlink,” *IEEE Commun. Mag.*, vol. 42, pp. 60-67, 2004.

[5] A. Paulraj and T. Kailath, ”Increasing Capacity in Wireless Broadcast Systems Using Distributed Transmission/Directional Reception (DTDR),” U.S. Patent 5,345,599, Sep. 6, 1994.

[6] D. Nojima, L. Lanante, Y. Nagao, M. Kurosaki, and H. Ochi, ”Performance evaluation for multi-user MIMO IEEE 802.11ac wireless LAN system,” in *Proceedings of the 2012 14th International Conference on Advanced Communication Technology (ICACT)*, PyeongChang, Korea, Feb. 19-22, 2012, pp. 804–808.

[7] A. Paulraj, R. Nabar, and D. Gore, *Introduction to Space-Time Wireless Communications*. New York, USA: Cambridge University Press, 2008.

[8] IEEE Draft Standard Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 4: Enhancements for Higher Throughput. P802.11n D3.00, Sept. 2007.

[9] 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (EUTRA); Multiplexing and channel coding (Release 9). 3GPP Organizational Partners TS 36.212 Rev. 8.3.0, May 2008.

[10] J. Hoydis, K. Hosseini, S. Ten Brink, and M. Debbah, ”Making smart use of excess antennas: Massive MIMO, small cells, and TDD,” *Bell Labs Technical Journal*, vol. 18, no. 2, pp. 5-21, Sep. 2013.

[11] R. Chataut and R. Akl, ”Optimal pilot reuse factor based on user environments in 5G Massive MIMO,” *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, NV, 2018, pp. 845-851.

[12] R. Chataut, R. Akl and U. K. Dey, ”Least Square Regressor Selection-Based Detection for Uplink 5G Massive MIMO Systems,” *2019 IEEE 20th Wireless and Microwave Technology Conference (WAMICON)*, Cocoa Beach, FL, USA, 2019, pp. 1-6.

[13] T. L. Marzetta, ”Noncooperative cellular wireless with unlimited numbers of base station antennas,” *Wireless Communications, IEEE Transactions on*, vol. 9, no. 11, pp. 3590-3600, 2010.

[14] E. G. Larsson, F. Tufvesson, O. Edfors, and T. L. Marzetta, ”Massive MIMO for Next Generation Wireless Systems,” *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 186-195, Feb. 2014.

[15] F. Rusek, D. Persson, B. K. Lau, E. G. Larsson, T. L. Marzetta, O. Edfors, and F. Tufvesson, ”Scaling up MIMO: Opportunities and Challenges with Very Large Arrays,” *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 40-60, Jan. 2013.

[16] T. L. Marzetta, ”Massive MIMO: An Introduction,” in *Bell Labs Technical Journal*, vol. 20, pp. 11-22, 2015.

[17] R. Chataut and R. Akl, ”Massive MIMO Systems for 5G and beyond Networks—Overview, Recent Trends, Challenges, and Future Research Direction,” *Sensors*, vol. 20, no. 10, p. 2753, 2020.

[18] R. Chataut, R. Akl, U. K. Dey, and M. Robaei, ”SSOR Preconditioned Gauss-Seidel Detection and Its Hardware Architecture for 5G and beyond Massive MIMO Networks,” *Electronics*, vol. 10, no. 5, p. 578, 2021.

[19] G. Dimic and N. D. Sidiropoulos, ”On downlink beamforming with greedy user selection: performance analysis and a simple new algorithm,” *IEEE Transactions on Signal Processing*, vol. 53, no. 10, pp. 3857-3868, Oct. 2005.

[20] J. Wang, D. J. Love, and M. D. Zoltowski, ”User selection with zero-forcing beamforming achieves the asymptotically optimal sum rate,” *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3713-3726, Aug. 2008.

[21] M. Kobayashi and G. Caire, ”Joint beamforming and scheduling for a multi-antenna downlink with imperfect transmitter channel knowledge,” *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 7, pp. 1468-1477, Sep. 2007.

[22] K. Lyu, ”Capacity of multi-user MIMO systems with MMSE and ZF precoding,” in *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, San Francisco, CA, 2016, pp. 1083-1084.

[23] D. L. Colon, F. H. Gregorio, and J. Cousseau, ”Linear precoding in multi-user massive MIMO systems with imperfect channel state information,” in *2015 XVI Workshop on Information Processing and Control (RPIC)*, Cordoba, 2015, pp. 1-6.

[24] M. Sharif and B. Hassibi, ”On the capacity of MIMO broadcast channels with partial side information,” *IEEE Trans. Inf. Theory*, vol. 51, no. 2, pp. 506–522, Feb. 2005.

[25] T. Al-Naffouri, M. Sharif and B. Hassibi, ”How much does transmit correlation affect the sum-rate scaling of MIMO Gaussian broadcast channels?,” *IEEE Trans. Commun.*, vol. 57, no. 2, pp. 562–572, Feb. 2009.

[26] T. Yoo and A. Goldsmith, ”On the optimality of multi-antenna broadcast scheduling using zero-forcing beamforming,” *IEEE J. Sel. Areas Commun.*, vol. 24, no. 3, pp. 528–541, Mar. 2006.

[27] H. Yang, ”User Scheduling in Massive MIMO,” in *Proceedings of the 2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Kalamata, Greece, 25–28 June 2018, pp. 1–5.

[28] S. Huang, H. Yin, J. Wu, and V.C.M. Leung, ”User selection for multi-user MIMO downlink with zero-forcing beamforming,” *IEEE Trans. Veh. Technol.*, vol. 62, pp. 3084–3097, Sept. 2013.

[29] Y. Cai, J. Yu, Y. Xu, and M. Cai, ”A comparison of packet scheduling algorithms for OFDMA systems,” in *Proceedings of the 2008 2nd International Conference on Signal Processing and Communication Systems*, Gold Coast, QLD, Australia, 15–17 Dec. 2008, pp. 1–5.

[30] M.F. Hamdi, R.A. Saeed, and A. Abbas, ”Downlink scheduling in 5G massive MIMO,” *J. Eng. Appl. Sci.*, vol. 13, pp. 1376–1381, Apr. 2018.

[31] K. Djouani, G. Maina, M. Mzyece, and G. Muriithi, ”A low complexity greedy scheduler for multiuser MIMO downlink,” in *Proceedings of the Southern African Telecommunications Networks and Applications Conference (SATNAC)*, Port Edward, South Africa, 5–8 Sept. 2010.

[32] R. Chataut and R. Akl, ”Channel Gain Based User Scheduling for 5G Massive MIMO Systems,” in *2019 IEEE 16th International Conference on Smart Cities: Improving Quality of Life Using ICT and IoT, and AI (HONET-ICT)*, Charlotte, NC, USA, 2019, pp. 049-053.

[33] R. Jain, D. Chiu, and W. Hawe, ”A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Systems, Digital Equipment Corporation,” *Technical Report DEC-TR-301*, Tech. Rep., 1984.

[34] R. Akl, ”An Efficient and Fair Scheduling for Downlink 5G Massive MIMO Systems,” in *11th IEEE Texas Symposium on Wireless and Microwave Circuits and Systems (TSWMCS 2020)*, May 2020.

**Robin Chataut** is an assistant professor in the School of Computing and Engineering at Quinnipiac University, Hamden, USA. He obtained his undergraduate degree in Electronics and Communication Engineering from Pulchowk Campus, Tribhuvan University, Nepal, in 2014 and his Ph.D. in Computer Science and Engineering from the University of North Texas, Texas, USA, in 2020. Prior to completing his Ph.D., he worked as a senior software developer.

His research interests are in the areas of wireless communication and networks, 5G, 6G, and beyond networks, vehicular communication, smart cities, Internet of Things, wireless sensor networks, and network security. He has designed, implemented, and optimized several algorithms and hardware architectures for precoding, detection, user scheduling, channel estimation, and pilot contamination mitigation for massive MIMO systems for 5G and beyond networks. He has authored and co-authored several research articles. He is an active reviewer in several international scientific journals and conferences.

**Robert Akl** received his B.S. in Computer Science and B.S. in Electrical Engineering in 1994, his M.S. in Electrical Engineering in 1996, and his D.Sc. in Electrical Engineering in 2000, all from Washington University in Saint Louis. He is currently a Tenured Associate Professor at the University of North Texas and a Senior Member of IEEE. He has designed, implemented, and optimized both hardware and software aspects of several wireless communication systems for cellular, Wi-Fi, and sensor networks.

Dr. Akl has broad expertise in wireless communication, Bluetooth, Cellular, Wi-Fi, VoIP, telephony, computer architecture, and computer networks. He has been awarded many research grants by leading companies in the industry and the National Science Foundation. He has developed and taught over 100 courses in his field. Dr. Akl has received several awards and commendation for his work, including the 2008 IEEE Professionalism Award and was the winner of the 2010 Tech Titan of the Future Award.

# A Blockchain-based Security Assessment Framework

N. Satyanarayana
eSecurity De*partment, Centre for Development of Advanced, Computing* Hyderabad, India
nanduris@cdac.in

***Abstract*–Using Blockchain Technology for Security Assessment results in effective monitoring capabilities especially when data analytics components are inbuilt in such a system. At present days, we can see the availability of many Security Information and Event Management (SIEM) tools that follow a client-server model for capturing data from different resources and performing data analysis on the server side. However, such tools serve the purpose of a single institute and depend on the trust level in a multi-institute or multi-center-project kind of environment where they can be used. Another limitation could be that if the server is attacked, the whole exercise would be futile. The lack of trust and concerns about data integrity in such an environment makes performing root cause analysis of security risks difficult. Blockchain technology ensures a tamper-proof, time-stamped, and decentralized storage repository that helps in maintaining data integrity even in complex and untrusted multi-institute or multi-center-project environments while assuring data provenance. This article presents a unified and comprehensive security assessment framework that produces a compliance report along with threat perception level by monitoring and assessing resources across multi-institute or multi-center-project in different geographical locations while supporting data privacy by leveraging Blockchain Technology capabilities.**

***Keywords* - *Blockchain, Security Assurance Policy, Continuous Monitoring***

## I. Introduction

Ccontinuous monitoring of critical digital resources, processes, networks, and resource utilization patterns, and reporting the incident about the observed violations is an essential part of any organization's security framework. The success of an audit process depends on data integrity while running security assessment tools such as SIEM tools. SIEM tools produce reports instantaneously when they run in a machine. For a detailed analysis of security assessment history of events and related data need to be captured in a manner such that the data integrity is maintained forever. Hence, we need a technology that maintains data provenance in a tamper-proof and time-stamped manner so that the security framework is assured of data integrity at any time.

Moreover, such a provision will help SIEM tools to produce more effective reports when data analytics components are integrated for fine-grained analysis. Apart from the above, there should be a mechanism using which one can see to what extent the underlying security policy is conformant and its current severity level to indicate a perceived threat. If unique state replication of data provenance is ensured at the premises of the service provider and other stakeholders then compliance to the organization's security policy framework can be provided as a Software Service with continuous monitoring capabilities in a decentralized manner.

This kind of arrangement ensures transparency and trust in the organization's security policy assessment process among the stakeholders. Blockchain Technology ensures unique state replication of data in the underlying network based on consensus.

In a multi-institute or multi-center-project scenario where critical resources are spread across different geographical locations, we can utilize a permissioned Blockchain-based security assessment mechanism for resource monitoring and security analysis. The usage of permissioned Blockchain eliminates the concerns about the integrity of security-relevant data that is available at different geographical locations as it maintains data in a tamper-proof, authentic, and shares information through a secured communication channel. On top of that the permissioned Blockchain also supports data privacy by sharing security-relevant data between intended members/stakeholders only.

Blockchain Technology is a distributed ledger technology that ensures unique state replication across the participating nodes in a tamper-proof and time-stamped manner. The data once stored cannot be modified or deleted. This is because data will be stored in the form of a block whose header contains a hash of the previous block and so on. The same process of appending a new block to the existing chain of blocks in each of the participating nodes will be ensured by the consensus algorithms in the Blockchain network. Hence, to modify data in the Blockchain network, an attacker not only has to recompute the hash of the corresponding block but also the hash of the next blocks up to the end block. And this entire effort has to be replicated in each and every participating node of the Blockchain network. Considering the effort required to modify the data in the Blockchain network we can be assured of the security and integrity of data that is stored in the Blockchain network. There are tools that provide Application Programming Interface to collect various metrics from the cloud providers and leave it to the interested party to deduce the inference, information about violations, and corresponding mitigation plan in a particular machine/VM etc. There is also research work done on using

Manuscript received Dec 09, 2022. This work was supported by the Centre for Development of Advanced Computing, Hyderabad through a funded project sponsored by the Ministry of Electronics and Information Technology, Government of India. The manuscript is submitted in the follow-up of the invited journal to the accepted and presented paper entitled "A Blockchain-based Security Information and Event Monitoring Framework" of the 25th International Conference on Advanced Communication Technology (2023).

N. Satyanarayana is a Joint Director at the Centre for Development of Advanced Computing, Hyderabad, India. (e-mail: nanduris@cdac.in).

Blockchain technology to maintain the provenance of data objects corresponding to the cloud platform. To the best of our knowledge, a unified approach where evidence collection in a tamper-proof and time-stamped manner and maintenance of the same based on consensus among its stakeholders in a distributed storage system on one side and presenting the standards conformance along with perceived severity level in the whole system on another side is not present. We addressed this problem in our paper.

Major contributions of this paper include,

- Maintenance of data provenance of security log information in a tamper-proof and time stamped using Hyperledger Fabric Blockchain Platform.
- Design of a unified security assurance platform considering best practices for preventing security threats and a corresponding verification mechanism with information about perceived severity threat level.
- Provision of Security Assessment as a Service with unique state replication across multiple nodes from which stakeholders can ascertain information about current security policy conformation.
- Describes an architecture wherein multi-centre or multi-centre-project stakeholders can perform data analysis in an independent manner while maintaining data in a decentralized environment.
- Briefs about how to circumvent issue of dealing with storage space availability in this distributed and decentralized system.

In section 2, we shall present the background study, our approach detailing the cyber security policy framework model and implementation details in section 3, results in section 4, and conclusion in section 5.

## II. Related Study

Hyperledger Fabric (HLF) is a Linux Foundation's Blockchain initiative. The transaction flow (data) of HLF, ledger maintenance, and the purpose of the channel can be seen in [3][20][21]. Researchers claimed that up to 2500 tps could be achieved by them when HLF is used as a Blockchain platform [19]. In HLF, a channel is nothing but a logical subnetwork that binds its members together so that any information can be exchanged among the members themselves and others will not be able to see the data exchange. We can logically group auditors, service providers, and customers of a business entity as members of one or more logical organizations and can make them subscribe to a particular channel. In HLF such logical organizations become part of a consortium. We can have multiple consortiums configured in HLF so that no two consortium members can share data between themselves because of a channel as described above.

Major security vulnerabilities that were exploited in Cloud services were reported in Top Security Threats in Cloud Computing [4] by Cloud Security Alliance (CSA). From the information provided in [4] it can be inferred that 24x7 monitoring of security baseline controls such as password quality verification, continuous monitoring of behavioral anomalies w.r.t users, processes etc., and network policies etc., would provide a protective or vigilant cloud environment. The CSA Cloud Controls Matrix (CCM) is a cyber-security control framework for cloud computing, composed of 133 control objectives that are structured in 16 domains. It can be used as a guide to determine which security controls should be implemented by which actor for the systematic assessment of a cloud implementation. The controls in the CCM are mapped against industry-accepted security standards, regulations, and control frameworks including but not limited to: ISO 27001/27002/27017/27018, NIST SP 800-53, AICPA TSC, ENISA Information Assurance Framework, German BSI C5, PCI DSS, ISACA COBIT, NERC CIP, and many others [5].

Cloudwatch[6] from Amazon collects monitoring and operational data in the form of logs, metrics, and events providing the end user with a unified view of AWS resources, applications, and services that run on AWS and on-premise servers. This is a service provided by Amazon itself for monitoring the health information of its instances and other relevant data to the end-users. Lynis [7] is a security tool for systems running Linux, macOS, or Unix-based operating systems. It performs an extensive health scan of systems to support system hardening and compliance testing. CIS-CAT (CIS-Configuration Assessment Tool) [8] compares the configuration of target systems to the security configuration settings recommended in machine-readable content, provided the content conforms to Security Content Automation Protocol (SCAP). Both the tools (Lynis and CIS-CAT) generate security-relevant information of critical resources in the same physical asset where the tool is deployed and depends on the running status of certain services such as "auditd" in Linux-based systems for the collection of information. These tools either leave the job of security assessment inference to the end-users through an API framework or they can be used in single nodes.

Several researchers have identified the need for addressing cloud security and explored using Blockchain in Cloud Platforms for security-related aspects [1][2][15][16][17]. Blockchain-based data provenance architecture for cloud environments has been proposed by [9][14][18]. In their approach, the focus was mainly on providing the ability to audit or privacy protection of cloud data objects or related operations of a cloud platform. The emphasis was more on using the data provenance capability of the underlying blockchain network for security analysis.

## III. Our Approach – Model, Policy Framework and Implementation

### A. Security Assurance Model

From the background study, we can observe that several efforts have been made in Cloud security by different organizations or standard bodies, primarily CSA, ISO, and individual cloud platform vendors. However, it is confusing which standard to follow from a plethora of standards by different organizations or else which platform to choose from the available platforms in the market as each one has its own metrics resulting in vendor lock-in. Each effort is in its own direction with a common goal of ensuring cloud security assurance. We need a holistic view of the security assurance framework. To fill this gap, we developed a security assurance policy framework based on the model defined in Fig. 1. and implemented monitoring of 14 controls as given in Table I below.

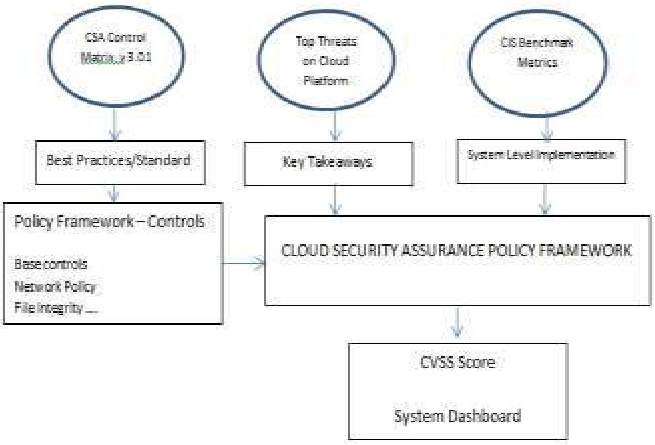

CSA Control Matrix v 3.01
Top Threats on Cloud Platform
CIS Benchmark Metrics
Best Practices/Standard
Key Takeaways
System Level Implementation
Policy Framework – Controls
Base controls
Network Policy
File Integrity ....
CLOUD SECURITY ASSURANCE POLICY FRAMEWORK
CVSS Score
System Dashboard

Fig. 1. Security Assurance Policy Framework Methodology

The list is not limited to these controls itself and can be extended in the future. The controls are all relevant to the key takeaways that were studied based on CSA Top Cloud Threats and corresponding preventive mechanisms. These controls have been implemented for continuous monitoring using the Centre for Internet Security benchmark metrics [10]. These controls have been mapped to different control domains of CSA Control Matrix v 3.0.1 in order to present the compliance w.r.t Cloud Control Matrix best practices. In Table I below, column 1 represents the description of the control being monitored, and column 2 represents the Cloud Control Matrix control domain to which the control name is mapped. When monitoring of a control name is implemented based on CIS benchmark metrics, its security threat perception can be graded in accordance with the Common Vulnerability Scoring System (CVSS) score. The combination of the control name, it's mapping to the control domain, and its CVSS score thus form the security assessment policy framework from which the overall compliance of all critical resources of the business entity can be ascertained. The CVSS score could be one of None, Low, Medium, High, or Critical. The scores would be calculated in pursuance to [12]. The rule engine to calculate the CVSS score for each policy rule can be obtained based on the below-shown pseudo code.

TABLE I
UNIFIED CLOUD SECURITY ASSURANCE POLICY FRAMEWORK

| Control Name | CCM Control Domain |
|---|---|
| File Integrity | AIS-04 Data Security/Integrity |
| Apache Loaded Modules | IVS-07 (OS Hardening and Base Controls) |
| User Login Attempts | IVS-07 (OS Hardening and Base Controls) |
| Password Policy | IVS-07 (OS Hardening and Base Controls) |
| Secure Boot Setup | IVS-07 (OS Hardening and Base Controls) |
| Process Monitoring | IVS-07 (OS Hardening and Base Controls) |
| Network Policy (IP Tables) | IVS-06 (Network Security)<br>IAM-01 (Audit Tool Access)<br>IVS-07 (OS Hardening and Base Controls) |
| Cron Service | IVS-07 (OS Hardening and Base Controls) |
| Syslog Configuration | IVS-07 (OS Hardening and Base Controls), IAM-01 (Audit Tool Access) |
| Secure Service User Accounts | IVS-07 (OS Hardening and Base Controls), IAM – 01 (Audit Tool Access) |
| Apache Logs Configuration | IVS-07 (OS Hardening and Base Controls), IAM – 01 (Audit Tool Access) |
| Apache User Configuration | IVS-07<br>(OS Hardening and Base Controls), , IAM – 01 (Audit Tool Access) |
| TCP Wrapper | IAM – 01 (Audit Tool Access), IVS-06 (Network Security) |
| SSH Service Configuration | IVS-07 (OS Hardening and Base Controls), IAM – 01 (Audit Tool Access) |

Implementation of each control name may have a dependence on the occurrence of more than one event associated with it. For example, Process monitoring control compliance would be verified based on various events such as (a) Whether a process is system-oriented or network-oriented, (b) if it is network oriented whether access restrictions are present in the network policy or not (c) whether the log information corresponding to the process has appropriate access permissions or not (d) if it is network oriented whether the process owner has "nologin" shell or not. For each event, the CVSS score would be computed and the highest severity level among all the events that correspond to a control name would become that control's severity level.

**Pseudo code for determining CVSS score**

```
foreach (observed_error) {
   if (user_account has loginshell) {
          if (network_outer_allowed) {
             if (public_ip) {
                av = network;
             } else {av = adjacent'}
          } else {av = local}
   } else {av = physical;}

   if (user_account == nologin_shell) {
         ac = high; scope_change = false; pr = high;
      confidentiality = none;
      availability = none;
   integrity = none; }
     elsif (passwd_policy == weak &&
                         user_account == login_shell) {
      ac = low; and scope_change = true; pr = low;
      confidentiality = high;
      availability = high;
      integrity = high;
   }elsif (passwd_policy == strong && user_account ==
login_shell) {
   ac = high; and scope_change = false; pr = high;
        confidentiality = low;
      availability = low;
      integrity = low;   }
    }
```

This model gives three important dimensions from which the security assessment of the business entity can be ascertained. Security Threat Perception of a critical resource being monitored. (2) Strength/Weakness of the critical resource in a particular domain in a particular control domain in accordance with the CCM. (3) Overall compliance to the security assessment policy framework of the business entity considering all the critical resources.

### *B. Implementation of Security Assurance Framework – System Architecture*

In order to realize the effectiveness of the policy framework the policy has to be implemented as a software module. Fig. 2. depicts the system architecture which implements the proposed security policy framework using Blockchain Technology.

We used Hyperledger Fabric (HLF) [11] as a blockchain platform. In each resource that is being monitored, there will be an agent program running hereinafter called 'Agent'. The agent program reads the log contents from the respective resource corresponding to each observable control as defined in Table 1, encapsulates the same in a JSON object format, and transfers the same to the Blockchain platform. During this process each resource that is being monitored encrypts the JSON object using its private key and the same will be sent to the HLF Blockchain node. The array containing error ids represents events where policy violations have been detected. A hash value of all the fields in the JSON object is computed and is also made part of the JSON object. This is useful for data validation upon receipt by the Blockchain node. The JSON object format to be stored in the Blockchain is as follows.

```
{
   "hashvalue":" ab232lalkn23,nlk….",
   "data": {
     "host":"xx.xx.xx.xx",
         "timestamp":"02-02-2021",
           "valid":"false" //in case of detection of   policy
                              violation
           "result":[errid1, errid2, errid3….] //list of
           observed violations
       "metricid":"3232" //policy control id
          "condition": {
          Loginshell: [portno, exists or not exists, uname]
              public_ip: yes or no //error can occur from
              outer network
          }
   }
}
```

Blockchain node upon receiving the JSON object would decrypt the received JSON object using the resource public key and recalculates the hash of all the fields in the JSON object and verify its integrity. Modified JSON objects would be rejected and will not be stored in the Blockchain node. This verification mechanism is implemented as a smart contract so that the same validation rules will be applied to each and every node of the Blockchain network.

### *C. Role of Blockchain Nodes*

The advantages of using Blockchain in this architecture are (a) The security policy-relevant information collected from each resource gets replicated in all the Blockchain nodes in a tamper-proof and time-stamped manner based on the underlying consensus mechanism (b) Reduces log processing load on the resource being monitored by delegating the data processing to the Blockchain platform. (c) Due to consensus-based unique state representation of data across all Blockchain nodes, all stakeholders viz. service providers, consumers, and auditors of a business entity would see the same data provenance or its analysis report at any given time without loss of generality. This unique state representation of data in the HLF blockchain network is supported by the Raft consensus algorithm [13] which is considered a robust crash-tolerant consensus algorithm. These Blockchain nodes each can be kept at different geographical locations or in a single data center as per the business entity's need. Since this is a permissioned Blockchain only authorized users can access/retrieve data from the Blockchain nodes. A more detailed use case scenario is explained in the next section.

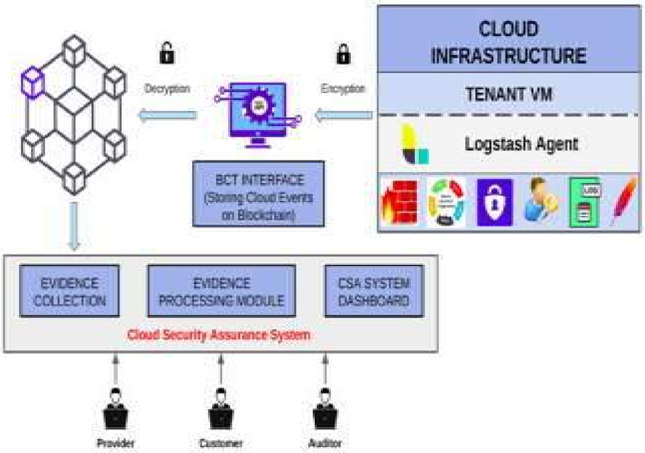


Fig. 2. The architecture of the Security Assurance Framework powered by Blockchain

In this system architecture, we defined three logical organizations namely Customer, Service Provider, and Information Security Auditor. Each logical organization can be assumed as a representation of real-time entities of respective stakeholders. The advantage of using three different logical organizations is that the nodes which are part of these organizations will maintain a unique state of log data as well as users of respective logical organizations only can initiate transactions (insert/query) on Blockchain network. Since logical organizations have been defined based on the functionalities of respective stakeholders who are concerned about the security aspects of the business entity, and all stakeholders are ensured of unique state replication with tamper-proof capability the security assessment of the business entity can be regarded as a fool-proof, robust and trustworthy system. Users from different geo-graphical locations can access such a system as Software-as-a-Service. Having been informed about the role of the Blockchain network and its necessity, we now focus on how the system reports its findings. For this purpose, we have incorporated two different metrics (a) Compliance rate – Which informs to what extent the service provider is compliant with the pre-defined security assurance policy as per table 1, (b) Severity level – Which informs the perceived threat severity level as per the criteria defined by CVSS scores to measure the overall health report of the resources controlled by the business entity.

Compliance Rate (CR) = $(X_{t-1} – X_t) / X_{t-1}$ where $X_t$ is no of errors observed w.r.t security assurance policy at time t and $X_{t-1}$ is at time t-1. CR represents the observed rate of change w.r.t identified non-compliance factors of security policy. The CR represents the compliance rate against the security policy pertaining to the most recent time window. Time window represents the gap between two successive vulnerability analysis attempts. Time window is a configurable parameter that defines how frequently the security log and other critical information have to be collected from the critical resources to be monitored.

Average Compliance Rate = $1/n \sum CRi$ where 'CR$_i$' is the value of the compliance rate at a given instant of a time window and 'n' is the no of such time windows chosen in a given period (e.g., in the last 24 hours, in the last 30 days etc). Severity Level is one of 'Critical', 'High', 'Medium', 'Low', 'None' labels which is decided based on Base Score. The base score is computed as shown below. The base score and other formulae computation are done in accordance with section 7.1 of CVSS specification.

| | |
|---|---|
| Impact = | |
| If Scope is Unchanged | 6.42 * ISS |
| If Scope is Changed | 7.52 * (ISS – 0.029) – 3.25 * (ISS – 0.02)$^{15}$ |
| Exploitability = | 8.22 * AttackVector * AttackComplexity * Privileges Required * User Interaction |
| Base Score = | |
| If Impact <= 0 | 0, else |
| If Scope is Unchanged | Roundup(Minimum[(Impact + Exploitability),10]) |
| If Scope is Changed | Roundup(Minimum[1.08 * (Impact + Exploitability),10]) |

**(Source:** https://www.first.org/cvss/v3.1/specification-document)

The Impact Sub-score (ISS) is calculated as $[1-[(1-confidentiality)*(1-integiry)*(1-Availability)]$

The severity level is decided based on base score value ranges as follows.

| Rating | CVSS Score |
|---|---|
| None | 0.0 |
| Low | 0.1 – 3.9 |
| Medium | 4.0 – 6.9 |
| High | 7.0 – 8.9 |
| Critical | 9.0 – 10.0 |

## IV. Implementation & Results

We have implemented an 'Agent' code using "logstash" for each control that is described in the security policy framework depicted in Table 1 above. The log information from each resource is routed through an intermediate gateway developed using Node.JS server-side program which acts as a client to the HLF Blockchain network. The Node server upon receiving the encrypted JSON object sends the same unaltered to the Blockchain network. The smart contract later decrypts the JSON Object and stores the same in the Blockchain network. The flow of user interaction with the system is depicted in Fig. 3. below. A Dashboard component also has been developed using the Angular programming framework which can be accessed from the web by end-users who are members of service providers, customer, and auditor organizations. The dashboard component displays the no of violations against the security policy framework as described in Table I in a graphical format. It also displays various metrics such as no of resources audited in the platform, the average compliance rate over a period of time (e.g., last 24hrs, 7 days, 30 days). We also wanted to understand the effectiveness of the security policy compliance rate being computed by the system. To observe this we relied upon measuring severity levels in accordance with CVSS scores. If the security policy has any impact, then it should get reflected in observed severity levels.

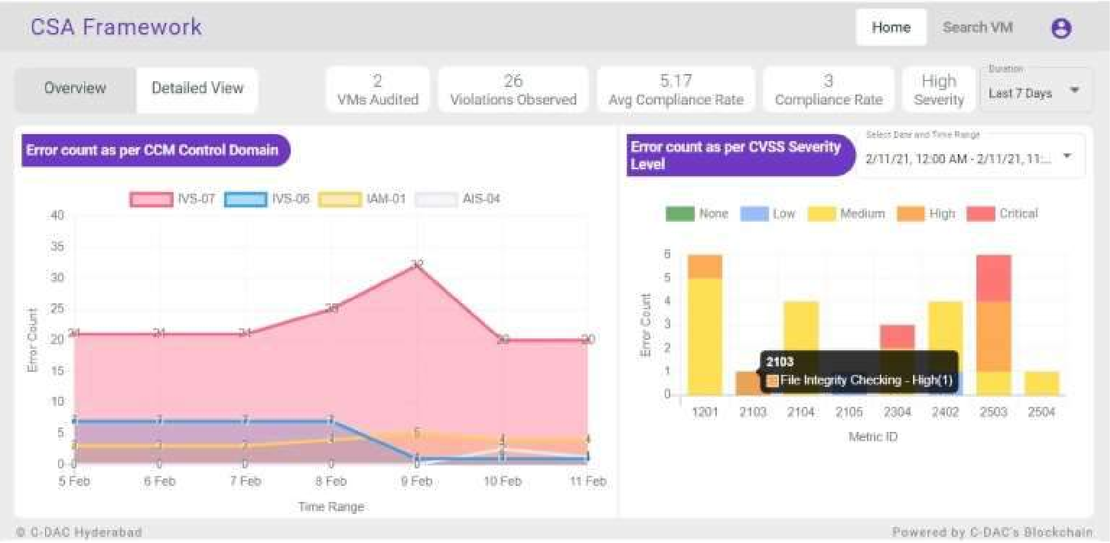


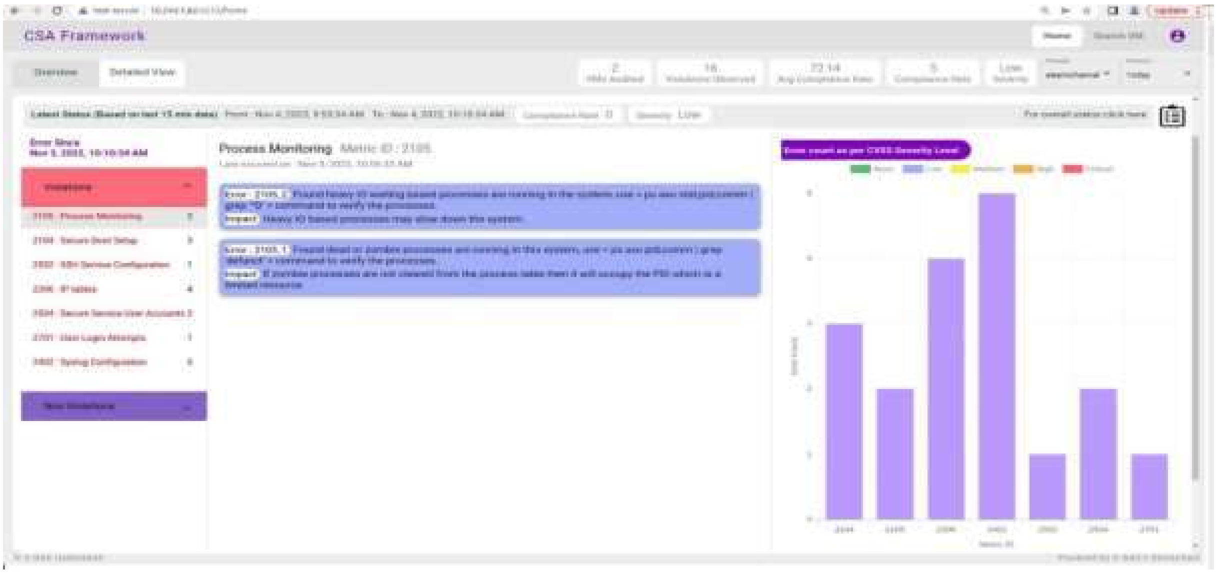
Fig. 3. Display of resources being monitored and their stats

Fig. 3. above depicts the snapshot of the dashboard where no of resources audited, compliance rate, severity level etc. The dashboard also provides a detailed error report comprising an error description, its impact, and a mitigation plan for each and every violation that is observed by the system.

In Fig. 3., it can be observed that the left side graph shows the observed events information which are considered as violations as per the underlying security policy. The graph shows the details in accordance with the CCMv3 guidelines of Cloud Security Alliance. The right side of the graph displays the no of violations observed corresponding to each category of events in terms of their CVSS score.

Fig. 4., depicts the user interface screen wherein the end-user can observe the different types of errors or violations that have been monitored by the underlying system, its severity level, and also the mitigation plan. From this screen one can observe how many controls with which the resources being monitored are compliant can be observed by clicking on the chart icon on the top right corner.

The screen can be used to observe the error description of each violation that is observed in a resource, its impact as well as mitigation plan also.


CSA Framework
Overview
Detailed View
Process Monitoring Metric ID : 2105
Violations
2105 - Process Monitoring 5
2104 - Secure Boot Setup 3
2602 - SSH Service Configuration 1
2206 - IP tables 4
2604 - Secure Service User Accounts 2
2701 - User Login Attempts 1
2402 - Syslog Configuration 6
Non Violations
Error count as per CVSS Severity Level
None
Low
Medium
High
Critical
Metric ID

Error : 2105_2 Found heavy IO waiting based processes are running in the system; use » ps axo stat,pid,comm | grep "D" » command to verify the processes.
Impact Heavy IO based processes may slow down the system.

Error : 2105_1 Found dead or zombie processes are running in this system; use » ps axo pid,stat,comm | grep defunct » command to verify the processes.
Impact If zombie processes are not cleared from the process table then it will occupy the PID which is a limited resource.

Fig. 4. Observed e/vents and corresponding event details

## A. Use Case Scenario

The use case scenario will give a clear picture of how the organizations can get benefitted using this software. The software can be used in a multi-institute or multi-center-project environment wherein different stakeholders from different geographical locations work in a collaborative manner. Multi-institutional or multi-center-project-based resources can be monitored for security vulnerability based on a pre-defined security policy framework and maintain relevant data in a Blockchain network, hosted by relevant stakeholders, spread across different geographical locations for data analytics. Since the data is maintained in Blockchain all relevant stakeholders are ensured of the uniqueness and correctness of data at their premises and use such data for security analysis purposes in an independent manner. Hence, Institutions working in a multi-center, multi-project, or collaborative manner with other institutes and having the requirement of 24x7 monitoring of critical resources for security breaching based on a pre-determined security policy framework that applies universally among the participating entities can use this model or system. One such example scenario could be the Cyber Insurance domain where the cyber insurer requires a mechanism wherein monitoring of resources can be done in a seamless manner even in an untrusted network environment. Another such example could be where a consortium project is being executed by multiple stakeholders in a consortium manner while utilizing resources located at different places under different network administration teams.

The functional requirements of deployment in this case is as follows.

- ✓ Blockchain Network: As far as this use case is considered Blockchain network represents Hyperledger Fabric-based Blockchain network architecture. In this architecture, the network administrator can define a consortium of logical organizations that would like to share data among members of respective organizations based on pre-defined signing policy and membership.

- ✓ Logical Organization: An organization from the perspective of a Blockchain network is one that binds peer nodes and users through a common membership. We can use this concept to create several logical organizations that group peer nodes and users, belonging to different physical institutes/centres/project groups. Once the logical organizations are defined then data received from Blockchain client applications can be shared among the participants of those organizations only.

- ✓ Channel: In order to share the data between the logical organization's peers of respective organizations have to join the specific channel which is nothing but a logical subnetwork that supports data exchange among its members only. Hence, a channel has to be created so that peers of different organizations can join the channel.

- ✓ Critical Resource: Each critical resource that has a public and private key certificate pair and is capable of sending its security log information in a specific data format through a secured transmission medium has to be identified. The critical resource then sends the data in the desired format to the blockchain client application in the respective centre/project group/institute through an agent program. The Blockchain client application then submits the data to the Blockchain network.

- ✓ Blockchain Client Node: A Blockchain client application accepts the data in the specified format from a critical resource and validates its authenticity and integrity based on the resource's certificates. The client application then submits the same as a blockchain transaction to the Blockchain network along with information like channel configuration, user details, and request parameters. The data to be inserted will be forwarded to all the peer nodes that are part of the logical organizations that are participating in the Blockchain network as a consortium. The Blockchain client also receives requests from another entity other than critical resources i.e., the User application to fetch data from the underlying Blockchain network corresponding to a limited time period like the last 24 Hours, last 7 days, etc., in a specific format.

- ✓ Blockchain Peer Node: Each transaction that is submitted by different client applications will be received by one of the peer nodes in each logical organization that is being administered by the Blockchain network. Upon successful processing of transaction data through a Smart Contract program deployed in each peer node, the data will eventually be added to the existing peer ledger in all the Blockchain nodes. Each peer will run the same version of the smart contract. At the same time, each peer can also run multiple smart contracts each bearing a specific version number. Each center/institute/project group can designate one peer node as a member of the blockchain-based logical organization.

- ✓ User App: The user application can be accessed by authorized users with valid certificates representing respective stakeholders such as Cyber Insurer, and System/Network Administrators representing logical organizations of the Blockchain network. Upon receiving the request for the resource's current security status from the end users through the user app, the Blockchain client application issues a query request to the underlying Blockchain platform to fetch the relevant information and then runs a machine learning model or data analytics program to ascertain the overall threat perception level and security compliance rate. Once the threat perception has been computed, the information will be sent as a response to the user's request. The authorized users then can view information like the number of critical resources that are being monitored across different institutes/project groups/centers, the average and overall compliance rate of specific resources being monitored, detailed information of captured events, their impact, and mitigation plan, threat perception level in accordance with CVSS score. Only registered Users Applications can communicate with the Blockchain client node.

- ✓ Ordering Service Nodes: The user application initially sends data to endorsing nodes (Blockchain Peer Nodes) in each organization and collects the read/write set upon executing a Smart Contract on endorsing node. The data will not be committed at this point in time. The User application after collecting read/write set information from each peer will send the same to the ordering service nodes for ordering transaction data and bundle all transactions in a Block in accordance with the pre-defined block size limit. Ordering service nodes after the creation of blocks will broadcast them to all the Blockchain peer nodes in each organization which commits the same upon successful revalidation of transaction data to observe whether any discrepancies are there in the transaction data from the time the read/write sets are collected earlier and before committing them in the same peer node. Blocks will then commit the data either as valid or invalid depending on the validation results. The Ordering Service nodes can be deployed at each participating Institute/Project Group/Centre or one of them can host the ordering service nodes. The same ordering service nodes can be shared and used for multiple project groups/institutes/centers.

In Fig. 5., it is envisaged that two Institutes A and B have joined hands to execute a collaborative project X whose resources are located in both A & B. The architecture is not limited to two only but can be used for many collaborating agencies. Both A & B would like to monitor the security compliance report or threat perception level of all the resources. Based on the above-depicted deployment architecture now it is possible for A and B to run their respective Blockchain client applications to receive data from their respective institute's critical resources and forward them to the Blockchain network.

The Blockchain network in this architecture has two logical organizations namely Org A and Org B which binds one peer node each (more peers can be run to ensure high availability if needed) along with users. A blockchain channel also has been created and made all peers of respective logical organizations members of the same for sharing data among the peers of those organizations only.

Once the Blockchain network receives data from different client applications they forward them to all the peers of both Org A and Org B through Ordering Service nodes and the data will be stored in their respective peer ledgers permanently.

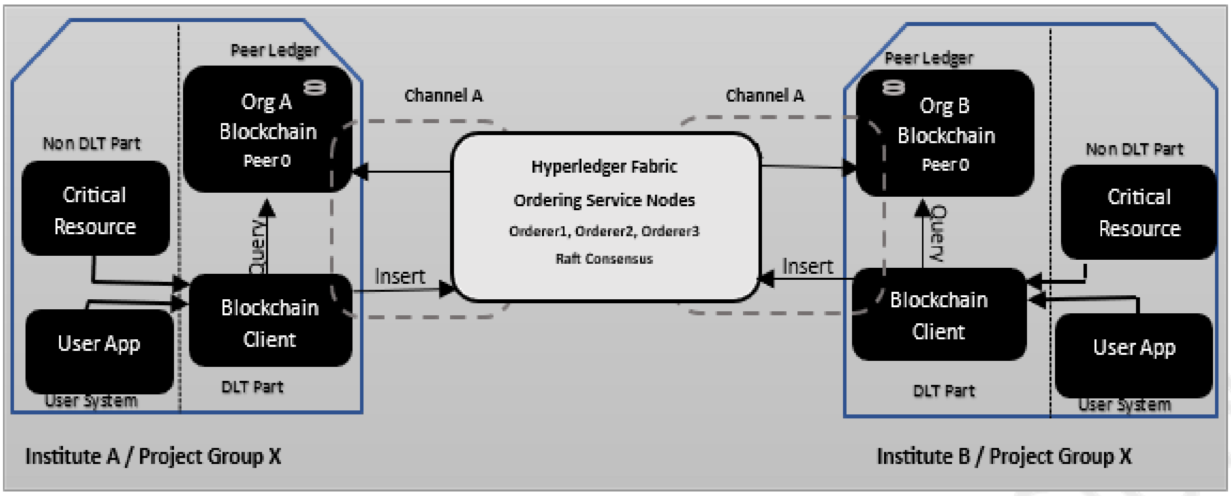


Fig. 5. Security Assessment Blockchain Architecture

The User Apps running at institutes A & B can have a unique representation of the data pertaining to Project X at any given point of time and can run their own analytics programs to infer security compliance information and threat perception level independently. Thus, this way the architecture supports a decentralized network with data replication capabilities while ensuring each participant relies upon a unique representation of data and facilitates each user app to run data analytics independently.

The architecture can support multiple consortiums with multiple channels to delineate the data and data analytics into separate groups and monitor independently.

### B. *Security and Identity Management*

Blockchain Peer and Orderer Node: A peer and orderer node can join the Blockchain network by establishing its membership within an organizational context and generating a public and private key for the peer and orderer nodes.

User Registration: Users with 'READER' and 'WRITER' roles can be registered with the system. READERS are allowed for querying the data and WRITERS are allowed for inserting data into the Blockchain network. User accounts with the WRITER role are used during critical resource registration time and the READER role is used for those user accounts which are meant for accessing the User App components.

Non-DLT System & User System: These are the components which are not related to Blockchain network but are nothing but data producers or consumers. The data source's (or critical resource being monitored) integrity protection is ensured by way of computing the hash of all the fields in the data source object and encrypting it using the data source's private key. The name and password used during key pair generation must be a registered user with the role of 'WRITER'. Since the data source's public key is shared with the Blockchain client (BCTClient), the client can verify the authenticity and integrity of the data once it receives the same and it can insert data into the Blockchain network.

User App: This component's IP address MUST be registered with the BCTClient in order to accept requests from only registered applications. Apart from that the request object coming from User App MUST also provide a username and password for authentication purposes. When the request is from a registered User App and the user is a registered user of the Blockchain network then only the request will be processed further.

Privacy: Blockchain peers, clients, and orderers of different organizations can exchange data among themselves provided they all join the same Blockchain 'channel'. Members of different channels cannot exchange information between themselves. This way by binding peers, clients, and orderers of different organizations that needs to exchange data among themselves only data privacy is achieved.

Event API: The user app that is installed at the respective stakeholder's premises can invoke an API call for retrieving data from the blockchain system periodically to compute overall and individual threat perception levels.

### C. *Performance of the tool*

The software module i.e., the intermediate gateway in our overall architecture, used for retrieval of data from the Blockchain network and to verify security policy conformance, should be designed in such a manner that it can withstand the vast amount of data that is required to be processed and improve the system response time.

It is observed that if logs are collected at 10min intervals in a day, a total of 144 iterations will be required to store the log information in the Blockchain platform. We observed that with 14 control data that are collected 144 times a day and each JSON object size of 594bytes on average, approximately a total of 1.14MB of data would be stored from each resource that is being monitored. For 30 days 34.27MB of data will have to be processed corresponding to a particular resource. After observing this, we designed the below data structure to cope up with data generated by multiple resources. In our experiment, we used nearly 8VMs for monitoring purposes and improved the system response time. The end-user, upon selecting the duration for which the analysis has to be performed in the dashboard, data pertaining to that period will be retrieved from the Blockchain network and maintained in accordance with the below data structure.

```
{
        metric_id: val    # id of the control name
        result: [error1, error2, error3, error4 ….] #stores id of every error that is observed
        timestamp: [ts1, ts2, ts3 ….]  #stores timestamps at which the errors have been observed
        machine_id: xxxx    #id of the resource
        hostname: val   #name of the host
}
```

By default, the data collection interval is set to 10min in our approach. In this format instead of processing individual JSON objects representing each log entry, we rearranged similar entries based on the time of their occurrence while responding to the request from the dashboard component. This has enhanced the system response time considerably.

Since the data is collected from different resources for a specific period it makes the data analysis more effective as events can be related and inferences can be drawn accordingly. We used the pseudo-code explained above for computing threat perception scores by related various events that occurred during the specified period.

### D. *Storage Management*

It is important to understand how the storage space is utilized in each peer node in the Blockchain network. Whenever a peer node confirms a transaction block that it has received from the ordering service node in the Hyperledger network, the peer node stores the same in its ledger. The ordering service ensures an atomic broadcast of the blocks once they are ordered as explained above. Usually, when a peer node in the Blockchain network goes down for any reason and when it rejoins the same network, it will try to contact the other peer nodes to obtain the missed entries in its peer ledger to be in sync with the blockchain network.

Here, the question arises whether we should ensure that all peer nodes must be homogeneous in terms of their storage capabilities. What happens if we use nodes in different logical organizations with different storage spaces?

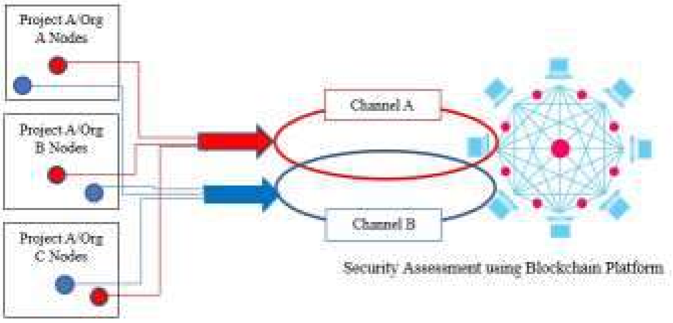


Fig. 6. Schematic diagram of transition from one channel to another channel to cope with available storage space in blockchain nodes

It is explained that from a single resource, approximately 1.14MB of log data has to be maintained in peer ledgers of Blockchain nodes in the network. Apart from that the same data will get replicated in other nodes of the Blockchain network. When multiple resources are monitored those many nodes data has to be stored in the Blockchain node.

When the storage space gets filled in the respective blockchain nodes it is not possible to replace the one which got filled first compared to other nodes. This is because if we replace a node that gets filled first with another node the newly added node will start trying to be sync with other blockchain nodes which are members of the same channel. This situation necessitates that all nodes of Blockchain nodes must be homogenous. This is a serious limitation as it will restrict blockchain usage in a multi-institute or multi-center-project environment where ensuring homogenous nodes is a difficult phenomenon.

We studied the impact of such a requirement and its practical applicability. In general, no production system or network can be brought down temporarily and hence we need to devise an alternative mechanism.

In this regard, we conducted a trial experiment in our lab following the below steps which resulted in a smooth transition of the existing production-grade blockchain network that could handle the above storage management-related issue and also eliminated the requirement of strictly using homogenous nodes across all the organizations.

As shown in Fig. 6., we maintained a single peer node in each logical organization of the Blockchain network and made the signing policy an IMPLICIT MAJORITY so that members of any of the two organizations, when they sign the transaction data, would be sufficient. We also made another node in each logical organization ready with new storage space available and made them members of a new 'channel' that binds only these newly added peers of respective organizations as members. Now we have two channels (please refer to what purpose a channel serves above), one which was there from the beginning and is being used currently. The newly created channel contains new peers of respective organizations. Once this setup is made (for which we need not bring down the production network), we deployed the same smart contract that is used for validating transaction data (the log data) in the existing channel on to the new channel also. Once the arrangements are completed, we made all subsequent requests from client applications to divert their future request submission to the new channel. Since new channel and smart contract combination is made available on newly added peers, they maintain their own peer ledgers on respective nodes. This way we handled the smooth transition of the production network from one channel to another channel with the same smart contract. This solution not only helped in dealing with storage management effectively but also eliminated the need for worrying about having homogeneous nodes only across the Blockchain network.

### E. *Comparative Analysis*

Table II below depicts the comparative analysis of developed solutions with different security analysis tools. Other tools given in Table II have certain shortcomings when compared to our solution as both of them generate a report in the system in which those tools are deployed. We used freeware versions. Whereas our solution provides a web interface to view the health report of all resources being monitored in one place. Moreover, we also have the CVSS score that displays the severity level to alert the concerned stakeholders. Another important feature of the developed solution is that it automatically collects data at regular intervals and sends them to the Blockchain platform for permanent storage in a tamper-evident manner. This way, we have designed and implemented a unified security policy framework with a 24x7 monitoring and alert system. The system facilitates a unique state of representation of security log information due to the usage of Blockchain because of which inferences can be drawn beyond any doubts on data integrity.

TABLE II
COMPARATIVE ANALYSIS OF DEVELOPED SOLUTION WITH OTHER SIMILAR INITIATIVES

| Controls | Lynis | CIS-CAT | Our Solution |
|---|---|---|---|
| File Integrity | No (Depends on auditd) | No (depends on auditd) | Yes (Own Implementation) |
| Continuous monitoring | No | No | Yes |
| Reports | Yes (On VM where it runs) | Yes (On VM where it runs) | Sends information to Blockchain |
| Impact Details | Partial | Yes | Yes |
| Firewall rules for all open ports | No | Yes | Yes |
| File System Base control | Yes (Doesn't say whether it is right?) | Yes (Say's whether it is right) | Yes (Say's whether it is right) |
| No login shell for system users/network services | No | Yes | Yes |
| Presentation | Text | Yes (html) | Yes (html) |

## V. CONCLUSIONS

We presented a unified and comprehensive security assessment framework that is supported by Blockchain Technology. The security policy framework was developed based on inputs drawn from key takeaways from top threats in the cloud platform, existing cloud best practices/standards such as Cloud Security Alliance, and implementation techniques to provide an early detection mechanism using CIS Benchmarks etc. In this attempt, we demonstrated how Blockchain technology supported the data provenance related to security aspects in the decentralized network and log management in a time-stamped and tamper-evident manner based on consensus among the relevant stakeholders. We also demonstrated an inbuilt mechanism for monitoring the effectiveness of the security policy framework by way of computing severity levels in various resources. During the course of using the Blockchain network the role of blockchain nodes, and challenges w.r.t storage management have been studied carefully and the system has been designed accordingly to provide better performance.

## ACKNOWLEDGMENT

The author acknowledges all their colleagues who supported them in coming out with the results and implementation of the software and MeitY, Government of India.

## REFERENCES

[1] G Ramachandra et., "A Comprehensive Survey on Security in Cloud Computing", *Procedia Comput. Sci.*, vol. 110, no. 2012, pp. 465–472, 2017.

[2] Monjur Ahmed, Mohammad Ashraf Hussain, "Cloud computing and security issues in the cloud", *International Journal of Network Security and Applications.*, Vol.6(1), 2014, pp. 25-36

[3] Z. Zheng, S. Xie, H. Dai, X. Chen and H. Wang. (2017) "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends," *IEEE International Congress on Big Data (BigData Congress),* Honolulu, HI, pp. 557-564.

[4] Technical report from Cloud Security Alliance, "*Top threats to cloud computing: Deep Dive – A case study analysis for 'The Treacherous 12: Top Threats to cloud computing*' and a relative security industry breach analysis.

[5] *Cloud Controls Matrix* from Cloud Security Alliance. Available: https://cloudsecurityalliance.org/research/cloud-controls-matrix/ on July 16, 2020

[6] *CloudWatch User Guide for monitoring Amazon instances health information.* Available: https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/WhatIsCloudWatch.html

[7] *Lynis Documentation* Available: https://cisofy.com/documentation/lynis/

[8] *CIS-CAT Lite tool* Available: https://www.cisecurity.org/blog/introducing-cis-cat-lite/

[9] X. Liang, S. Shetty, D. Tosh, C. Kamhoua, K. Kwiat and L. Njilla, "ProvChain: A Blockchain-Based Data Provenance Architecture in Cloud Environment with Enhanced Privacy and Availability," 2017 *17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, Madrid, 2017, pp. 468-477, doi: 10.1109/CCGRID.2017.8.

[10] *CIS Benchmarks for Ubuntu Linux* Available: https://www.cisecurity.org/cis-benchmarks/

[11] *Hyperledger Architecture, Volume 1 (2017) Introduction to Hyperledger Business Blockchain Design Philosophy and Consensus*. Available: https://www.hyperledger.org/wp-content/uploads/2017/08/Hyperledger_Arch_WG_Paper_1_Consensus.pdf

[12] *Common Vulnerability Scoring System v3.1: Specification Document* Available: https://www.first.org/cvss/v3.1/specification-document

[13] D. Ongaro, J. Ousterhout. (2014) "In search of an understandable consensus algorithm", *Proc. USENIX Conf. USENIX Annu. Tech. Conf. (USENIX ATC)*, pp. 305-320.

[14] C. Yang, L. Tan, N. Shi, B. Xu, Y. Cao and K. Yu, "AuthPrivacyChain: A Blockchain-Based Access Control Framework With Privacy Protection in Cloud," in *IEEE Access*, vol. 8, pp. 70604-70615, 2020, doi: 10.1109/ACCESS.2020.2985762.

[15] Park JH, Park JH. Blockchain Security in Cloud Computing: Use Cases, Challenges, and Solutions. *Symmetry*. 2017; 9(8):164. https://doi.org/10.3390/sym9080164

[16] S. Singh, Y.-S. Jeong, and J. H. Park, "A survey on cloud computing security: Issues, threats, and solutions," *J. Network and. Computer. Applications*., vol. 75, pp. 200–222, Nov. 2016.

[17] B. Duncan, D. J. Pym and M. Whittington, "Developing a Conceptual Framework for Cloud Security Assurance," *2013 IEEE 5th International Conference on Cloud Computing Technology and*

*Science*, Bristol, 2013, pp. 120-125, doi: 10.1109/CloudCom.2013.144.

[18] Sachin Shetty, Val Red, Charles Kamhoua, Kevin Kwiat and Laurent Njilla "Data provenance assurance in the cloud using Blockchain", *Proc SPIE 10206, Disruptive Technologies in Sensors and Sensor Systems*, 102060I (2 May 2017); Available: https://doi.org/10.1117/12.2266994

[19] *Channels — hyperledger-fabricdocs main documentation* Available: "https://hyperledger-fabric.readthedocs.io/en/release-2.2/channels.html"

[20] *Transaction Flow — hyperledger-fabricdocs* main documentation. Available: "https://hyperledger-fabric.readthedocs.io/en/release-2.2/channels.html"

[21] *Ledger — hyperledger-fabricdocs main documentation* Available: "https://hyperledger-fabric.readthedocs.io/en/release-2.2/ledger/ledger.html"

**N Satyanarayana** lives in Hyderabad, India, and is born in 1976. He did his Master of Technology a post-graduation degree in computer science from Jawaharlal Nehru Technological University, Hyderabad, India. Prior to this, he completed his Master of Computer Application post-graduate degree in computer applications from Sri Venkateswara University, Tirupati, India in the year 1999.

He is currently working as Joint Director in the Centre for Development of Advanced Computing (CDAC), Hyderabad, India. He has been working for CDAC for the past twenty years and played an instrumental role in developing various applications in the areas like eLearning, Peer to Peer, Network Management, and Blockchain Technology.

Mr. N Satyanarayana published several papers at National and International conferences in various areas such as Peer to Peer Computing, Network Management, e-Learning, and Blockchain. His current research interest includes Blockchain consensus algorithms and reference architectures. He is also contributing towards best practices/standards in the respective fields of his work being a member of the technical committees of the Bureau of Indian Standards.

# A Horizontal Federated Learning Approach to IoT Malware Traffic Detection: An Empirical Evaluation with N-BaIoT Dataset

Phuc Hao Do[*,***], Tran Duc Le[**], Vladimir Vishnevsky[****], Aleksandr Berezkin[*], Ruslan Kirichek[*, ****]

[*]*The Bonch-Bruevich Saint-Petersburg State University of Telecommunications, Saint-Petersburg, Russia*

[**]*University of Science and Technology – The University of Danang, Da Nang, Viet Nam*

[***]*Danang Architecture University, Da Nang, Viet Nam*

[****] *V. A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, Moscow, Russia*

**haodp@dau.edu.vn, letranduc@dut.udn.vn, vishn@inbox.ru, berezkin.aa@sut.ru, kirichek@sut.ru**

***Abstract* —The increasing prevalence of botnet attacks in IoT networks has led to the development of deep learning techniques for their detection. However, conventional centralized deep learning models pose challenges in simultaneously ensuring user data privacy and detecting botnet attacks. To address this issue, this study evaluates the efficacy of Federated Learning (FL) in detecting IoT malware traffic while preserving user privacy. The study employs N-BaIoT, a dataset of real-world IoT network traffic infected by malware, and compares the effectiveness of FL models using Convolutional Neural Network, Long Short-Term Memory, and Gated Recurrent Unit models with a centralized approach. The results indicate that FL can achieve high performance in detecting abnormal traffic in IoT networks, with the CNN model yielding the best results among the three models evaluated. The study recommends the use of FL for IoT malware traffic detection due to its ability to preserve data privacy.**

***Keyword* — IoT, abnormal traffics, malware detection, federated learning, AI model**

## I. INTRODUCTION

As a result of the rapid growth of the Internet of Things (IoT) technology, IoT devices have become an integral part of people's daily life. However, it inevitably introduces some network security challenges [1][2]. IoT devices are easy targets for malicious attacks such as malware attacks due to their heterogeneity and vulnerability. The prevalence of privacy and security concerns is rising due to the continual expansion of IoT devices and the resulting exposure of more and more private data online. It is essential to monitor IoT networks to prevent malicious cyberattacks on IoT devices [3]. By studying the traffic of IoT devices, network intrusion detection in the IoT ecosystem may be enhanced, and cyberspace security can be guaranteed [4][5][6].

In recent years, the expansion of deep learning has played a significant role in advancing IoT intrusion detection research [7]. Deep neural networks are employed to automatically identify dataset features, thereby reducing the feature engineering workload and improving data processing performance without requiring human intervention. Despite its benefits, the application of the Internet anomaly detection approach to the IoT is not straightforward due to the large amount of data required. Most existing machine learning or artificial intelligence solutions rely on a single server to collect data from various IoT devices and build global models [8]. However, this approach is not always effective, particularly when device actions involve sensitive or private information that could severely impact environmental security and privacy if disclosed to unauthorized parties in IoT networks.

In the context of preserving information privacy and integrity, Federated Learning (FL) [9] has gained increasing relevance. FL involves decentralizing the training model among several nodes or clients that use local data. Each decentralized node trains a distinct model on its data and distributes the model parameters (not the private data) to others through a central entity known as a server or via a peer-to-peer methodology [10]. The model parameters are then aggregated to produce a singular and global model. After several iterations, each client obtains a global model by

Manuscript received on Jan. 19, 2023. This work was supported by the Russian Science Foundation within of scientific project No. 22-49-02023 and it is also supported by a follow-up of the invited journal the accepted & presented paper entitled "A Horizontal Federated-Learning Model for Detecting Abnormal Traffic Generated by Malware in IoT Networks" of the 25th International Conference on Advanced Communication Technology (ICACT2023)

P. H. Do, he is a Ph.D student in the Department of Communication Networks and Data Transmission at Bonch-Bruevich Saint-Petersburg State University of Telecommunications, Russia. (email: haodp@dau.edu.vn).

T. D. Le, he is a professor in Information Technology Faculty, The University of Danang - University of Science and Technology, Danang, Vietnam (email: letranduc@dut.udn.vn).

V. Vishnevsky, he is a professor at V.A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, Russia. (email: vishn@inbox.ru).

Aleksandr Berezkin, he is working at the Bonch Bruevich Saint Petersburg State University of Telecommunications as the Associate Professor of Department of Programming Engineering and Computer Science. (email: berezkin.aa@sut.ru).

Dr. Sc. Ruslan Kirichek, he is a professor in the Department of Communication Networks and Data Transmission at Bonch-Bruevich Saint-Petersburg State University of Telecommunications, Russia. (corresponding author, phone: +7-921-970-0160, email: kirichek@sut.ru).

aggregating their unique models. This strategy naturally supports data privacy because data is not shared with external identities.

In the paper, we aim to build a horizontal Federated-Learning model to detect abnormal traffics, specifically DDoS attack traffic generated by malware, such as Botnet in IoT networks. To achieve this, we plan to utilize popular models like Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU).

In this study, we utilize the N-BaIoT [11] dataset for model training due to its real-world origin, as opposed to other datasets that simulate network traffic. Our primary objective is to evaluate the efficacy of Federated Learning (FL) in comparison to traditional centralized artificial intelligence models. To achieve this goal, we address research questions such as whether FL can accurately detect malware or malicious traffic in IoT networks and which training models are suitable when applied to this decentralized approach.

Hence, the primary contributions of this study are as follows:

- Employing feature selection and feature engineering techniques to preprocess the N-BaIoT dataset and select features that have the most considerable impact on the model's accuracy and performance;
- Deploying and evaluating various deep learning models, including CNN, LSTM, and GRU, in the FL approach with different dataset portions to determine their compatibility and select the most suitable model for the FL approach;
- Conducting a comparative analysis between the FL model and traditional deep learning models using the same dataset, thus assessing the feasibility of FL in detecting abnormal traffic in IoT networks.

The results of this study will be included in the Draft Recommendation ITU-T Q.TSRT_IoT “Test specifications for remote testing of Internet of Things using the probes”.

## II. RELATED WORKS

Several researchers have conducted a thorough investigation to find a more trustworthy anomaly detection strategy for IoT networks. To solve the issue of constrained device resources, the study in [12] created a unique, lightweight attack detection method that uses the Support Vector Machines (SVM) algorithm to recognize hazardous assaults in the IoT space. To offer a distributed intrusion detection system, Ferdowsi et al. recommended anomaly detection utilizing distributed Generative Adversarial Networks (GANs) [13]. Each device may protect user privacy while applying a detection model to its data since it is not maintained centrally. A centralized IoT intrusion detection system based on fog computing was suggested in the publication [14]. Two cascading recurrent neural networks (RNNs) are used in the design, each with its hyperparameters and attack-specific tuning. The system administrator is notified if any of the two RNNs determines that an input instance is malicious. Time-series data from IIoT sensors were utilized by the authors of [15] to identify abnormalities using an attention-based convolutional neural network with LSTM [16].

In recent years, FL has risen to prominence in cybersecurity. This IoT security paradigm has already been used in several works. In this context, the study proposal [17] underlined the problem with traditional AI-based solutions' lack of data privacy. However, a private dataset was used for the examination. Despite sharing similar objectives, the research mentioned in [18, 19] mainly focused on industrial IoT devices. They looked at application samples and sensor readings rather than network data.

The authors introduced the Federated Averaging (FedAVG) approach in [20], which is a strong basis for many FL-based investigations. In this strategy, a central server coordinates the training of a global model across several clients using their datasets. The server is responsible for averaging the client-sent models' parameters and sending the resulting global model back to the clients. Until a termination condition is met, this process is repeated. FL has significantly advanced since several researchers [21] examined the most current advancements in this area.

Two more robust model aggregation functions were suggested in [22]. They are based explicitly on the coordinate-wise mean and median of the clients' models supplied to the server. The authors of [23] suggested resampling as a way to lessen the variability in the distribution of the models that the clients supplied. It should be used before employing a powerful aggregating function. When used with models trained on non-IID datasets, it seeks to lessen any negative consequences that such a function could have.

The paper [24] focuses on the challenging task of optimizing neural architectures in the federated learning framework. The authors describe recent work on federated neural architecture search, which involves searching for optimal neural architectures across multiple clients in a distributed manner. They categorize these approaches into online and offline implementations, as well as single- and multi-objective search approaches. They also explain the different types of federated learning, including horizontal, vertical, and hybrid.

The authors [25] propose a Federated Learning (FL) based IoT Traffic Classifier (FLITC) that uses Multi-Layer Perception (MLP) neural networks to classify traffic data while keeping local data on IoT devices. FLITC sends only the learned parameters to the aggregation server, reducing communication costs and latency. The main idea behind FLITC is to train a shared model on distributed devices without exchanging raw data, thereby preserving privacy and reducing communication overhead. The method is expected to improve the accuracy and efficiency of IoT traffic classification while maintaining data privacy.

The study [26] proposes a federated-learning traffic classification protocol (FLIC) for Internet traffic classification without compromising user privacy. FLIC aims to achieve accuracy comparable to centralized deep learning for Internet application identification without privacy leakage. It can classify new applications on-the-fly when a participant joins the learning process with a new application, which has not been done in previous works. The authors implemented the FLIC prototype using TensorFlow, allowing clients to gather packets, perform on-device training, and exchange training results with the FLIC server. They demonstrated that federated learning-based packet classification achieves an accuracy of 88% under non-independent and identically distributed (non-IID) traffic across clients. When a new application was added dynamically as a client participating in the learning process, an accuracy of 92% was achieved.

The study [27] focuses on the security challenges posed by the deployment of IoT devices and proposes a federated learning-based edge device identification (FedeEDI) method to control access and manage internal devices. The authors note that external attackers often exploit vulnerable IoT devices to gain access to the target's internal network and cause security threats. They review existing literature on deep learning-based algorithms for edge device identification and highlight the limitations of centralized learning-based EDI (CentEDI) methods that train all data together. The authors propose a federated learning-based approach that addresses data security concerns and is suitable for deployment on edge devices.

Overall, the studies presented in this section highlight the potential of federated learning as a viable solution to the challenges posed by the deployment of IoT devices.

## III. THE HORIZONTAL FEDERATED LEARNING

### *A. Horizontal FL*

Federated learning can be divided into three categories: horizontal federated learning, vertical federated learning, and federated transfer learning [28].

Horizontal federated learning is designed for scenarios where participating clients' datasets share the same feature space but contain different samples. The term "horizontal" originates from instance-distributed learning, as illustrated in Fig. 1a, where datasets are horizontally partitioned across data samples and allocated to clients. In federated learning, data can be considered horizontally partitioned when different clients generate data with the same attributes (features) but distinct samples. Similarly, as indicated by the part surrounded by the two dashed lines in Fig. 1b, the data can be considered horizontally partitioned in federated learning when different data are generated on different clients that have the same attributes (features). For example, two hospitals in different regions may have distinct patients but perform the same tests and collect the same personal information (e.g., name, age, gender, and address). There are three main differences between instance-distributed learning and horizontal federated learning [29]:

- Data are typically independent and identically distributed (IID) in distributed learning but may be non-IID in horizontal federated learning. Designers can manually allocate subsets of client data to be IID in distributed learning to enhance convergence, while in horizontal federated learning, the central server has no access to raw data, which is usually non-IID on different clients link.springer.com.
- Horizontal federated learning involves a large number of connected clients, whereas instance-distributed learning often does not have as many workers. Too many workers may worsen distributed training performance when the total data amount is fixed.
- Global model update mechanisms differ. In instance-distributed learning, a deep neural network synchronously updates the global model once local gradients of mini-batch data are calculated. This approach is not suitable for horizontal federated learning due to communication constraints.

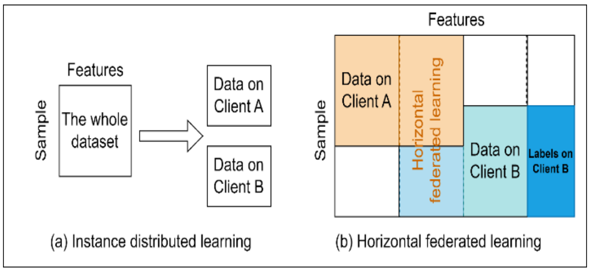


Fig. 1. (a) Instance distributed learning
(b) horizontal federated learning

Horizontal federated learning faces three main challenges compared to centralized learning: reducing communication resources, improving convergence speed, and ensuring no private information leakage.

In contrast, vertical federated learning is applicable when datasets share the same sample space but have different feature spaces. Vertical federated learning is similar to feature distributed learning, where the central server acts as a coordinator to compute the total loss instead of aggregating uploaded weights.

Federated transfer learning, on the other hand, leverages vertical federated learning with a pre-trained model trained on a similar dataset for solving a different problem.

Typical horizontal federated learning (Fig. 2) algorithms, such as the FedAvg, consist of the following main steps:

- Initialize the global model parameters on the server and download the global model to every participating (connected) client.
- Every connected client learns the downloaded global model on its own data for several training epochs. Once completed, the updated model parameters or gradients (gradients here mean the difference between the downloaded model and the updated model) would be sent to the server. Note that the clients may have different amounts of training data and unbalanced computational resources. As a result, the server is not able to receive the uploads from different clients at the same time.
- The server aggregates the received uploads (synchronously or asynchronously) to update the global model.
- Repeat the above two steps until convergence.

From the above steps, we can find that the central server can only receive model weights or gradients of the participating clients and has no access to any local raw data. Therefore, users' privacy is immensely protected in horizontal federated learning.

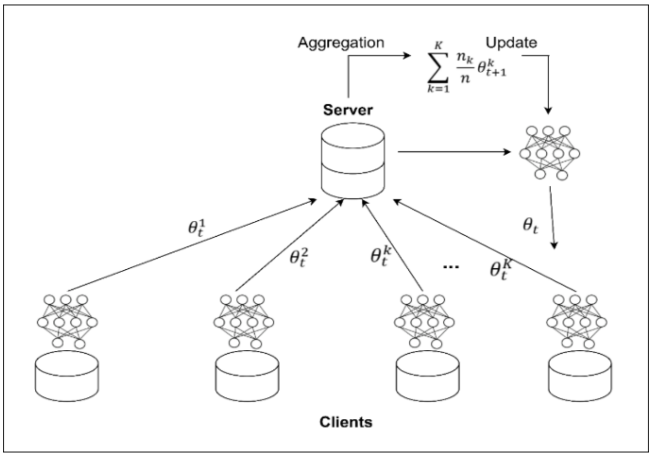


Fig. 2. Flowchart of federated learning. $\theta$ is the global model parameters, $n_k$ is the data size of client $k$, $K$ is the total number of clients and $t$ is the communication round in federated learning. We initialize global model parameters randomly at the beginning of the communication round and use updated model parameters afterward

In this study, the use of this model is suitable the fact that we use a single dataset, which is N-BaIoT. Thus the features are the same.

The Fig 3 displays the approach used in this study, which consists of the dataset, data processing, data aggregation, divide "attribute class", and classifier. Standardization and minimum-maximum normalization min-max normalization) are also used as data preparation techniques.

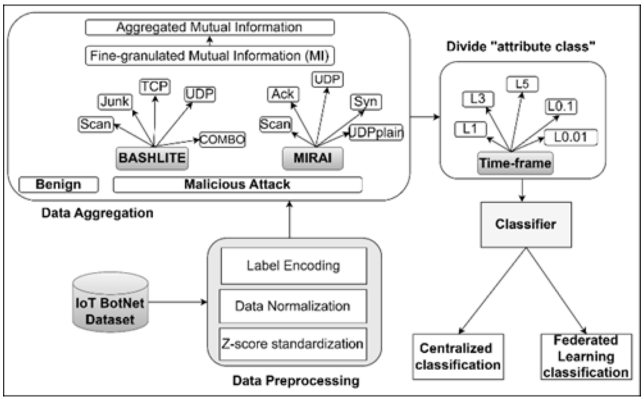


Fig. 3. The research flow

## B. Dataset

There are many public datasets related to IoT security as follows:

- N-BaIoT [1]: This dataset tackles the shortage of available botnet datasets, particularly for the Internet of Things. It implies actual traffic data, collected from nine commercial IoT devices that were actually infected with Bashlite and Mirai malware.
- MedBIoT [2]: It is collected from actual and simulated Internet of Things devices in a medium-sized network (i.e., 83 devices). The data collection is divided into categories based on the kind of traffic (malware or normal traffic), making it simple to label the data and extract characteristics from the raw pcap files.
- Bot-IoT [3]: The BoT-IoT dataset was developed using a realistic network environment constructed at the Cyber Range Lab of the UNSW Canberra Cyber Center. The dataset consists of Service Scan, DDoS, DoS, Keylogging, and Data exfiltration attacks. In addition, DDoS and DoS attacks are categorized based on the protocol employed.
- TON-IoT [4]: This dataset contains heterogeneous data sources gathered from Telemetry datasets of IoT and IIoT sensors, Windows OS, Ubuntu, and network traffic datasets. The datasets were acquired from the Cyber Range and IoT Labs' realistic and wide network.
- IoT-23 [5]: IoT-23 contains 20 malware captures (scenarios) conducted on IoT devices, and 3 benign IoT traffic captures. The authors ran a particular malware sample on a Raspberry Pi for each malicious scenario, which utilized multiple protocols and carried out distinct tasks.

The N-BaIoT dataset is collected separately for each device, so it is very suitable to implement the FL empirical model. Therefore, we will use it for our research. IoT devices in the N-BaIoT dataset were targeted by the Bashlite and Mirai botnet attack families. Each file has 115 features as well as a class label. The dataset has also been built to support binary and multi-class classification. The target class labels are: "TCP attack," "benign" for detection, and "Mirai" or "Bashlite" attack types for multi-class classification.

The N-BaIoT dataset comprises feature headers that describe various aspects of network traffic data, which can be grouped into categories based on their purpose and scope. The headers related to stream aggregation include H, which provides statistics summarizing recent traffic from the packet's host (IP); HH, which summarizes recent traffic from the packet's host to the packet's destination host; HpHp, which summarizes recent traffic from the packet's host+port (IP) to the packet's destination host+port; and HH_jit, which summarizes the jitter of the traffic from the packet's host to the packet's destination host. Additionally, the time-frame decay factor Lambda determines how much recent history of the stream is captured in these statistics, with values such as L5, L3, L1, and others.

The dataset also provides statistics extracted from the packet stream, including the weight of the stream, which can be viewed as the number of items observed in recent history. The mean represents the average value of the stream, while the standard deviation is denoted by std. The root squared sum of the two streams' variances is represented by radius, while the root squared sum of the two streams' means is denoted by magnitude. Moreover, an approximated covariance between two streams is represented by cov, and an approximated covariance between two streams is denoted by *pcc*.

Overall, the N-BaIoT dataset provides a comprehensive set of feature headers that can be used to analyze various aspects of network traffic data, which can be useful for developing and evaluating intrusion detection systems in IoT networks.

In this study, we will focus on multi-classification. We use part of the dataset to proceed with the experiment. The multi-class classification will be evaluated.

[1] https://www.kaggle.com/datasets/mkashifn/nbaiot-dataset
[2] https://cs.taltech.ee/research/data/medbiot/
[3] https://ieee-dataport.org/documents/bot-iot-dataset
[4] https://research.unsw.edu.au/projects/toniot-datasets
[5] https://www.stratosphereips.org/datasets-iot23

### C. *Preprocessing*

Although data preparation [30] is difficult and time-consuming [31], its importance has been demonstrated for speeding the training process and enhancing its effectiveness. Consequently, this study employs label encoding, min–max normalization, and standardization as pre-preprocessing procedures.

#### *1) Label Encoding*

The class label has 11 different category values (one "Benign" class and ten subclasses of attack type). As a result, before these attributes are used with the models, they are converted into numerical values. There are several methods for converting categorical values, including one-hot encoding [32], ordinal encoding [33], similarity encoding [34], entity embedding [35], and multi-hot encoding. Among them, one-hot encoding and ordinal encoding are the most widely employed. This study uses the one-hot encoding technique for encoding categorical values.

#### *2) Normalization and Standardization*

Suppose the columns in a dataset contain values with varying ranges. In that case, the performance of both regression and classification models is negatively impacted. Mahfouz et al. in [36] demonstrated how this issue causes the performance of the models to decrease when uneven scales of features are observed in a dataset. It is required to determine the acceptable range for the insignificant and dominating values to address such problems. The two most used methods are min-max normalizing and z-score standardization:

- The following equation is used to apply min-max normalization to change the values of the dataset's feature values into the range [0, 1]:

$$X_{normalized} = \frac{X - X_{min\_value}}{X_{max\_value} - X_{min\_value}} \quad (1)$$

where $X_{normalized}$ represents the normalized value, $X_{min\_value}$ and $X_{max\_value}$ are the intended interval's boundary range, which is [0, 1], and $X$ is the initial value that would be altered inside those ranges.

- Z-score standardization is used to rescale dataset features, reflecting the characteristics of a normal distribution with mean $\mu = 0$ and standard deviation $\sigma = 1$.

$$X_{normalized} = \frac{X - \mu}{\sigma} \quad (2)$$

#### *3) Data aggregation*

After encoding the target class, the "benign" class was added to the N-BaIoT dataset, which has 115 characteristics and 10 class labels. To improve the performance of the classification process, we can implement some feature selection methods. In the scope of the paper, we choose the Mutual Information (MI) method for the feature selection process of the input data.

An aggregated MI with multiple rank aggregation functions is developed and evaluated for the multi-class dataset. The concept of aggregated MI is explained as follows:

- Calculate the information gain score for each feature, $f_i$, in dataset $D$ relative to class type $c \in C$. The features are then ranked based on the aggregator functions listed below. Calculate the information gain score for each feature, *fi*, in dataset D relative to class type $c \in C$. After that, the features are ranked based on the aggregation methods (Min, Max, Mean). Only part of preserved features are supplied to the classifiers, and the total performance is assessed.
- List of aggregators:
  - ○ *Min:* Chooses the class type $c_i$ as the target class and takes the relevance score with the lowest value.
  - ○ *Max:* Chooses the class type $c_i$ as the target class and takes the relevance score with the highest value.
  - ○ *Mean:* Chooses the class type $c_i$ as the target class and takes the relevance score with the mean value.

#### *4) Dividing attribute class*

In this research study, we aim to evaluate the classification performance of the N-BaIoT dataset by dividing its attributes into different classes based on the time-frame property. The dataset includes time-frames such as L5, L3, L1, L0.1, and L0.01, which correspond to the traffic capture time. We will utilize these time frames to create attribute sets for each subclass. Specifically, Table I shows the attribute set for the time frame L5, which contains 23 attributes. We will follow the same procedure for the remaining time frames. This approach allows us to assess the classification performance of each class containing different attributes, providing insights into the effectiveness of the dataset's features for anomaly detection in IoT networks.

TABLE I
SOME PROPERTIES OF THE TIME-FRAME (L5)

| No | Attribute | No | Attribute |
|---|---|---|---|
| 1 | MI_dir_L5_weight | 13 | HH_L5_pcc |
| 2 | MI_dir_L5_mean | 14 | HH_jit_L5_weight |
| 3 | MI_dir_L5_variance | 15 | HH_jit_L5_mean |
| 4 | H_L5_weight | 16 | HH_jit_L5_variance |
| 5 | H_L5_mean | 17 | HpHp_L5_weight |
| 6 | H_L5_variance | 18 | HpHp_L5_mean |
| 7 | HH_L5_weight | 19 | HpHp_L5_std |
| 8 | HH_L5_mean | 20 | HpHp_L5_magnitude |
| 9 | HH_L5_std | 21 | HpHp_L5_radius |
| 10 | HH_L5_magnitude | 22 | HpHp_L5_covariance |
| 11 | HH_L5_radius | 23 | HpHp_L5_pcc |
| 12 | HH_L5_covariance | | |

## *D. Deep Learning Techniques Applied to Abnormal Traffics Detection*

### *1) Convolutional Neural Networks (CNN)*

Convolutional Neural Networks (CNN) [37]: CNNs were initially developed for image recognition tasks, but researchers have adapted them to process textual data in phishing detection with great success. CNNs utilize convolutional layers to automatically learn features and patterns from the email content by applying multiple filters to different regions of the input text. These filters can capture local patterns, such as word groupings or specific textual structures, which can indicate phishing attempts.

The formula for CNN classification varies depending on the specific architecture and design of the CNN model. However, a general formula for CNN classification can be broken down into the following steps:

- Convolutional Layers: The data input is passed through one or more convolutional layers. Each convolutional layer applies a set of filters to the data input, creating feature maps that capture different aspects of the data.
- Activation Function: After each convolutional layer, an activation function is applied to introduce non-linearity to the output feature maps. The most commonly used activation function is ReLU (Rectified Linear Unit).
- Pooling Layers: After the activation function, the feature maps are passed through one or more pooling layers. Pooling layers downsample the feature maps by taking the maximum or average value within a specific window size. This reduces the size of the feature maps and makes the model more computationally efficient.
- Flatten: After the final pooling layer, the feature maps are flattened into a one-dimensional vector.
- Fully Connected Layers: The flattened feature vector is then passed through one or more fully connected layers. Each fully connected layer applies a set of weights to the input vector and outputs a new vector of a specified size.
- Softmax: The final fully connected layer uses the softmax function to convert the output vector into a probability distribution over the different classes in the classification task. The class with the highest probability is then selected as the predicted class.

The above steps can be represented as a mathematical formula for a basic CNN classification model as follows:

$$y = sm(W_2 * relu\left(W_1 * pool\left(conv(x)\right) + b_1\right) + b_2) \qquad (3)$$

where $x$ is the data input, conv represents the convolutional layers, the pool represents the pooling layers, $W_1$ and $b_1$ represent the weights and biases of the first fully connected layer, RELU represents the activation function, $W_2$ and $b_2$ represent the weights and biases of the final fully connected layer, and *sm* is softmax function, softmax function represents the final activation function that outputs the probability distribution over the different classes.

### *2) Long Short-Term Memory (LSTM)*

LSTM [38] is a recurrent neural network (RNN) that uses feedback to remember portions of the input and make predictions. RNNs are intended to process sequential input and have thus found widespread use in speech recognition and machine translation. The well-known problem of vanishing gradients affects the long-term memory of conventional RNNs. It restricts their ability to make predictions based on the most recent data in the sequence. LSTM (Fig. 4) overcomes the problem of vanishing gradients and can thus handle longer sequences (long-term memory). LSTM can extract context from a succession of features. Using a gate function, it can add or delete information from the hidden state vector, preserving vital information in the hidden layer vectors.

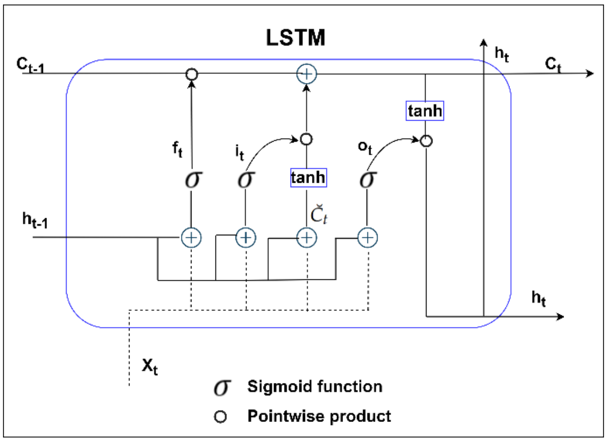


Fig. 4. LSTM Model

There are three gate functions in LSTM: the forget, the input, and the output. The forget gate is used to regulate how much information from $C_{t-1}$ is maintained throughout the computation of $C_t$, and $i_t$ the forget vector may be written as follows:

$$f_t = \sigma\left(U^f x_t + W^f h_{t-1} + b_f\right) \qquad (4)$$

where $U^f$, $W^f$, and $b_f$ are the forget gate's parameters, $x_t$ is the input vector in step $t$, and $h_{t-1}$ is the hidden state vector in the previous step. The input gate determines the amount of $x_t$ information added to $C_t$ and may be stated as follows:

$$f_t = \sigma\left(U^f x_t + W^f h_{t-1} + b_f\right) \qquad (5)$$

where $U^i$, $W^i$, and $b_i$ are the input gate's parameters and $C_t$ may be determined by using both the input gate's vector $i_t$ and the forget gate's vector $f_t$ as shown below:

$$f_t = \sigma\left(U^f x_t + W^f h_{t-1} + b_f\right) \qquad (6)$$

Where $\breve{C}_t = \tanh(U^c x_t + W^c h_{t-1} + b_C)$ reflected the information represented by the vector of the hidden layer. Note that * represents the Hadamard product (element-wise). The output gate regulates the output in $C_t$, and:

$$o_t = \sigma(U^o x_t + W^o h_{t-1} + b_o), h_t = o_t * \tanh(C_t) \qquad (7)$$

where $U^o$, $W^o$, and $b_o$ are the output gate parameters and $C_t$ is the internal state at time step t.

### 3) *Gated Recurrent Unit (GRU)*

A GRU model [39] (Fig. 5) is also an RNN and LSTM variant. However, GRU has only two gates: the update and the reset gates. Due to its simplicity and ease of training, GRU is superior to LSTM, in which data transmitted to the output is decided by the gates, which are two vectors. The update gate helps the model to calculate how much information from the past must be transmitted to the future. For step *t*, the update gate $Z_t$ is computed using the following formula:

$$z_t = \sigma(U^z x_t + W^z h_{t-1}) \quad (8)$$

where $U^z$ , $W^z$ are the weights of the update gate and $h_{t-1}$ stores data for the previous *(t – 1)* units. The reset gate is used to determine how much of the past data to forget, which is computed using the following formulas:

$$r_t = \sigma(U^r x_t + W^r h_{t-1}) \quad (9)$$

where $U^r$ , $W^r$ are the reset gate's weights and $h_{t-1}$ holds information for the previous *(t – 1)* units. Using the reset gate, the current memory content will preserve the essential information from the past:

$$c_t = \tanh(U^c x_t + r_t * W^c h_{t-1}) \quad (10)$$

where $U^c$ , $W^c$ are the weights. Note that ∗ represents the Hadamard product (elementwise). The final step involves calculating the vector $h_t$ that contains the information for the current unit and transmits it along the network:

$$h_t = z_t * h_{t-1} + (1 - z_t) * c_t \quad (11)$$

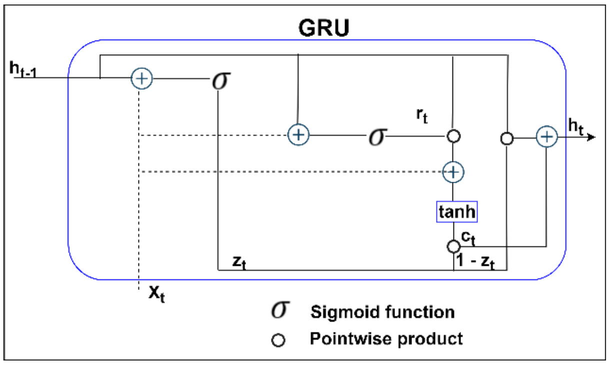


Fig. 5. GRU Model

A GRU network acquires a long spatial (or temporal) sequence with less computational complexity than a conventional encoder-decoder. GRU can handle longer sequences than regular RNNs because it can solve the vanishing gradient problem with gating techniques. To build context between features spread out across a large area or assess the degree of interconnections between features, GRU and LSTM may be used for sequences of spatial data.

### 4) *Experiment Flow for Centralized Approach*

We should recall that we use the N-BaIoT dataset, in which abnormal traffic is caused by malware, especially Botnets. Therefore, here we use "Botnet" as a general concept to indicate the source of abnormal traffic.

The Fig 6 shows the operational flow of the experimental process of Botnet detection in the centralized approach. Data is centralized in one place. First, we perform data processing and feature selection from the dataset data to process the data and select the attributes that are good enough to include in the training model. The dataset is divided into 70% for training, 10% for validation, and 20% for testing.

After that, we will apply three models, CNN, LSTM, and GRU, to detect abnormal traffic generated by Botnet.

The results are obtained, evaluated, and compared between these models.

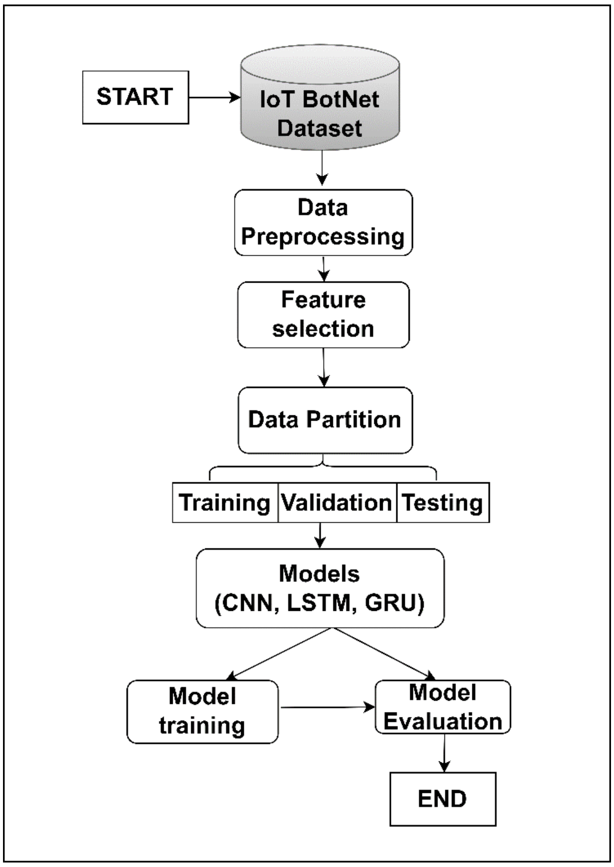


Fig. 6. Experiment Flow for Centralized Approach

## E. *Federated learning Model*

This section describes the architecture of the FL approach. We will present the components and how they interact during the training and testing of models. In addition, it illustrates how the framework is implemented for our validation use case, which uses the N-BaIoT dataset. The Fig 7 depicts the framework architecture, which consists of K clients, each of which owns data from a single device and a server that coordinates the FL process. And the Fig 8 shows the back-and-forth interaction between the client-server and the preprocessing and feature selection process.

### 1) *Model training component*

Two FL techniques derived from the well-known FedAVG, Mini-batch aggregation, and Multi-epoch aggregation, are considered. The primary distinction between FedAVG and the other algorithms is that FedAVG considers the aggregation function as a parameter. As a result, the server can experiment with aggregating methods other than average. In this paper, we will implement the multi-epoch aggregation method. The model is trained for all E epochs simultaneously before being transmitted to the server for multi-epoch aggregation.

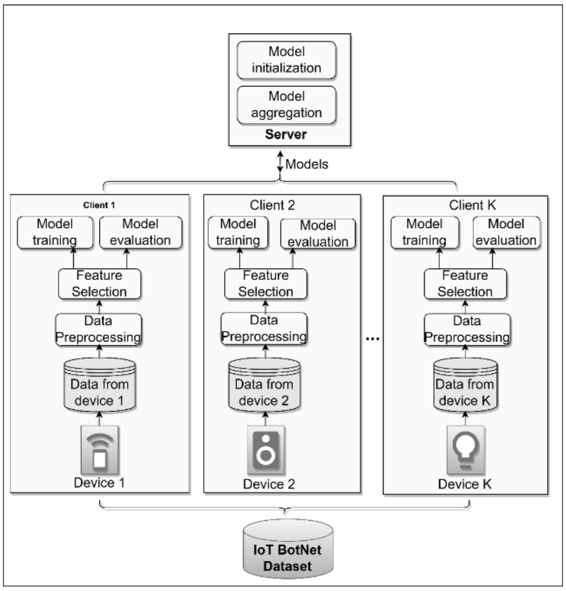


Fig. 7. FL framework architecture and its components

According to [20], averaging models may produce arbitrarily poor results due to the objective's non-convexity. This issue is significantly more likely to arise with multi-epoch aggregation since the models are trained individually for a much more extended time before being combined. To minimize this issue, multi-epoch aggregation training is performed for T = 30 rounds with a decreasing learning rate.

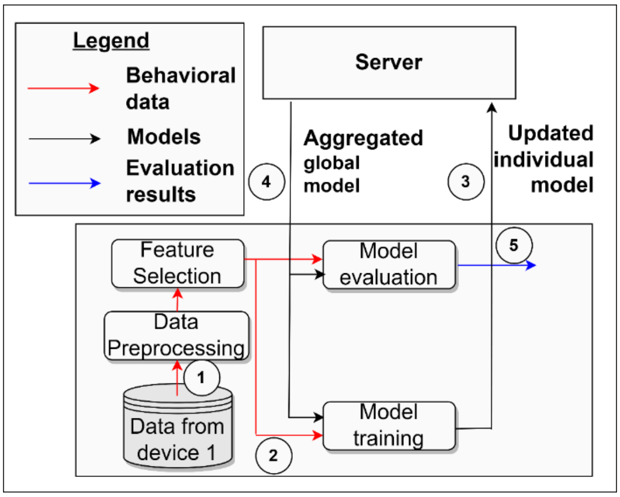


Fig. 8. Detailed view of the training process in the FL

### 2) Server component

The server is responsible for organizing the clients' training activities. In particular, it initializes the model from scratch. It compiles the models that the clients send into a so-called global model. The server is responsible for initializing the initial model's weights. All clients will have access to the first model when it is finished, and training may begin. It is important to note that every client starts with the same model. The server must combine the updated parameters from each client after receiving them to create the new global model parameters.

### F. Evaluations

Using the N-BaIoT dataset, this experiment aims to evaluate how well our approach discovers abnormal IoT traffic caused by malware. It is crucial to compare the federated learning technique with conventional solutions to ensure that it is appropriate for IoT networks.

This research study evaluates the effectiveness of three well-known models, CNN, LSTM, and GRU, using three different approaches for anomaly detection in IoT networks. The first approach is a centralized approach that uses all attributes of the dataset. The second approach is Federated Learning with all attributes, while the third approach is Federated Learning with the division of attributes based on the time-frame characteristics of the dataset. The time-frame characteristics divide the attributes into five groups, corresponding to L1, L3, L5, L0.1, and L0.01, based on the time intervals of the collected streams. The evaluation provides insights into the performance of the different models and training approaches for anomaly detection in IoT networks, which can inform the development of more accurate and efficient intrusion detection systems.

The Fig 9 shows the accuracy of the centralized model based on all attributes of the dataset. The CNN model achieves the highest accuracy of 90.9%, while the LSTM and GRU models reach their highest accuracies of 88.39% and 90.66%, respectively. However, it is important to note that the LSTM and GRU models have more variability in their accuracies across epochs than the CNN model. For example, the LSTM model's accuracy fluctuates between 46.08% and 88.39%, while the GRU model's accuracy ranges from 56.5% to 90.9%. Furthermore, the accuracy of the LSTM model drops significantly after epoch 14, whereas the CNN and GRU models continue to perform well. The GRU model also shows a significant improvement in accuracy from epoch 1 to epoch 3, indicating that it may require more epochs to converge. Overall, the results suggest that the CNN model is the most reliable and consistent performer, while the LSTM and GRU models may require further optimization and fine-tuning to achieve optimal accuracy.

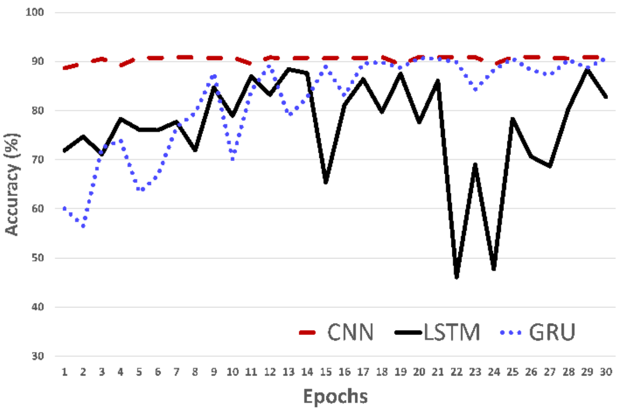


Fig. 9. The performance of the centralized learning model based on all attributes of the dataset

The Fig 10 shows the loss value when training the model with centralized data corresponding to CNN, GRU, and LSTM algorithms. It is interesting to note that the loss values of the LSTM and GRU models are much higher than those of the CNN model, especially in the earlier epochs. This could be due to the fact that the LSTM and GRU models are more complex than the CNN model, and therefore require more training epochs to converge to a similar level of accuracy. It is also worth mentioning that the loss values of the LSTM model show significant fluctuations throughout the training process, with some epochs having much higher losses than others. This could be an indication that the LSTM model is more sensitive to the specific data samples used for each epoch of training. Overall, the CNN model achieves the lowest loss values consistently across all epochs, followed by the LSTM model and then the GRU model. This suggests that the CNN model is the most effective at reducing the error between predicted and actual values during training.

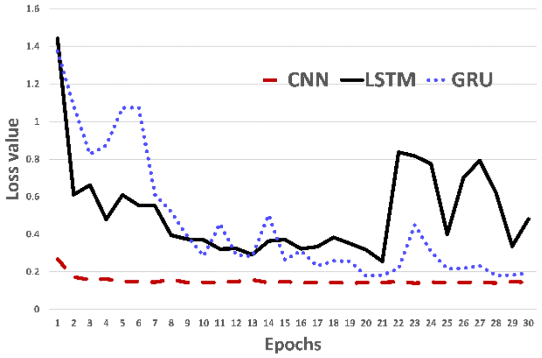


Fig. 10. Loss value of centralized learning model based on all attributes of the dataset

The Federated Learning approach's results of model training based on all attributes of the dataset with distributed data on each client are shown in Fig 11.

In Fig. 11, with the FL approach, the CNN model performs the best with the highest accuracy scores in most of the epochs. It starts with an accuracy of 85.017% in the first epoch and reaches up to 90.831% in the 25th epoch before slightly decreasing in the last few epochs.

On the other hand, the LSTM model has significantly lower accuracy scores compared to the CNN model in most of the epochs. It starts with an accuracy of 33.997% in the first epoch and gradually improves before reaching its peak at 88.803% in the 29$^{th}$ epoch. However, it shows a significant fluctuation in its performance throughout the epochs, with some epochs having a sharp drop in accuracy.

Similarly, the GRU model also has lower accuracy scores compared to the CNN model, but it performs better than the LSTM model in most of the epochs. It starts with an accuracy of 35.42% in the first epoch and gradually improves before reaching its peak at 89.56% in the last epoch. However, like the LSTM model, it also shows significant fluctuation in its performance in some epochs. Overall, the CNN model seems to perform better than the LSTM and GRU models in terms of classification accuracy on this dataset.

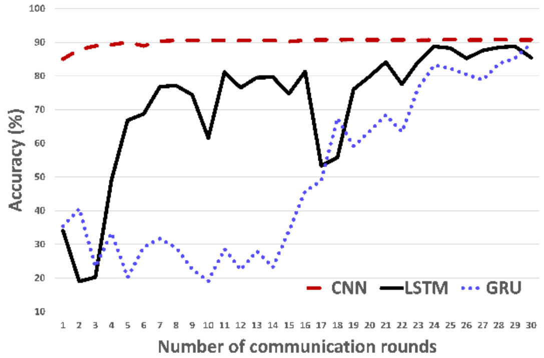


Fig. 11. Classification performance of the FL model based on all attributes of the dataset

The Fig 12 shows the loss value when training distributed data with the FL approach using all attributes. We can see that the loss values of CNN decrease gradually with each epoch, indicating that the model is improving in performance. On the other hand, the loss values for LSTM and GRU are more erratic, with occasional spikes in loss value. This could be due to the more complex nature of these models.

It is also worth noting that the loss values for LSTM and GRU are generally higher than those of CNN, suggesting that CNN is better suited for this particular task. Overall, the results suggest that CNN outperforms LSTM and GRU in terms of loss value.

Observing the accuracy of the models when deployed for centralized and distributed data, we see that the accuracy does not differ too much. The value fluctuates around 1% for accuracy. However, with centralized data, data privacy cannot be guaranteed. So in the case of ensuring data privacy, especially in IoT networks, the Federated Learning model will respond better.

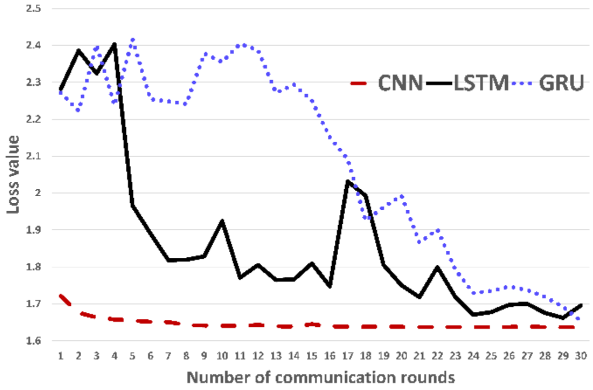


Fig. 12. Loss value of FL model based on all attributes of the dataset

The Fig 13 compares the training times of the centralized and FL-based approaches. The results show that the training time of the model when the data is concentrated is much better. Obviously, the distributed data training adds transmission, computation, and configuration update time for all clients so that the execution time will be longer.

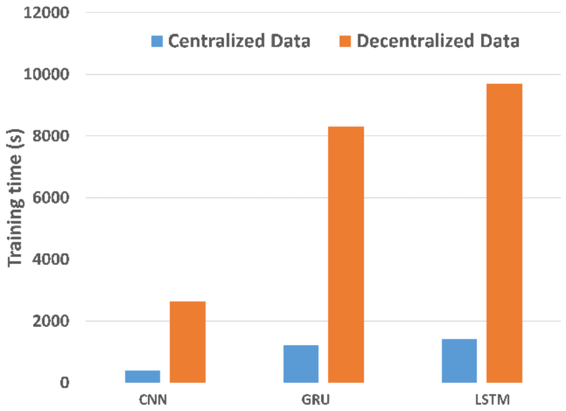


Fig. 13. Time training between centralized and decentralized based on all attributes of the dataset

Table II presents a comparison of three different algorithms, CNN, LSTM, and GRU, based on the metrics of precision, recall, F1-score, and accuracy, for centralized learning based on a time frame of L0.01.

According to the table, the CNN algorithm outperforms the LSTM and GRU algorithms in all four metrics, achieving the highest precision (0.94), recall (0.90), F1-score (0.87), and accuracy (0.90) compared to LSTM and GRU. LSTM achieves the second-best performance with a precision of 0.84, recall of 0.89, F1-score of 0.85, and accuracy of 0.89. GRU, on the other hand, performs the worst among the three algorithms, achieving a precision of 0.78, recall of 0.76, F1-score of 0.73, and accuracy of 0.76. The results suggest that the CNN algorithm is the most suitable for centralized learning based on a time frame of L0.01, as it achieves the highest accuracy and the best balance between precision and recall.

TABLE II
COMPARASION TABLE OF THE ALGORITHMS FOR CENTRALIZED LEARNING BASED ON TIME FRAME (L0.01)

| Metrics | CNN | LSTM | GRU |
|---|---|---|---|
| Precision | 0.94 | 0.84 | 0.78 |
| Recall | 0.90 | 0.89 | 0.76 |
| F1-score | 0.87 | 0.85 | 0.73 |
| Accuracy | 0.90 | 0.89 | 0.76 |

In Fig 14, the classification results using the FL model and based on the set of attributes of the time-frame L0.01 of the dataset, the results show that the CNN model achieves the highest accuracy among the three models with an average of 89.44% across all 30 epochs. LSTM and GRU models achieve significantly lower accuracy compared to CNN, with LSTM averaging 75.89% and GRU averaging 76.92% accuracy across all epochs.

It is also observed that the accuracy of all three models tends to increase over the course of the 30 epochs, with fluctuations in some epochs. However, the overall trend shows an increasing trend, with CNN having the most consistent increase in accuracy over time.

Additionally, it is worth noting that the accuracy of the LSTM and GRU models experiences more fluctuations compared to CNN. This may suggest that CNN is more stable and reliable than LSTM and GRU in this particular classification task.

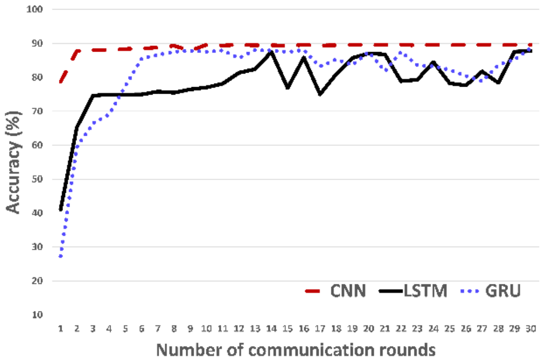


Fig. 14. Classification performance of the FL model based on class attribute L0.01 of the dataset

Table III shows the Precision, Recall, F1-score, and accuracy measures of CNN, LSTM, and GRU algorithms when training centralized data using only L0.1 time-frame algorithms. i.e. use a time frame of 0.1(s) to collect the stream information. The results show that the CNN algorithm gives the best results with an accuracy of 90%, while the LSTM algorithm gives relatively low results, only about 78%. As for the GRU algorithm, the classification result is 88%. Based on the experimental results, using only the attributes of the time frame L0.1 gives quite similar results to using all the attributes of the data set. However, with the L0.1 time-frame attribute set, the LSTM algorithm gives quite low results, while the CNN and GRU algorithms give many good results.

TABLE III
COMPARASION TABLE OF THE ALGORITHMS FOR CENTRALIZED LEARNING BASED ON TIME FRAME (L0. 1)

| Metrics | CNN | LSTM | GRU |
|---|---|---|---|
| Precision | 0.93 | 0.69 | 0.89 |
| Recall | 0.90 | 0.78 | 0.88 |
| F1-score | 0.87 | 0.70 | 0.85 |
| Accuracy | 0.90 | 0.78 | 0.88 |

In Fig 15, the classification results using the FL model and based on the attribute set of the time-frame L0.1 of the data set, the results show that the CNN algorithm gives the best results and reaches the value close to 90 % and converge very quickly starting from round 2. As for the LSTM algorithm, the results fluctuate quite a lot through each round, the learning ability of the LSTM model for this L0.1 time-frame attribute set Not good. Experiments show that using the attribute set of this time-frame L0.1 also gives quite similar results to using all the attributes of the data set.

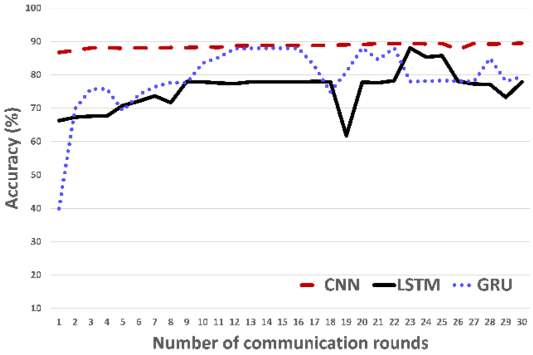


Fig. 15. Classification performance of the FL model based on class attribute L0.1 of the dataset

Table IV presents the evaluation metrics of Precision, Recall, F1-score, and accuracy for the CNN, LSTM, and GRU algorithms when trained on centralized data using L1 time-frame algorithms. L1 refers to a time frame of 1 second to gather the stream's information. According to the findings, the CNN algorithm performed the most accurately, with an accuracy rate of 89%, while the LSTM and GRU algorithms also produced satisfactory results, with accuracy rates of 88% and 86%, respectively. Moreover, these outcomes indicate that utilizing only the L1 time-frame attributes can yield comparable results to those obtained by using the complete set of dataset attributes.

TABLE IV
COMPARASION TABLE OF THE ALGORITHMS FOR CENTRALIZED LEARNING BASED ON TIME FRAME (L1)

| Metrics | CNN | LSTM | GRU |
|---|---|---|---|
| Precision | 0.91 | 0.83 | 0.82 |
| Recall | 0.89 | 0.88 | 0.86 |
| F1-score | 0.85 | 0.85 | 0.82 |
| Accuracy | 0.89 | 0.88 | 0.86 |

In Fig 16, the performance of the Federated Learning (FL) model is evaluated based on the L1 time-frame attributes of the dataset. The results indicate that the CNN algorithm outperforms the LSTM and GRU algorithms, achieving an accuracy of 90% with a fast convergence rate, starting from round 4. In contrast, the GRU algorithm's performance fluctuates considerably throughout each round, demonstrating poor learning ability for the L1 time-frame attribute set. Furthermore, the study suggests that using this L1 time-frame attribute set can yield results similar to those obtained when using all the attributes of the dataset.

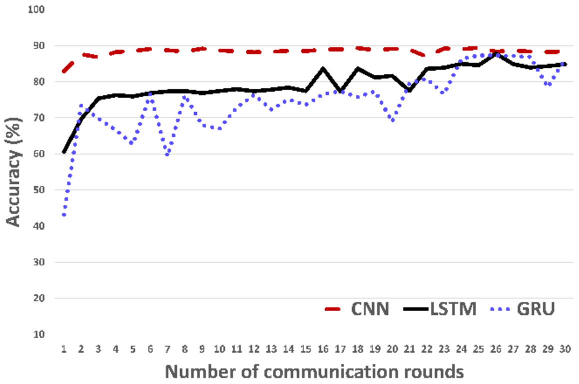


Fig. 16. Classification performance of the FL model based on class attribute L1 of the dataset

Table V presents the evaluation metrics of the CNN, LSTM, and GRU algorithms when trained on centralized data using only the L3 time-frame attribute set, which collects information from a time frame of 3 seconds.

The experimental results indicate that the CNN and LSTM algorithms achieve the highest accuracy of 88%, while the GRU algorithm yields comparatively low results of around 78% when trained on the L3 time-frame attribute set. These findings suggest that using the L3 time-frame attribute set can yield similar results to those achieved with the entire dataset.

TABLE V
COMPARASION TABLE OF THE ALGORITHMS FOR CENTRALIZED LEARNING BASED ON TIME FRAME (L3)

| Metrics | CNN | LSTM | GRU |
|---|---|---|---|
| Precision | 0.90 | 0.83 | 0.73 |
| Recall | 0.88 | 0.88 | 0.78 |
| F1-score | 0.85 | 0.84 | 0.70 |
| Accuracy | 0.88 | 0.88 | 0.78 |

In Fig 17 depicts the classification results obtained using the FL model and the L3 time-frame attribute set of the data set. The results indicate that the CNN algorithm achieves the highest accuracy of 90%, with rapid convergence starting from round 6. Conversely, the GRU algorithm produces fluctuating results throughout each round, indicating poor learning ability for this L3 time-frame attribute set.

Empirical findings indicate that adopting the L3 time-frame attribute set achieves comparable results to using the complete attribute set of the data.

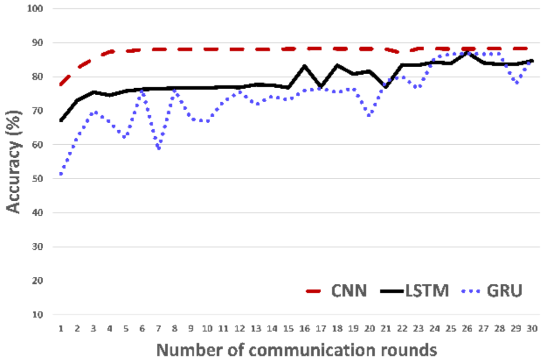


Fig. 17. Classification performance of the FL model based on class attribute L3 of the dataset

Table VI presents the evaluation metrics of the CNN, LSTM, and GRU algorithms when trained on centralized data using only the L5 time-frame attribute set, which collects information from a time frame of 5 seconds.

Based on the evaluation metrics presented in Table VI, the CNN, LSTM, and GRU algorithms were compared when trained on centralized data using only the L5 time-frame attribute set. The results indicate that the CNN algorithm achieved the highest accuracy of 87%, while the LSTM algorithm achieved an accuracy of 82%. However, the GRU algorithm gave relatively lower results with an accuracy of only 77%. It is noteworthy that the experimental results demonstrate that using only the L5 time-frame attributes produces quite similar results to using all the attributes of the data set.

TABLE VI
COMPARASION TABLE OF THE ALGORITHMS FOR CENTRALIZED LEARNING BASED ON TIME FRAME (L5)

| Metrics | CNN | LSTM | GRU |
|---|---|---|---|
| Precision | 0.87 | 0.66 | 0.79 |
| Recall | 0.87 | 0.77 | 0.82 |
| F1-score | 0.83 | 0.70 | 0.79 |
| Accuracy | 0.87 | 0.77 | 0.82 |

The Fig 18 illustrates the classification results obtained by the FL model based on the L5 time-frame attribute set of the data set. The results indicate that the CNN algorithm provides the highest accuracy of 90% and exhibits the fastest convergence rate, starting from round 6. Conversely, the GRU algorithm's results exhibit significant fluctuations across each round, indicating that the model's learning ability for this L5 time-frame attribute set is insufficient. Based on the experimental results, it can be inferred that employing only the L5 time-frame attribute set yields results that are comparable to using all attributes of the dataset.

Based on the experiments conducted, the results indicate that the use of attribute sets based on time-frames yields comparable outcomes to using the entire dataset. Furthermore, a minor variation in accuracy was observed between centralized and distributed data models, fluctuating around 1%. However, the centralized approach is not adequate for ensuring data privacy, which is crucial in IoT networks. Therefore, the Federated Learning model is a more appropriate approach to preserving data privacy.

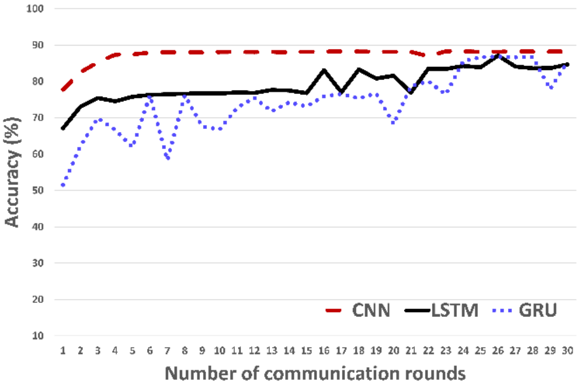


Fig. 18. Classification performance of the FL model based on class attributes L5 of the dataset

## IV. Conclusion

Based on the results obtained from the N-BaIoT dataset, this study recommends using the Federated Learning (FL) approach for detecting IoT malware abnormal traffic while preserving user privacy. Two different methods were compared, i.e., the Federated Learning method, where each device owner trains a separate model, and the Centralized approach, where each device owner trains a single, isolated model using a centralized dataset. Although the Centralized approach still has slightly higher accuracy in detecting abnormal IoT traffic than FL, the difference is insignificant. However, the Centralized approach model fails to ensure the security and privacy of confidential information, unlike the FL model. Furthermore, when comparing the CNN, LSTM, and GRU models applied in the FL approach, CNN yields the best results and is suitable for simple FL models.

In future studies, our research aims to further optimize the Federated Learning model, specifically focusing on the global parameter to enhance its accuracy in detecting abnormal IoT traffic. Additionally, we plan to improve the processing time to achieve more efficient and effective results.

## Acknowledgment

The study was financially supported by the Russian Science Foundation within of scientific project No. 22-49-02023 ”Development and study of methods for obtaining the reliability of tethered high-altitude unmanned telecommunication platforms of a new generation.

## References

[1] Fan, Xiaochen, et al. "BuildSenSys: Reusing building sensing data for traffic prediction with cross-domain learning." *IEEE Transactions on Mobile Computing* 20.6 (2020): 2154-2171.

[2] Alrawi, Omar, et al. "Sok: Security evaluation of home-based iot deployments." *2019 IEEE symposium on security and privacy* (sp). IEEE, 2019.

[3] Kulik, V., & Kirichek, R. (2018, November). The heterogeneous gateways in the industrial internet of things. *In 2018 10th International congress on ultra modern telecommunications and control systems and workshops* (ICUMT) (pp. 1-5). IEEE.

[4] Kumar, J. Sathish, and Dhiren R. Patel. "A survey on internet of things: Security and privacy issues." *International Journal of Computer Applications 90.11* (2014).

[5] Butun, Ismail, Patrik Österberg, and Houbing Song. "Security of the Internet of Things: Vulnerabilities, attacks, and countermeasures." *IEEE Communications Surveys & Tutorials 22.1* (2019): 616-644.

[6] Radhakrishnan, Divya. "Internet of things: Privacy and security issues: A qualitative study about the internet of things and its privacy and security challenges from a developer’s perspective." (2021).

[7] Tsimenidis, S., Lagkas, T., & Rantos, K. (2022). Deep learning in IoT intrusion detection. *Journal of network and systems management*, 30, 1-40.

[8] Aldweesh, A., Derhab, A., & Emam, A. Z. (2020). Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues. *Knowledge-Based Systems*, 189, 105124.

[9] Kairouz, Peter, et al. "Advances and open problems in federated learning." *Foundations and Trends® in Machine Learning* 14.1–2 (2021): 1-210.

[10] Otoum, Safa, Ismaeel Al Ridhawi, and Hussein T. Mouftah. "Blockchain-supported federated learning for trustworthy vehicular networks." *GLOBECOM 2020-2020 IEEE Global Communications Conference*. IEEE, 2020.

[11] Meidan, Yair, et al. "N-baiot—network-based detection of iot botnet attacks using deep autoencoders." *IEEE Pervasive Computing* 17.3 (2018): 12-22.

[12] Arbex, Gustavo Vitral, et al. "IoT DDoS Detection Based on Stream Learning." *2021 12th International Conference on Network of the Future* (NoF). IEEE, 2021.

[13] Ferdowsi, Aidin, and Walid Saad. "Generative adversarial networks for distributed intrusion detection in the internet of things." *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019.

[14] Almiani, Muder, et al. "Deep recurrent neural network for IoT intrusion detection system." *Simulation Modelling Practice and Theory* 101 (2020): 102031.

[15] Xia, Qinyu, Shi Dong, and Tao Peng. "An Abnormal Traffic Detection Method for IoT Devices Based on Federated Learning and Depthwise Separable Convolutional Neural Networks." *2022 IEEE International Performance, Computing, and Communications Conference* (IPCCC). IEEE, 2022.

[16] Sim, Khe Chai, et al. "Domain Adaptation Using Factorized Hidden Layer for Robust Automatic Speech Recognition." *Interspeech*. 2018.

[17] Almiani, Muder, et al. "Deep recurrent neural network for IoT intrusion detection system." *Simulation Modelling Practice and Theory* 101 (2020): 102031.

[18] Vladimirov, Sergey, and Ruslan Kirichek. "The IoT identification procedure based on the degraded flash memory sector." *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*. Springer, Cham, 2017. 66-74.

[19] Lou, Yang, et al. "Predicting network controllability robustness: A convolutional neural network approach." *IEEE Transactions on Cybernetics* 52.5 (2020): 4052-4063.

[20] Amanullah, Mohamed Ahzam, et al. "Deep learning and big data technologies for IoT security." *Computer Communications* 151 (2020): 495-517.

[21] Sim, Khe Chai, et al. "Domain Adaptation Using Factorized Hidden Layer for Robust Automatic Speech Recognition." *Interspeech*. 2018.

[22] Siniosoglou, Ilias, et al. "A unified deep learning anomaly detection and classification approach for smart grid environments." *IEEE Transactions on Network and Service Management* 18.2 (2021): 1137-1151.

[23] Larriva-Novo, Xavier, et al. "An IoT-focused intrusion detection system approach based on preprocessing characterization for cybersecurity datasets." *Sensors 21.2* (2021): 656.

[24] Zhu, Hangyu, Haoyu Zhang, and Yaochu Jin. "From federated learning to federated neural architecture search: a survey." *Complex & Intelligent Systems* 7 (2021): 639-657.

[25] Abbasi, Mahmoud, Amir Taherkordi, and Amin Shahraki. "FLITC: A Novel Federated Learning-Based Method for IoT Traffic Classification." *2022 IEEE International Conference on Smart Computing (SMARTCOMP)*. IEEE, 2022.

[26] Mun, Hyunsu, and Youngseok Lee. "Internet traffic classification with federated learning." *Electronics* 10.1 (2020): 27.

[27] He, Zhimin, et al. "Edge device identification based on federated learning and network traffic feature engineering." *IEEE Transactions on Cognitive Communications and Networking* 8.4 (2021): 1898-1909.

[28] Wen, Jie, et al. "A survey on federated learning: challenges and applications." *International Journal of Machine Learning and Cybernetics* (2022): 1-23.

[29] Zhu, Hangyu, Haoyu Zhang, and Yaochu Jin. "From federated learning to federated neural architecture search: a survey." *Complex & Intelligent Systems* 7 (2021): 639-657.

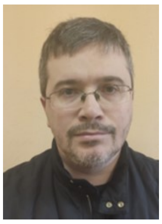
[30] Do, Phuc Hao, et al. "An Efficient Feature Extraction Method for Attack Classification in IoT Networks." *2021 13th International*

*Congress on Ultra Modern Telecommunications and Control Systems and Workshops* (ICUMT). IEEE, 2021.
[31] Cohen, Patricia, Stephen G. West, and Leona S. Aiken. Applied multiple regression/correlation analysis for the behavioral sciences. *Psychology press*, 2014.
[32] Conrod, Patricia J., et al. "Effectiveness of a selective, personality-targeted prevention program for adolescent alcohol use and misuse: a cluster randomized controlled trial." *JAMA psychiatry* 70.3 (2013): 334-342.
[33] Cerda, Patricio, Gaël Varoquaux, and Balázs Kégl. "Similarity encoding for learning with dirty categorical variables." *Machine Learning* 107.8 (2018): 1477-1494.
[34] Guo, Cheng, and Felix Berkhahn. "Entity embeddings of categorical variables." *arXiv preprint* arXiv:1604.06737 (2016).
[35] Zhou, Yuyang, et al. "Building an efficient intrusion detection system based on feature selection and ensemble classifier." *Computer networks* 174 (2020): 107247.
[36] Mahfouz, Ahmed, et al. "Ensemble classifiers for network intrusion detection using a novel network attack dataset." *Future Internet* 12.11 (2020): 180.
[37] Yamashita, Rikiya, et al. "Convolutional neural networks: an overview and application in radiology." *Insights into imaging* 9 (2018): 611-629.
[38] Chung, Junyoung, et al. "Empirical evaluation of gated recurrent neural networks on sequence modeling." *arXiv preprint* arXiv:1412.3555 (2014).
[39] Volkov, A., Khakimov, A., Muthanna, A., Kirichek, R., Vladyko, A., & Koucheryavy, A. (2017, June). Interaction of the IoT traffic generated by a smart city segment with SDN core network. *In International Conference on Wired/Wireless Internet Communication* (pp. 115-126). Springer, Cham.

**Phuc Hao Do** received his MS degree in Computer science from the University of Danang - University of Science and Technology in 2017. He is currently a Ph.D. student in the Department of Communication Networks and Data Transmission at the Bonch-Bruevich Saint- Petersburg State University of Telecommunications, Russia. His research interests include AI, ML, D and its application in different fields like network, blockchain.

**Dr. Tran Duc Le** acquired his degree of Ph.D. at Admiral Makarov State University of Maritime and Inland Shipping, Russia in 2018. He has been working in Information Technology Faculty, The University of Danang - University of Science and Technology, Danang, Vietnam since 2019. His research areas include the Internet of Things, Security Analytics, Malware Analysis.

**Dr. Sc. Vladimir M. Vishnevsky** received the Engineering degree in applied mathematics from the Moscow Institute of Electronics and Mathematics, Russia, in 1971, the Ph.D. degree in queuing theory and telecommunication networks and the D.Sc. degree in telecommunication networks from the V. A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences (ICS RAS), in 1974 and 1988, respectively. He became a Full Professor with ICS RAS in 1989 and the Moscow Institute of Physics and Technology in 1990. He was an Assistant Head of the Institute of Information Transmission Problems of RAS from 1990 to 2010 and an Assistant Head of laboratory with ICS RAS from 1971 to 1990. He is currently the Head of Telecommunication Networks Laboratory, ICS RAS. He is a member of Expert Councils of Russian High Certifying Commission and Russian Foundation for Basic Research, member of IEEE Communication Society, International Telecommunications Academy and New York Academy of Science. He has authored over 300 papers in queuing theory and telecommunications. He is a Co-Chair of IEEE conferences - ICUMT, RTUWO, and the General Chair of DCCN conference. His research interests lie in the areas of computer systems and networks, queuing systems, telecommunications, discrete mathematics (extremal graph theory, mathematical programming) and wireless information transmission networks.



**Dr Aleksandr Berezkin**, is working at the Bonch Bruevich Saint Petersburg State University of Telecommunications as the Associate Professor of Department of Programming Engineering and Computer Science. Science interest are Computer Vision and Machine Learning. In 2009, he defended his thesis with the topic "Model and method of decoding error correction based on neural network". Now he is doctoral student at the Department of Programming Engineering and Computer Science.

**Dr. Sc. Ruslan Kirichek** is working at the Bonch Bruevich Saint Petersburg State University of Telecommunications as the head of Department of Programming Engineering and Computer Science. He was born in 1982 in Tartu (Estonia). He graduated Military-Space Academy A.F. Mozhaiskogo and the Bonch-Bruevich St. Petersburg State University of Telecommunications in 2004 and 2007, respectively. He received Ph.D. at the Bonch-Bruevich St. Petersburg State University of Telecommunications in 2012 and Dr.Sc. at the Povolzhskiy State University of Telecommunications and Informatics in 2018. From 2008 to 2013 he worked as a senior researcher at the Federal State Unitary Enterprise "Center-Inform". Since 2012 he has been working as the Head of the Internet of Things Laboratory at the Bonch-Bruevich Saint Petersburg State University of Telecommunications. Since 2017 he has been working as ITU-T Q12/11 Rapporteur in “Testing of Internet of things, its applications and identification systems”. Since 2023 he has been working as the Rector of the Bonch-Bruevich Saint Petersburg State University . He is a General Chair of the International Conference “Internet of Things and Its Enablers” (inthiten.org).

# A Deep learning Framework for Cultural Heritage Damage Detection for Preservation; Based on the case of Heunginjimun and Yeongnamnu in South Korea

Sang-Yun Lee*, Daekyeom Lee**

*Electronics and Telecommunications Research Institute (ETRI), Daejeon, Republic of Korea

** SEASON Co., Ltd., Sejong, Republic of Korea

syllee@etri.re.kr, daek29@season.co.kr

***Abstract*—In general, there are many restrictions on investigations for safety inspection due to the uniqueness of cultural heritages. Methods such as visual inspection and non-destructive inspection, which are mainly used as inspection methods, are regularly carried out, but there are limitations on time and cost. This is insufficient to identify and respond quickly when an abnormal symptom appears in cultural heritage. As a basic study of system development for rapid abnormal detection of architectural, cultural properties through Deep Learning, this paper organized a Deep Learning framework for detecting tilt in buildings for the roof of Heunginjimun Gate (Korea Treasure No. 1) and Yeongnamnu Pavilion (Korea Treasure No. 147). A framework was developed using a Convolutional Neural Network (CNN). As a result of an application, EfficientnetB0 and EfficientnetB2 models showed excellent accuracy in detecting the tilt of the roof of Heunginjimun with an average accuracy of 99.66% and 99.69%, respectively. In addition, EfficientnetB0, EfficientnetB2, and Shufflenet_v2 models showed excellent accuracy in detecting tilt of the roof of Yeongnamnu with 98.81%, 99.80%, and 98.48% accuracy. Additionally, the Grad-CAM experiment was conducted as a basis for whether the model made the proper judgment to confirm the criteria for determining abnormal detection according to the results of each model. These findings quickly detect abnormalities occurring in cultural heritages from the perspective of cultural heritage management and preservation, enabling rapid response, and are valuable for research on artificial intelligence technology related to cultural heritages.**

***Keyword*—Conventional Neural Network, Cultural Heritage, Grad-CAM, Preservation**

## I. INTRODUCTION

IN recent years, disastrous incidents have occurred during the attempts to preserve cultural assets, such as the tilting of Cheomseongdae in Gyeongju due to the earthquake in Pohang in 2016, the burning of Sungnyemun Gate, the attempted arson of Heunginjimun Gate in 2018, and the collapse of the Gongsanseong Fortress Wall due to torrential rain in July 2020.

According to the data on the status of emergency repair of cultural properties by cause of damage that occurred for 6 years in the data of 'Cultural heritage in statistics (2021, 2022)' published by the Cultural Heritage Administration of South Korea, total 319 cases of damage occurred for 6 years, storm and flood damage (typhoon, strong wind, heavy rain), biological damage (termites, pests), others (collapse, fall, unknown cause, etc.), cold waves, and earthquakes accounted for 168 cases, 48 cases, 36 cases, 19 cases, and 16 cases, respectively [1, 2].

In line with the situation above, the cost of maintenance and repair of cultural properties is also increasing. According to the current status of cultural asset repair and maintenance by the Cultural Heritage Administration of the Republic of Korea, the amount spent on cultural property repair was about 432 billion won in 2018, about 526 billion won in 2019, about 570 billion won in 2020, about 577 billion won in 2021, and about 574 billion won in 2022. It shows a gradually increasing trend [3].

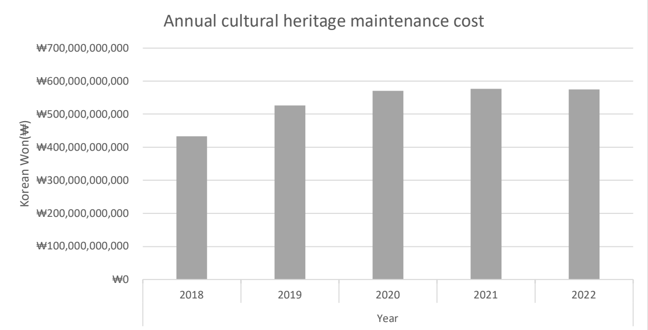


**Fig. 1. Annual cultural heritage maintenance cost in South Korea (Reorganized based on KOSIS, 2022)**

South Korea's cultural heritages are valuable assets that embodies human culture, so it is imperative that they be

Manuscript received Dec. 13, 2022. This paper was funded by the government (Cultural Heritage Administration) in 2023 with the support of the National Research Institute of Cultural Heritage (Grant ID: No.2021A01D06-001), and a follow-up of the invited journal to the accepted & presented paper entitled " Damage Detection and Safety Diagnosis for Immovable Cultural Assets Using Deep Learning Framework" of the 25th International Conference on Advanced Communication Technology (ICACT2023).

Sang-Yun Lee, he is a Ph. D. and working for Electronics and Telecommunications Research Institute (ETRI) as a principal researcher in Daejeon, 34129, Republic of Korea (corresponding author to provide phone: +82-10-2995-0928; E-mail: syllee@etri.re.kr)

DaeKyeom Lee, he is working ofr Season Co., Ltd in Sejong-si, 30127, Republic of Korea. He is now leading the AI Convergence Technology Research Team. (phone: +82-10-3287-9100; E-mail: daek29@season.co.kr)

handed down to future generations. Therefore, regular inspection and maintenance work is required to sufficiently maintain the structural stability of aging cultural properties due to issues such as climate change, Artificial activity, and deterioration. In addition, since damage to cultural heritage is irreversible once incited, efforts should be made to prevent them from being destroyed or damaged. However, due to the unique qualities of cultural heritage, there are many restrictions on safety inspections, so inspection is usually conducted by methods such as visual inspection and other non-destructive methods. Methods such as visual inspection and non-destructive inspection are usually performed on a regular basis, but there are limitations on time and cost. In addition, when abnormal symptoms appear in cultural properties, identifying them and taking immediate initial responses is not enough. In accordance to this issue, attempts are being made to combine cultural heritage conservation and artificial intelligence technology to respond to cultural heritage damage [4-14]. As a representative case, Mishra et al implemented a case study on Dadi - Poti tom in New Deli Hauz Khas Village. They developed a You Only Look Once (YOLOv5) real-time object detection algorithm and ResNet 101-based R-CNN through customized defect detection and localization. A study was conducted using the model to detect four types of defects: discoloration, brick exposure, cracks, and delamination [4]. Mansuri and Patel used the R-CNN model to build an automatic visual inspection system in Surat, India, and British and Dutch cemeteries as well have found 'breakthroughs', 'exposed brick', 'cracks' spalling exposed bricks, and cracks existing. A study was conducted to detect these three types of defects in heritage structures [5]. Yu et all generated data and used deep networks to carry out the Dunhuang cultural heritage protection project known as Thousand Buddha Caves [7]. The papers above performed machine learning mainly on stone structures, unlike this paper, which is for wooden buildings.

In this paper, Deep Learning video data taken by CCTV of the roof of Heunginjimun (Treasure No. 1), and Yeongnamnu (Treasure No. 147), were used for Deep Learning. In the case of Heunginjimun, it is the only one among the gates of the capital city surrounding Hanyang, the capital of the Joseon Dynasty, that maintains the form of Onseong (a double wall surrounding the gate to protect it). Like Sungnyemun, which is National Treasure No. 1, the gatehouse has multiple levels (double-layer), and like Gyeongbokgung Palace, the existence of Jabsang (Small figures on the roof. People believed that they protected the building from evil spirits, especially fire spirits.) on the roof indicates that the building holds significant historical value [15,16]. Also, in the case of Yeongnamnu, it is a pavilion-style building that is a representative form of South Korean traditional architecture. It is also a representative wooden structure of the late Joseon Dynasty. It is a cultural property with high preservation value, as it is being called one of the three significant pavilions in Korea [17,18,19]. In addition, as a preceding study, [20] proposed a Deep Learning Framework that can detect fine gradients for Heunginjimun. However, unlike this paper, since the experiment was performed with only a single subject of Heunginjimun, this paper additionally included Yeongnamru as a subject, and confirmed the results with Grad-CAM to verify that Deep Learning is performed well.

By using the data from the aforementioned cultural heritages, we will identify an optimal Deep Learning model and propose a building abnormality detection system by constructing a framework. The Deep Learning framework developed in this study is believed to be of great help for the preservation and transmission of cultural assets. The structure of this thesis is as follows. In Chapter 2, the data set composition and preprocessing process for generating abnormal data were explained, and in Chapter 3, the methodology used in this experiment and the parameter setting of the used model were described. In Chapter 4, the experimental results of each model and Grad-CAM results for verify were presented, and finally, the conclusion is made in Chapter 5.

## II. DATA COLLECTION AND PREPROCESSING

### *A. The composition of the dataset*

In order to construct a dataset for use in the damage detection Deep Learning framework of cultural heritage, images of cultural heritage filmed through CCTV were used. CCTV images installed at the front of Heunginjimun and Yeongnamnu were used, and the data recorded by CCTV are converted into AVI video format in IRAS (IDIS CCTV monitoring SW). The converted AVI video was divided into 600 frame units, saved as JPG pictures, and input to the pre-trained model. The dataset constructed using CCTV images as above was divided into five categories: Clear-Day, Clear-Night, Rainy-Day, Rainy-Night, and Cloudy-Day by environment. The composition of each environment dataset consists of 10000 pieces of data and consists of 5000 pieces of the training set, 2500 pieces of the validation set, and 2500 pieces of the test set. The ratio of the experiment's train set, validation set, and test set was 50% 25% 25%. Table. 1 shows the composition and ratio of each data set of Heunginjimun and Yeongnamnu.

TABLE I
THE COMPOSITION AND RATIO OF THE DATASET

| Heunginjimun | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Dataset | Normal/Abnormal (Ratio) | Train/Validation/ Test (Quantity) | Train set (Quantity) | | Validation set (Quantity) | | Test set (Quantity) | |
| | | | Normal | Abnormal | Normal | Abnormal | Normal | Abnormal |
| Clear-Day | 5:5 | 5000/2500/2500 | 2500 | 2500 | 1250 | 1250 | 1250 | 1250 |
| Clear-Night | | | | | | | | |
| Rainy-Day | | | | | | | | |
| Rainy-Night | | | | | | | | |
| Cloudy-Day | | | | | | | | |

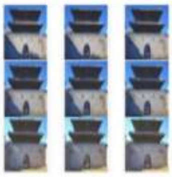
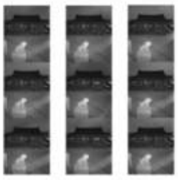
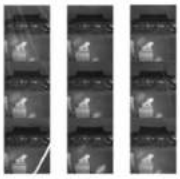
| Yeongnamnu | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Dataset | Normal/Abnormal (Ratio) | Train/Validation/ Test (Quantity) | Train set (Quantity) | | Validation set (Quantity) | | Test set (Quantity) | |
| | | | Normal | Abnormal | Normal | Abnormal | Normal | Abnormal |
| Clear-Day | 5:5 | 5000/2500/2500 | 2500 | 2500 | 1250 | 1250 | 1250 | 1250 |
| Clear-Night | | | | | | | | |
| Rainy-Day | | | | | | | | |
| Rainy-Night | | | | | | | | |
| Cloudy-Day | | | | | | | | |

### *B. Configuring the data set's Environments*

In order to predict the tilt of the roof part, prediction experiments were conducted in various environments. We tried to reflect on the most common environments according to our data. The roof of Heunginjimun was divided into five environments: Clear-Day, Clear-Night, Rainy-Day, Rainy-Night, and Cloudy-Day. Also, it was divided into Day and

Night based on the time when CCTV converted to night camera. Yeongnamnu also divided the environment into same. Table. 2 shows examples of data for each environment.

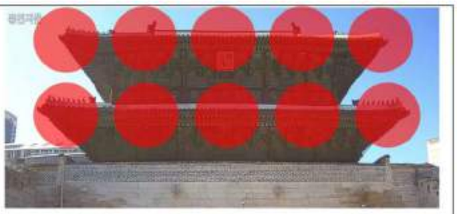
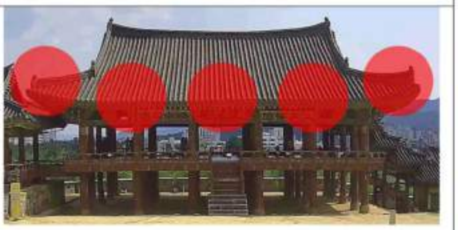
TABLE II
EXAMPLES OF DATASETS BY ENVIRONMENT

| Data Example | | |
|---|---|---|
| Environment | Heunginjimun | Yeongnamnu |
| Clear-Day |  | |
| Clear-Night | |  |
| Rainy-Day | | |
| Rainy-Night | |  |
| Cloudy-Day | | |

### C. Generation of Abnormal data

As with the normal data we needed for Deep Learning, an example of abnormal data was needed to explore the abnormal state of architecture. However, once a cultural heritage is damaged, it is often difficult to restore it to its original state. Even if it can be restored, it takes a tremendous amount of money and time, so collecting enough abnormal data for machine learning is very difficult. Therefore, as an alternative method, in the case of abnormal data, the Adobe Photoshop action function was used to generate abnormal data for each scenario in each environment by stretching the eaves. In the case of the roof of Heunginjimun, it is composed of two layers, and displacements were applied to a total of 10 areas. The area where the displacement was applied is marked by a red circle. Unlike Heunginjimun, the roof of Yeongnamnu is composed of a single layer, and displacements were applied to five areas accordingly. As with Heunginjimun, the area marked with a red circle is the area where the displacement was used. Table. 3 shows the displacement points of Heunginjimun and Yeongnamnu.

TABLE III
DISPLACEMENT POINTS FOR GENERATING ABNORMAL DATA

| | |
|---|---|
| Heunginjimun |  |
| Yeongnamnu |  |

The samples of the abnormal data set generated by applying displacement to the points of 10 areas of the roof of Heunginjimun and 5 areas of the roof of Yeongnamnu are shown in Fig. 2 and Fig. 3 below. When checking the abnormal data compared to the original, it can be seen that the tilt of the displaced area (the red rectangle in Fig. 2 and Fig. 3) is slightly different.

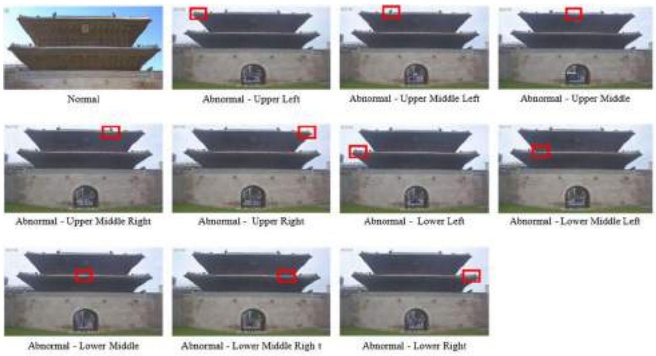


**Fig. 2. Samples of abnormal images of Heunginjimun**

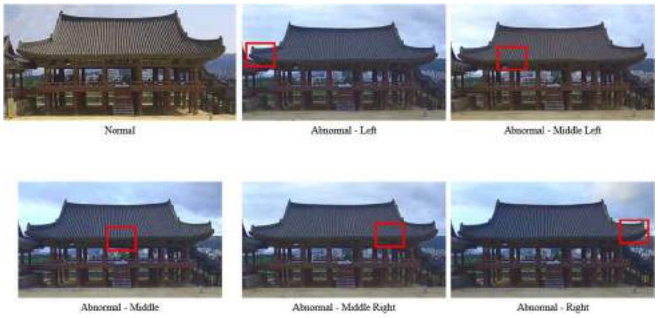


**Fig. 3. Samples of abnormal images of Yeongnamnu**

## III. METHODOLOGY

### A. Convolutional Neural Network (CNN)

Convolution Neural Network (CNN) is a technique announced by LeCun and Bengio in 1995 [21]. It is an algorithm that shows excellent performance in image recognition or voice recognition among Deep Learning methods, and its structure is designed according to neurobiological principles [22-25]. In the case of CNN, it has the ability to analyze the specific characteristics of the object in question regardless of the location and the direction in which it is placed. CNN consists of a convolution layer that extracts features from an input image and a pooling layer that compresses the extracted information. Convolution and pooling are repeated to extract the features [25]. Furthermore, CNN uses a filter in the convolution step to extract the features of an image. As shown in Fig. 4, the Filter composed of matrices in the Image Matrix scans the image from top left to bottom right.

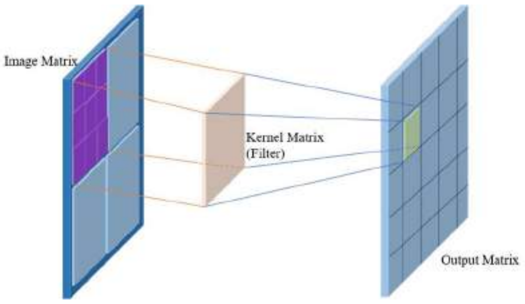


**Fig. 4. Feature extraction in convolution**

Subsequently, after going through the feature extraction step in which convolution and pooling are repeated, the classification step is passed as shown in Fig.5. The feature matrix extracted in the flattening step is flattened into vectors. Then the sorted vectors go through a fully connected layer and an activation function such as softmax or sigmoid is ordered to output a class and classify the image and create a result

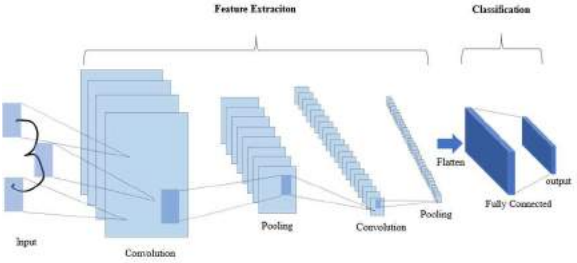


**Fig. 5. Convolutional Neural Network Architecture**

### B. Experimental Environment

This experiment was performed on Ubuntu 20.040.2. and under the LTS environment. The specifications and other packages of the devices used in this study are as follows.

- Memory 64GB
- CPU 24 core
- GPU 1080(GeForce GTX 1080ti)
- CUDA 11.3
- Python 3.9
- efficientnet-pytorch==0.7.1
- matplotlib==3.6.0
- nnpack==0.1.0
- opencv-python==4.6.0.66
- Pillow==9.2.0
- scikit-learn==1.1.2
- tqdm==4.64.1
- numpy==1.23.3
- decord==0.6.0
- imgaug==0.4.0
- ttach
- tqdm

### C. Framework design and implementation

A Deep Learning framework was designed to find the tilt that occurs on the roof of Heunginjimun and the roof of Yeongnamnu. The Deep Learning framework is shown in Fig. 6

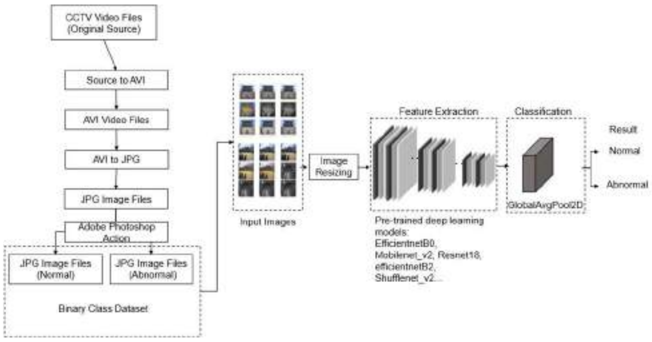


**Fig. 6. The framework for cultural heritage damage detection**

The Deep Learning framework first converts the video generated by CCTV filming cultural heritage into an AVI video format, cuts them in frame units, and extracts them into JPG images. The extracted image is used as normal data, and in the case of abnormal data, it is created by applying distortion to the normal data. This method of applying distortion to normal data uses the action cam function of Adobe Photoshop. Thereafter, an image is adjusted through a resizing procedure to use the data in a neural network model. After image resizing, the dataset is divided into training, testing, and verification sets and used as input values for neural network models. Finally, through the feature extraction part and classification part of the model, it is classified into two classes: normal and abnormal.

### D. Model selection and parameter settings

The neural network model used in machine learning loads image data from the dataset created through the preprocessing process and uses normal and abnormal images as input values for the neural network model. For the neural network model used for classification, the most commonly used models were selected: efficientnetB0 [26], mobilenet_v2 [27], resnet18 [28], efficientnetB2, shufflenet_v2 [29], alexnet [30], mnasnet [31], inception_v3[32], densernet161[33], efficientnetb4, and 10 types of pre-learning-based Deep Learning models were used. The dataset used for pre-learning is ImageNet. Learning is conducted to classify it into normal and abnormal classes through the feature extraction and classification parts of the model. Table. 4 shows the parameters used for machine learning. Optimal parameters were used as long as they were versatile for each environment. In the case of the optimizer, it was adopted through Stochastic Gradient Descent (SGD). The initial learning rate was set to 0.002 for Alexanet and 0.05 for EfficientnetB0, EfficientnetB2, Shufflenet_v2, and Mnasnet. The momentum value was set to 0.9, and 10 to 20 epochs were performed.

TABLE IV
PARAMETERS OF HEUNGINJIMUN AND YEONGNAMNU

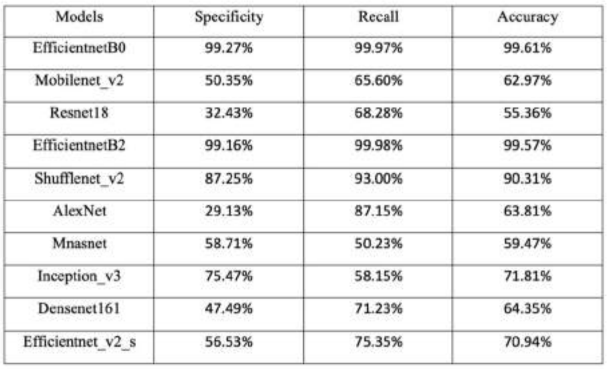
| | Heunginjimun | | | | Yeongnamnu | | |
|---|---|---|---|---|---|---|---|
| | Models | Input size | Optimizer | Epochs | Input size | Optimizer | Epochs |
| Clear-Day | efficientnetB0 | 224 | SGD (lr =0.05, momentum = 0.9) | 20 | 224 | SGD (lr =0.05, momentum = 0.9) | 10 |
| | mobilenet_v2 | 224 | SGD (lr =0.05, momentum = 0.9) | 20 | 224 | SGD (lr =0.05, momentum = 0.9) | 10 |
| | resnet18 | 224 | SGD (lr =0.05, momentum = 0.9) | 15 | 224 | SGD (lr =0.05, momentum = 0.9) | 10 |
| | efficientnetB2 | 260 | SGD (lr =0.05, momentum = 0.9) | 15 | 260 | SGD (lr =0.05, momentum = 0.9) | 10 |
| | shufflenet_v2 | 224 | SGD (lr =0.05, momentum = 0.9) | 15 | 224 | SGD (lr =0.05, momentum = 0.9) | 10 |
| | alexnet | 224 | SGD (lr =0.002, momentum = 0.9) | 15 | 224 | SGD (lr =0.002, momentum = 0.9) | 10 |
| | mnasnet | 224 | SGD (lr =0.05, momentum = 0.9) | 15 | 224 | SGD (lr =0.05, momentum = 0.9) | 10 |
| | inception_v3 | 299 | SGD (lr =0.05, momentum = 0.9) | 15 | 299 | SGD (lr =0.05, momentum = 0.9) | 10 |
| | densenet161 | 224 | SGD (lr =0.05, momentum = 0.9) | 15 | 224 | SGD (lr =0.05, momentum = 0.9) | 10 |
| | efficientnet_v2_s | 384 | SGD (lr =0.05, momentum = 0.9) | 15 | 384 | SGD (lr =0.05, momentum = 0.9) | 20 |
| Clear-Night | efficientnetB0 | 224 | SGD (lr =0.05, momentum = 0.9) | 15 | 224 | SGD (lr =0.05, momentum = 0.9) | 10 |
| | mobilenet_v2 | 224 | SGD (lr =0.05, momentum = 0.9) | 15 | 224 | SGD (lr =0.05, momentum = 0.9) | 10 |
| | resnet18 | 224 | SGD (lr =0.05, momentum = 0.9) | 15 | 224 | SGD (lr =0.05, momentum = 0.9) | 10 |
| | efficientnetB2 | 260 | SGD (lr =0.05, momentum = 0.9) | 15 | 260 | SGD (lr =0.05, momentum = 0.9) | 10 |
| | shufflenet_v2 | 224 | SGD (lr =0.05, momentum = 0.9) | 15 | 224 | SGD (lr =0.05, momentum = 0.9) | 10 |
| | alexnet | 224 | SGD (lr =0.002, momentum = 0.9) | 15 | 224 | SGD (lr =0.002, momentum = 0.9) | 10 |
| | mnasnet | 224 | SGD (lr =0.05, momentum = 0.9) | 15 | 224 | SGD (lr =0.05, momentum = 0.9) | 10 |
| | inception_v3 | 299 | SGD (lr =0.05, momentum = 0.9) | 15 | 299 | SGD (lr =0.05, momentum = 0.9) | 10 |
| | densenet161 | 224 | SGD (lr =0.05, momentum = 0.9) | 15 | 224 | SGD (lr =0.05, momentum = 0.9) | 10 |
| | efficientnet_v2_s | 384 | SGD (lr =0.05, momentum = 0.9) | 15 | 384 | SGD (lr =0.05, momentum = 0.9) | 20 |
| Rainy-Day | efficientnetB0 | 224 | SGD (lr =0.05, momentum = 0.9) | 15 | 224 | SGD (lr =0.05, momentum = 0.9) | 10 |
| | mobilenet_v2 | 224 | SGD (lr =0.05, momentum = 0.9) | 15 | 224 | SGD (lr =0.05, momentum = 0.9) | 10 |
| | resnet18 | 224 | SGD (lr =0.05, momentum = 0.9) | 15 | 224 | SGD (lr =0.05, momentum = 0.9) | 10 |
| | efficientnetB2 | 260 | SGD (lr =0.05, momentum = 0.9) | 15 | 260 | SGD (lr =0.05, momentum = 0.9) | 10 |
| | shufflenet_v2 | 224 | SGD (lr =0.05, momentum = 0.9) | 15 | 224 | SGD (lr =0.05, momentum = 0.9) | 10 |
| | alexnet | 224 | SGD (lr =0.002, momentum = 0.9) | 15 | 224 | SGD (lr =0.002, momentum = 0.9) | 10 |
| | mnasnet | 224 | SGD (lr =0.05, momentum = 0.9) | 15 | 224 | SGD (lr =0.05, momentum = 0.9) | 10 |
| | inception_v3 | 299 | SGD (lr =0.05, momentum = 0.9) | 15 | 299 | SGD (lr =0.05, momentum = 0.9) | 10 |
| | densenet161 | 224 | SGD (lr =0.05, momentum = 0.9) | 15 | 224 | SGD (lr =0.05, momentum = 0.9) | 10 |
| | efficientnet_v2_s | 384 | SGD (lr =0.05, momentum = 0.9) | 15 | 384 | SGD (lr =0.05, momentum = 0.9) | 20 |
| Rainy-Night | efficientnetB0 | 224 | SGD (lr =0.05, momentum = 0.9) | 15 | 224 | SGD (lr =0.05, momentum = 0.9) | 10 |
| | mobilenet_v2 | 224 | SGD (lr =0.05, momentum = 0.9) | 15 | 224 | SGD (lr =0.05, momentum = 0.9) | 10 |
| | resnet18 | 224 | SGD (lr =0.05, momentum = 0.9) | 15 | 224 | SGD (lr =0.05, momentum = 0.9) | 10 |
| | efficientnetB2 | 260 | SGD (lr =0.05, momentum = 0.9) | 15 | 260 | SGD (lr =0.05, momentum = 0.9) | 10 |
| | shufflenet_v2 | 224 | SGD (lr =0.05, momentum = 0.9) | 15 | 224 | SGD (lr =0.05, momentum = 0.9) | 10 |
| | alexnet | 224 | SGD (lr =0.002, momentum = 0.9) | 15 | 224 | SGD (lr =0.002, momentum = 0.9) | 10 |
| | mnasnet | 224 | SGD (lr =0.05, momentum = 0.9) | 15 | 224 | SGD (lr =0.05, momentum = 0.9) | 10 |
| | inception_v3 | 299 | SGD (lr =0.05, momentum = 0.9) | 15 | 299 | SGD (lr =0.05, momentum = 0.9) | 10 |
| | densenet161 | 224 | SGD (lr =0.05, momentum = 0.9) | 15 | 224 | SGD (lr =0.05, momentum = 0.9) | 10 |
| | efficientnet_v2_s | 384 | SGD (lr =0.05, momentum = 0.9) | 15 | 384 | SGD (lr =0.05, momentum = 0.9) | 10 |
| Cloudy-Day | efficientnetB0 | 224 | SGD (lr =0.05, momentum = 0.9) | 10 | 224 | SGD (lr =0.05, momentum = 0.9) | 10 |
| | mobilenet_v2 | 224 | SGD (lr =0.05, momentum = 0.9) | 15 | 224 | SGD (lr =0.05, momentum = 0.9) | 10 |
| | resnet18 | 224 | SGD (lr =0.05, momentum = 0.9) | 15 | 224 | SGD (lr =0.05, momentum = 0.9) | 10 |
| | efficientnetB2 | 260 | SGD (lr =0.05, momentum = 0.9) | 15 | 260 | SGD (lr =0.05, momentum = 0.9) | 10 |
| | shufflenet_v2 | 224 | SGD (lr =0.05, momentum = 0.9) | 15 | 224 | SGD (lr =0.05, momentum = 0.9) | 10 |
| | alexnet | 224 | SGD (lr =0.002, momentum = 0.9) | 15 | 224 | SGD (lr =0.002, momentum = 0.9) | 10 |
| | mnasnet | 224 | SGD (lr =0.05, momentum = 0.9) | 15 | 224 | SGD (lr =0.05, momentum = 0.9) | 10 |
| | inception_v3 | 299 | SGD (lr =0.05, momentum = 0.9) | 15 | 299 | SGD (lr =0.05, momentum = 0.9) | 10 |
| | densenet161 | 224 | SGD (lr =0.05, momentum = 0.9) | 15 | 224 | SGD (lr =0.05, momentum = 0.9) | 10 |
| | efficientnet_v2_s | 384 | SGD (lr =0.05, momentum = 0.9) | 15 | 384 | SGD (lr =0.05, momentum = 0.9) | 10 |

## IV. CONCLUSION OF EXPERIMENTAL RESULTS

### A. *Performance results of each model*

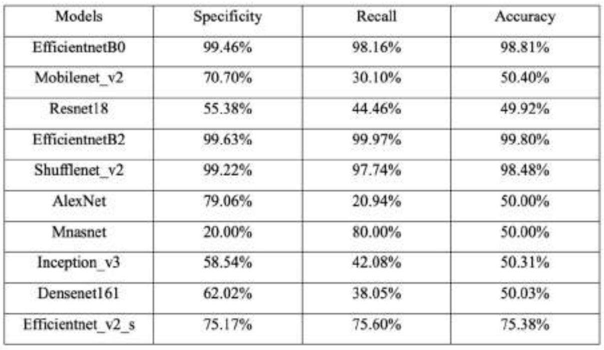
Using datasets for each environment, a normal and abnormal classification experiment was conducted on the roof of Heunginjimun and the roof of Yeongnamnu. The average values are shown in Table. 5 and Table. 6 by synthesizing the evaluation index results of each model that performed environmental classification. In the case of Heunginjimun, among ten machine learning models, EfficientnetB0, EfficientnetB2, and Shufflenet_v2 models showed excellent values of 99.61%, 99.57%, and 90.31%, respectively, based on the accuracy value. In the case of Yeongnamnu, among ten machine learning models, EfficientnetB2, EfficientnetB0, and Shufflenet_v2 showed values of 99.90%, 98.81%, and 98.48% in that order.

TABLE V
CLASSIFICATION EVALUATION METRICS VALUE OF THE MODELS FOR ROOF OF HEUNGINJIMUN



| Models | Specificity | Recall | Accuracy |
|---|---|---|---|
| EfficientnetB0 | 99.27% | 99.97% | 99.61% |
| Mobilenet_v2 | 50.35% | 65.60% | 62.97% |
| Resnet18 | 32.43% | 68.28% | 55.36% |
| EfficientnetB2 | 99.16% | 99.98% | 99.57% |
| Shufflenet_v2 | 87.25% | 93.00% | 90.31% |
| AlexNet | 29.13% | 87.15% | 63.81% |
| Mnasnet | 58.71% | 50.23% | 59.47% |
| Inception_v3 | 75.47% | 58.15% | 71.81% |
| Densenet161 | 47.49% | 71.23% | 64.35% |
| Efficientnet_v2_s | 56.53% | 75.35% | 70.94% |

TABLE VI
CLASSIFICATION EVALUATION METRICS VALUE OF THE MODELS FOR ROOF OF YEONGNAMNU



| Models | Specificity | Recall | Accuracy |
|---|---|---|---|
| EfficientnetB0 | 99.46% | 98.16% | 98.81% |
| Mobilenet_v2 | 70.70% | 30.10% | 50.40% |
| Resnet18 | 55.38% | 44.46% | 49.92% |
| EfficientnetB2 | 99.63% | 99.97% | 99.80% |
| Shufflenet_v2 | 99.22% | 97.74% | 98.48% |
| AlexNet | 79.06% | 20.94% | 50.00% |
| Mnasnet | 20.00% | 80.00% | 50.00% |
| Inception_v3 | 58.54% | 42.08% | 50.31% |
| Densenet161 | 62.02% | 38.05% | 50.03% |
| Efficientnet_v2_s | 75.17% | 75.60% | 75.38% |

### B. *Grad-CAM result of optimal model*

To verify the machine learning experiment, Gradient-weighted Class Activation Mapping (Grad-CAM) visualization was performed to find the part that is the grounds of the Machine Learning results. Visualization results were shown in Table. 7 and Table .8. In the case of Heunginjimun Gate, it can be seen that the characteristics of the 10 areas with displacement are also reflected in the Grad-CAM results, which serves as the basis for abnormal judgment. Likewise, Yeongnamnu also, it was confirmed that the area of the five areas to which the displacement was applied was referred to as the basis for abnormal judgment.

TABLE VII
HEUNGINJIMUN GRAD-CAM RESULTS

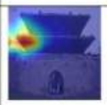
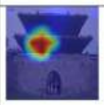
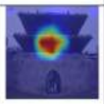
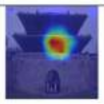
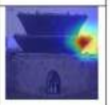
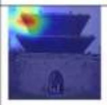
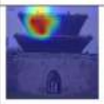
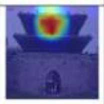
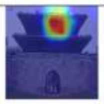
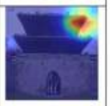
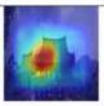
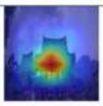
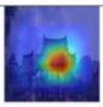
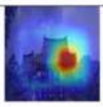
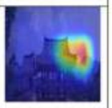
| Inclined area | Lower Left | Lower Middle Left | Lower Middle | Lower Middle Right | Lower Right |
|---|---|---|---|---|---|
| Input Image | | | | | |
| Grad-CAM Result |  |  |  |  |  |
| Inclined Area | Upper Left | Upper Middle Left | Upper Middle | Upper Middle Right | Upper Right |
| Input Image | | | | | |
| Grad-CAM Result |  |  |  |  |  |

TABLE VIII
YEONGNAMNU GRAD-CAM RESULTS

| Inclined Area | Left | Middle Left | Middle | Middle right | Right |
|---|---|---|---|---|---|
| Input Image | | | | | |
| Grad-CAM Result |  |  |  |  |  |

## V. CONCLUSION

In this paper, basic research was conducted on the development of a system for detecting anomalies in architectural, cultural properties so that abnormal symptoms can be identified or prompt initial responses can be made for the preservation of cultural properties. Heunginjimun and Yeongnamnu, treasures of the Republic of Korea, were used as subjects of basic research. CCTV images of Heunginjimun and Yeongnamnu were collected to build a dataset to determine the roof's displacement. A Deep Learning model was trained with the built data to test the abnormal detection performance so that the model can detect the degree of tilt corresponding to damage to cultural heritage.

The experimental results for 10 models on the roof of Heunginjimun were as follows. Summarizing only the top 3 models with optimal results, the EfficientnetB0, EfficientnetB2, and Shufflenet_V2 models showed excellent values of 99.61%, 99.57%, and 90.31% based on the accuracy. Specificity also showed 99.27%, 99.16%, and 87.25% values for EfficientnetB0, EfficientnetB2, and Shufflenet_V2. Finally, based on the recall, EicientnetB0, EfficientnetB2, and Shufflenet_V2 showed values of 99.97%, 99.98%, and 87.15% in order. In the case of the roof of Yeongnamnu, as with the roof of Heunginjimun, a comparative experiment was conducted on 10 models. Based on the accuracy value, the EfficientnetB0, EfficientnetB2, and Shufflenet_V2 models showed values of 98.81%, 99.80%, and 98.48%, and based on the specificity, they showed values of 99.46%, 99.63%, and 99.22%. In addition, based on the recall, values of 98.16%, 99.97%, and 97.74% were shown. As a result, the Efficientnet series models of EfficientnetB0 and EfficientnetB2 showed optimal performance for tilt detection.

Then, using Grad-CAM, we tried to check whether the EfficientnetB0 model, which represents the Efficientnet algorithm, produced results based on appropriate judgment grounds. As a result of Grad-CAM, it was confirmed that the model made a judgment based on the change in tilt displacement of the roof. The framework proposed in this experiment was proposed to perform rapid damage detection of cultural heritage and to help apply artificial intelligence, which remains at the introductory stage in managing and preserving cultural heritage. First, a normal dataset was constructed for Deep Learning and abnormal data based on normal data was generated. In addition, in order to find the most effective and universally usable Deep Learning model, we tried to derive a model that is optimal for all environments filmed by CCTV using 10 commonly used models. In conclusion, the Efficientnet series algorithm performed the best in detecting the tilt of cultural properties, which means that the Efficientnet model is the most universally effective model in situations where parameter tuning is limited. However, this study did not reflect seasonal characteristics due to an insufficient data collection period. The five environments covered the most basic environments, so learning by applying more displacement scenarios and mixed environments in future experiments is necessary. Complementing these limitations will further improve the robustness of the framework.

## ACKNOWLEDGMENT

This paper was funded by the government (Cultural Heritage Administration) in 2023 with the support of the National Research Institute of Cultural Heritage (No.2021A01D06-001, Development of damage detection and alarm technology based on intelligent image analysis for safety diagnosis of Immovable cultural heritage)

## REFERENCES

[1] E. C. Choi, "Cultural heritage in statistics," *Cultural Heritage Administration of the Republic of Korea, 2022*, Available:https://www.cha.go.kr/cop/bbs/selectBoardArticle.do?nttId=85272&bbsId=BBSMSTR_1020&pageIndex=1&pageUnit=10&searchCnd=&searchWrd=&ctgryLrcls=&ctgryMdcls=&ctgrySmcls=&ntcStartDt=&ntcEndDt=&searchUseYn=&mn=NS_03_07_04. Accessed 31 March 2023

[2] H.M. Kim, "Cultural heritage in statistics," *Cultural Heritage Administration of the Republic of Korea, 2023*, Available:https://www.cha.go.kr/cop/bbs/selectBoardArticle.do?nttId=82245&bbsId=BBSMSTR_1020&pageIndex=1&pageUnit=10&searchCnd=&searchWrd=&ctgryLrcls=&ctgryMdcls=&ctgrySmcls=&ntcStartDt=&ntcEndDt=&searchUseYn=&mn=NS_03_07_04. Accessed 31

[3] "Current Status of Maintenance and Maintenance of Cultural Heritage," *Korean Statistical information Service(KOSIS), 2023*, Available:https://kosis.kr/common/meta_onedepth.jsp?vwcd=MT_OTITLE&listid=150. Accessed 01 July 2023

[4] M. Mishra, T.Barman, and G.V.Ramana, "Artificial intelligence-based visual inspection system for structural health monitoring of cultural heritage," *Journal of Civil Structural Health Monitoring*, pp. 1-18, 2022, https://doi.org/10.1007/s13349-022-00643-8

[5] L.E.Mansuri, D.A. Patel,"Artificial intelligence-based automatic visual inspection system for built heritage," *Smart and Sustainable Built Environment,* vol. 11(3), pp. 622-646, 2022, https://doi.org/10.1108/SASBE-09-2020-0139

[6] M. Mishra," Machine learning techniques for structural health monitoring of heritage buildings: A state-of-the-art review and case

studies," *Journal of Cultural Heritage*, vol 47, pp. 227-245, 2021, https://doi.org/10.1016/j.culher.2020.09.005

[7] T.Yu, C.Lin, S. Zhang, C. Wang, X. Ding, H. An, and J.Zhang, "Artificial Intelligence for Dunhuang Cultural Heritage Protection: The Project and the Dataset," *International Journal of Computer Vision* vol.130(11), pp.2646-2673, 2022, https://doi.org/10.1007/s11263-022-01665-x

[8] A. Belhi, H. Gasmi, A. Bouras, T. Alfaqheri, A. S. Aondoakaa, A.H.Sadka, and S.Foufou, "Machine learning and digital heritage: the CEPROQHA project perspective," *Springer Singapor,* 2020, [In 4th International Congress on Information and Communication Technology: ICICT 2019 London, 2019, vol.2, pp.363-374], https://doi.org/10.1007/978-981-32-9343-4_29

[9] N. Wang, X. Zhao, P. Zhao, Y. Zhang, Z. Zou, and J. Ou, "Automatic damage detection of historic masonry buildings based on mobile deep learning," *Automation in Construction,* vol.103, pp.53-66, 2019, https://doi.org/10.1016/j.autcon.2019.03.003

[10] D. Bienvenido-Huertas, JE. Nieto-Julián, JJ. Moyano, JM. Macías-Bernal, J. Castro, "Implementing artificial intelligence in h-bim using the J48 algorithm to manage historic buildings," *Int J Archit Herit*, vol.14(8), pp.1148–1160, 2019, https://doi.org/10.1080/15583058.2019.1589602

[11] T. Bakirman, B. Kulavuz, and B. Bayram, "Use of Artificial Intelligence Toward Climate-neutral Cultural Heritage," *Photogrammetric Engineering & Remote Sensing*, vol. 89(3), pp.163-171, 2023, https://doi.org/10.14358/PERS.22-00118R2

[12] I. Garrido, J. Erazo-Aux, S. Lagüela, S. Sfarra, C. Ibarra-Castanedo, E. Pivarčiová, and P. Arias, "Introduction of deep learning in thermographic monitoring of cultural heritage and improvement by automatic thermogram pre-processing algorithms," *Sensors, vol. 21(3), pp.750*, 2021, https://doi.org/10.3390/s21030750

[13] Z. Zou, X. Zhao, P. Zhao, F. Qi, and N. Wang, "CNN-based statistics and location estimation of missing components in routine inspection of historic buildings," *Journal of Cultural Heritage*, vol. 38, pp. 221-230, 2019, https://doi.org/10.1016/j.culher.2019.02.002

[14] T. Sharma, P. Agrawal, and N. K. Verma, "Detection of dust deposition using convolutional neural network for heritage images,"*Springer Singapore*, 2019 [In Computational Intelligence: Theories, Applications and Future Directions-Volume II: ICCI-2017, *pp.*347-359], https://doi.org/10.1007/978-981-13-1135-2_27

[15] E. J. Jeong, "Heunginjimun and Heritages around the Gate,"*Art History & Cutural Heritages*, vol. 5, pp. 79-109, 2016.

[16] S. A. Park, K. W. Min, and J. S. Choi, "Ambient vibration analysis of Heunginjimun," *Journal of The Architectural Institute of Korea: Structure & Construction*, vol. 27(5), pp. 19-26, 2011.

[17] T. G. Eom, S. J. Kim, J. L. Park, H. M. Kang, and W. K. Sim, "Interpretation of Cultural Landscape at the Geumsidang (今是堂) sibigyung (12 Landscapes) in Miryang, Gyungnam," *Journal of the Korean Institute of Traditional Landscape Architecture*, vol. 29(2), pp. 1-18, 2011.

[18] S. L. Ryoo, "A Study on the Changes of the Government Pavilion, Miryang Yeongnamnu in terms of Function and Spatiality," *Journal of the Architectural Institute of Korea Planning & Design,* vol. 34(8), pp. 69-76, 2019, https://doi.org/10.5659/JAIK_PD.2018.34.8.69

[19] H. Y. Lee, "A Study on the Historic Changes of Yungnam-Ru in Historic Periods and Architectural Building Forms," *Journal of architectural history*, vol. 9(1), pp. 7-25, 2000.

[20] S. Y. Lee, H. H. Cho,"Damage Detection and Safety Diagnosis for immobable Cultural Assets Using Deep Learning Framwork," *ICACT2023*, pp. 310-313, 2023.

[21] Y. LeCun, Y. Bengio, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks* 3361(10), 1995.

[22] J. Wu, "Introduction to convolutional neural networks," *National Key Lab for Novel Software Technology*, vol. 5(23), pp. 495, 2017, Nanjing University. China.

[23] R. Chauhan, K. K. Ghanshala, and R. C. Joshi, "Convolutional neural network (CNN) for image detection and recognition," *2018 first international conference on secure cyber computing and communication (ICSCCC 2018),* pp. 278-282, 10.1109/ICSCCC.2018.8703316.

[24] M. Jogin, M. S. Madhulika, G. D. Divya, R. K. Meghana, and S. Apoorva, "Feature extraction using convolution neural networks (CNN) and deep learning," *2018 3rd IEEE international conference on recent trends in electronics, information & communication technology (RTEICT)*, pp. 2319-2323], 10.1109/RTEICT42901.2018.9012507.

[25] C. D. James, J. B. Aimone, N. E. Miner, C. M. Vineyard, F. H. Rothganger, K. D. Carlson, and S. J. Plimpton, "A historical survey of algorithms and hardware architectures for neural-inspired and neuromorphic computing applications," *Biologically Inspired Cognitive Architectures* vol. 19, pp.49-64, 2017, https://doi.org/10.1016/j.bica.2016.11.002.

[26] M. Tan, Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *International conference on machine learning 2019*, pp. 6105-6114, 2019, May.

[27] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv, 2017,* preprint arXiv:1704.04861, https://doi.org/10.48550/arXiv.1704.04861.

[28] K. He, X. Zhang, S. Ren, and J. Sun, " Deep residual learning for image recognition," *Proceedings of the IEEE conference on computer vision and pattern recognition 2016,* pp. 770-778, 2016.

[29] N. Ma, X. Zhang, H. T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient cnn architecture design," *2018 Proceedings of the European conference on computer vision (ECCV),* pp. 116-131,2018.

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60(6), pp. 84-90, 2017, https://doi.org/10.1145/3065386

[31] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le, "Mnasnet: Platform-aware neural architecture search for mobile," *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition 2019,* pp. 2820-2828.

[32] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *Proceedings of the IEEE conference on computer vision and pattern recognition 2019*, pp. 2818-2826.

[33] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, (2017) "Densely connected convolutional networks," *Proceedings of the IEEE conference on computer vision and pattern recognition 2017,* pp. 4700-4708.

**Sang-Yun Lee** (B'94–M'96–D'07) was born in South Korea 1971, is a principal researcher at Police Science & Public Safety ICT Research Center of Digital Convergence Research Laboratory in ETRI. He has been working at ETRI since 1999. In 2008, he received Ph.D. in Electronics and Telecommunications Engineering at the University of Hanyang(Rep. of Korea). He has been developing technologies in the fields of Broadcasting Communication, System Software, Embedded Software, Artificial Intelligence, and etc.

His main research interests have been in Computational Sciences. Currently, he is involved in developing a technique to detect displacement using Artificial Intelligence technology for CCTV images of cultural assets. He also leads several projects including disaster management for cultural heritage. Since 2016, He acts as an editor of the Study Group 16 (SG16) in the ITU-T and has been developing international standards. He is an international standards expert at TTA and a member of the Korean ITU-T Research Committee. He has been carrying out more than 10 government-funded projects. He is author of more than 70 scientific papers and has registered more than 10 patents.

**Daekyeom Lee** (B' 16 – M'18) is a Lead of AI Convergence Technology Research Team at Season. From 2021 to present, he has been working at Season Co., Ltd. He graduated from Dongguk University (Rep. of Korea) with a double major in energy systems and economics in 2016 and received a master's degree in engineering from Sejong University (Rep. of Korea) in 2018. Since 2018, he has conducted research in statistics, data analysis, and artificial intelligence.

His main research interest was big data analysis and data science. Additionally, after joining Season Co., Ltd., he has been working as a team leader developing artificial intelligence technology in terms of data utilization. He mainly works on the application of artificial intelligence technology using manufacturing data, object detection using deep learning, and improvement of learning speed through transfer learning. In addition, he is conducting joint research on technology utilization in cooperation with related research institutes and companies and has authored 8 domestic and foreign journals.

# A Study on Connectivity Evaluation Among Peer Groups in Pure P2P Networks

Yutaka Naito*, Takumi Uemura*, Takashige Hoshiai**

*Faculty of Computer and Information Sciences, Sojo University, 4-22-1 Ikeda, Nishi-ku, Kumamoto, 860-0082 Japan
Sojo University IoT/AI Center, 4-22-1 Ikeda, Nishi-ku, Kumamoto, 860-0082 Japan
**naito@cis.sojo-u.ac.jp, t_uemura@cis.sojo-u.ac.jp, hoshiai_takashige@yahoo.co.jp**

***Abstract*— Recently P2P (peer-to-peer) network garners attention as a technology for developing a peer group by use of connection of peers which function as autonomously distributed and cooperative units virtualized from computer resources. In this paper, we focus on pure P2P networks which form peer groups by connecting peers without the need for intermediates. We execute performance evaluation by using computer simulation and propose a method to measure peer groups' connectivity utilizing mean number of connected peer groups without using peers' arrival and departure rates on peer groups.**

***Keyword*— P2P, Pure model, Cluster model, Performance evaluation, Peer groups' connectivity**

## I. INTRODUCTION

RECENTLY P2P (peer-to-peer) network garners attention as a technology for developing a peer group by use of connection of peers which function as autonomously distributed and cooperative units virtualized from computer resources. JXTA [1], [2], SOBA [3], [4], and SIONet [5]-[8] are representative examples. Lately a P2P technology called blockchain [9] has attracted particular attention.

P2P network is classified into two types: pure model (flat model) [10], in which there is no intermediate to connect peer groups, and cluster model [11], in which there is an intermediate.

In the cluster model, an intermediate always connects peer groups, so there is no fragmentation among peer groups.

On the other hand, in the pure model, no intermediate exists to connect peer groups, so any peer connects peer groups by autonomously connecting with peers in other peer groups. Therefore, when a peer connecting peer groups departs from a peer group, the peer groups are broken up into fragments. As a result, it becomes difficult to share information among peer groups. Thus, in the pure model, it is important to evaluate connectivity among peer groups from the perspective of information sharing among peer groups [12], [13].

In this paper, we define connectivity among peer groups (%) as follows.

$$\frac{\text{Maximum number of connected peers}}{\text{Total number of peers}} \times 100 \qquad (1)$$

In other words, the connectivity indicates how many percent of the peers are connected among all the peers, and no fragmentation among peer groups occurs when the connectivity is 100%.

While it is considered important to quantitatively evaluate the connectivity among peer groups in the pure model, some research papers [10], [12], [14]-[17] report on the evaluation of the connectivity among peer groups. However, in all of them, it is necessary to survey in advance "the arrival rate of peers in a peer group" and "the departure rate from a peer group," which are input parameters in the evaluation model of the connectivity. However, in this case, it is a big burden for the evaluators to obtain the arrival rate and the departure rate in advance by the actual survey.

Therefore, in this paper, we examine the performance of the evaluation model using computer simulations. We propose a method to calculate the connectivity among peer groups by using the mean number of connected peer groups without using the arrival and departure rates of peers as input parameters of the evaluation model [18]-[20].

The rest of this paper is organized as follows. The mechanism of the pure model is explained in section II. Next, in section III, we introduce a Community Coexistence Society (CCS), as an example of the pure model application to realize a sustainable welfare society in Japan. In section IV, we explain the performance evaluation model (simulation model) and scale to obtain the connectivity among peer groups. In section V, we show the results of the performance evaluation by simulation and discuss the results. Section VI concludes the paper and discusses future work.

## II. MECHANISM OF THE PURE MODEL

In this chapter, we describe the mechanism for connecting peer groups in the pure model. In figure 1(a) and figure 1(b), we consider a virtualized peer as the minimum unit of autonomous distributed cooperation and a peer group formed as a set of peers.

To deepen the understanding of the pure model, we discuss the cluster model, which is a contrast to the pure model. In the cluster model, peer groups are connected to each other through an intermediate, as shown in figure 1(a). In this method, there is no fragmentation between peer groups unless

Manuscript received Oct. 10, 2022.

This work is a follow-up of the invited journal to the accepted & presented paper entitled "A Study on Connectivity Evaluation Among Peer Groups in Pure P2P Networks" of the 25th International Conference on Advanced Communication Technology (ICACT2023).

Yutaka Naito is with Sojo University, Kumamoto, 860-0082 JAPAN (corresponding author e-mail: naito@cis.sojo-u.ac.jp).

Takumi Uemura is with Sojo University, Kumamoto, 860-0082 JAPAN (e-mail: t_uemura@cis.sojo-u.ac.jp).

Takashige Hoshiai is with Sojo University IoT/AI Center, Kumamoto, 860-0082 JAPAN (e-mail: hoshiai_takashige@yahoo.co.jp).

the intermediate goes down, but the operating cost of the intermediate is often burdensome. Meanwhile, in the pure model, as shown in figure 1(b), peer A and peer B, which belong to peer group III, simultaneously participate in (belong to) peer group II and peer group I, respectively, so that peer α and peer Y can be connected through peer A and peer B. As a result, peer group I, peer group II, and peer group III can be connected. On the other hand, when peer A and/or peer B departs the peer group, the peer group may be fragmented, and cooperation and information sharing among peer groups may be disrupted.

Thus, although the pure model has the problem of fragmentation, it is unique in that it connects peer groups in a "loose," "flexible," and "autonomous distributed cooperation" form without the need for an intermediate, and it is widely used in a variety of fields because it does not require the operating costs of an intermediate.

There are many studies on fragmentation among peer groups. As examples of peer's departure from peer groups as a cause of fragmentation, a hardware failure of peers [14] and interruption of file exchange services [15] causes departure of peers from peer groups. Also, blockchain mining cost incurs peers' departure from a peer group and produces fragmentation of peer group [16], [17]. Furthermore, many examples of the application of the pure model concept to inter-operational and inter-organizational collaboration in real-world business models have been reported. For example, there is a reference [21] that reports that when bank employees (peers) multitask among departments (peer groups), such as the teller service department and the loan counselling department at a bank, fragmentation occurs when employees depart from the department. Furthermore, reference [13] reports that hotel employees (peers) simultaneously participate in (belong to) different departments such as the front desk and the guest room cleaning (peer groups) to connect departments and that fragmentation occurs when hotel employees depart the department.

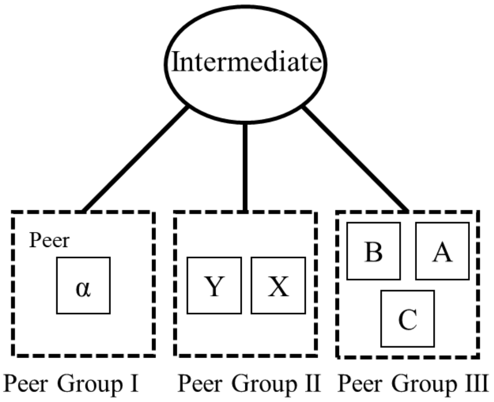


**Fig. 1(a).** Peer group connecting method (cluster model)

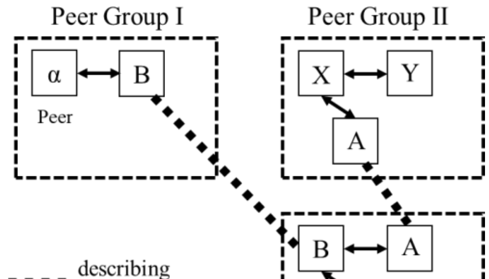


**Fig. 1(b).** Peer group connecting method (pure model)

Although there is a problem of fragmentation, the trend towards applying the pure model, which does not require high operating costs for intermediates, to inter-organizational collaboration is notable in business categories involving customer service, such as bank tellers and restaurants, and in labor-intensive industries, such as hotels, lodging, nursing care, and so on. It has been reported that 24.4% of hotels and inns have introduced the pure model in recent years and have achieved positive results [22]. While these case studies are reported, in the next chapter we introduce CCS as an example using pure model inter-organizational collaboration in Japan.

## III. AN EXAMPLE OF THE PURE MODEL USE

In this chapter, we introduce CCS as an example of use for improving the understanding of the pure model inter-organizational collaboration. In recent years, the number of single-person households in Japan has been increasing due to the aging of the population and the change in customs, such as parents and children living separately to respect privacy, and the isolation of individuals due to the reduction of welfare and medical services such as nursing care caused by the lack of financial resources due to the long-term economic stagnation has become a social problem.

So far, to prevent isolation, local public support organizations have provided information to residents by visiting households and making phone calls to confirm the safety of residents and to introduce them to consultation services. However, the number of users has decreased due to the declining population, and it has also become difficult to secure professional human resources, making it difficult to stably operate local public support organizations.

Therefore, the Japanese government advocates the establishment of a CCS in which information is shared by connecting groups such as public support providers, local businesses, local volunteer groups, and the homes of local residents in the pure model to prevent the isolation of individuals in the community [23].

In the past, public support has been based on information sharing in the form of the cluster model using an intermediate. For example, in the cluster model, as shown in figure 2(a), a facility for the disabled has members such as a staff and disabled persons who receive services. Likewise, in a agricultural corporation, there are a staff and laborers, and in a foreigner support volunteer organization, there are a staff and foreigner receiving support. In general, representative meeting is held for the purpose of cooperation and information sharing among the organizations, for example, once a month. This makes it possible for organizations to cooperate and share information with each other, and there is no fragmentation among the organizations unless the representative meeting is cancelled. Note that the representative meeting, organization, and members correspond to the intermediate, peer group, and peers, respectively, in figure 1(a).

However, the method of cooperation and information sharing among organizations through representative meeting has the following issues.

Issue 1: Delay in information transfer

Information on other organizations obtained by a staff member at a representative meeting is transferred from the staff member to the other members, which requires time for information sharing, resulting in a time delay.

Issue 2: Difficulty in creating a sense of ownership

Information about other organizations relayed by staff members tends to become someone else's business for members and members tend to lack a sense of ownership.

Issue 3: Overloading of staff members

Information is gathered by staff members who attend to a representative meeting, and the burden of information transmission is concentrated on them. Mental and physical exhaustion and employee turnover due to the concentration of workload on staff members have already become a social problem in the nursing care and medical fields [24].

In contrast, the peer group connecting method in the pure model, which is being used to realize a CCS, members of a facility for the disabled autonomously and simultaneously participate in or depart from an agricultural corporation or a foreigner support volunteer organization, and voluntarily share information with the members of the participating organizations. For example, as shown in figure 2(b), when disabled person C participates in the facility for the disabled and the agricultural corporation at the same time, and when laborer Y participates in the agricultural corporation and the foreigner support volunteer organization at the same time, the facility for the disabled, the agricultural corporation, and the foreigner support volunteer organization are connected as a result. Thus, collaboration and information sharing among the organizations become possible. Here, note that each member and each organization correspond to a peer and a peer group, in figure 1(b), respectively.

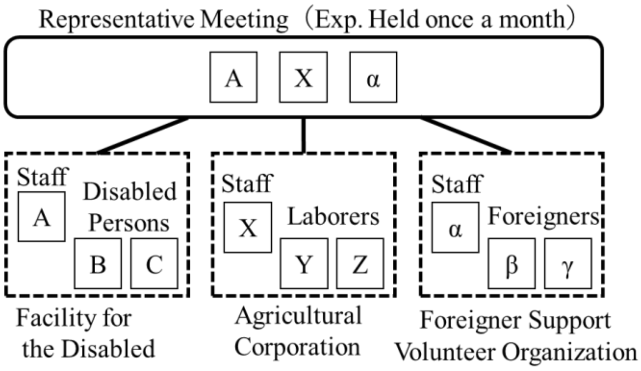

Representative Meeting（Exp. Held once a month）
A
X
α
Staff
A
Disabled Persons
B
C
Staff
X
Laborers
Y
Z
Staff
α
Foreigners
β
γ
Facility for the Disabled
Agricultural Corporation
Foreigner Support Volunteer Organization

**Fig. 2(a).** Collaboration and information sharing among organizations through representative meeting

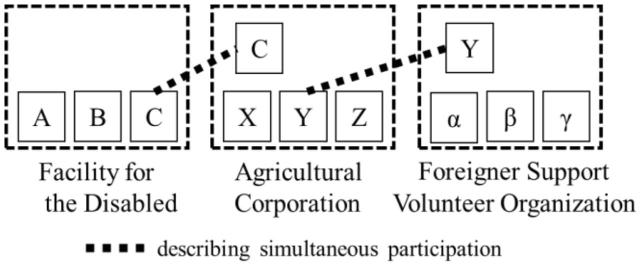

C
Y
A
B
C
X
Y
Z
α
β
γ
Facility for the Disabled
Agricultural Corporation
Foreigner Support Volunteer Organization
■ ■ ■ ■ describing simultaneous participation

**Fig. 2(b).** Collaboration and information sharing among organizations in the pure model

This pure model solves the above three issues seen in the cluster model as follows.

Solution to issue 1: Speeding up information transfer

Compared to the representative meeting held periodically, the pure model has the advantage that information sharing is less likely to be delayed because the facilities for the disabled and the agricultural corporation are constantly connected through disabled person C.

Solution to issue 2: Creating ownership of information

Disabled person C, who belongs to a facility for the disabled, also belongs to an agricultural corporation, so he/she can have a sense of ownership of the information about the agricultural corporation.

Solution to issue 3: Reduction of staff workload

The information can be shared within the facility for the disabled by having C, who belongs to the facility for the disabled, simultaneously participating in the agricultural corporation and sharing the information obtained in the agricultural corporation without the involvement of staff A. This type of collaboration between the disabled and the agricultural corporation is widely called "agricultural welfare collaboration" and has been attracting attention in recent years as a place where the disabled can play an active and autonomous role [25]. In realizing a CCS, it is important for various entities including individuals to autonomously participate in the activities of not only one organization but also various organizations called “second place” and “third place,” and the Japanese government is actively promoting the introduction of peer group collaboration in the community using the pure model [23].

On the other hand, a structural problem of the pure model, the connectivity decreases due to the fragmentation among the peer groups. For example, in figure 2(b), when a disabled person (peer) C departs from the agricultural corporation (peer group), the facility for the disabled peer group is separated from the peer group of the agricultural corporation and the foreigner support volunteer organization. In other words, information sharing among peer groups is disrupted, and at the same time, peers A, B, and C are isolated from other peers.

Therefore, in the pure model, quantitative evaluation of connectivity is important to prevent fragmentation. In previous studies, information on participation in and departure from each peer group (organization) by each peer (member) was collected over a long period of time to determine the arrival rate and the departure rate in advance [10], [12], [14]-[17], and computer simulations were used to calculate the connectivity among peer groups, which caused an excessive research cost.

Therefore, in this paper, we examine the performance of the evaluation model using computer simulations. We propose a method to calculate the connectivity among peer groups by using the mean number of connected peer groups without using the arrival and departure rates of peers as input parameters of the evaluation model [18]-[20]. The mean number of connected peer groups is the mean number of peer groups in which any peer participates at the same time.

The reason for using the mean number of connected peer groups as an input parameter for the evaluation model instead of the arrival and departure rates is its convenience. It is relatively easier to obtain the mean number of connected peer groups than to obtain the arrival and departure rates by a measurement survey. For example, in a questionnaire survey, the question "How many groups, on average, did you belong to?" is easier to answer than "How often did you join in and depart from each group?" Hence, the proposed method requires less effort and cost to obtain the information needed to calculate the connectivity.

The next section describes the performance evaluation model used in simulations to calculate the connectivity using the proposed method.

## IV. PERFORMANCE EVALUATION MODEL

In this section, we describe a performance evaluation model for quantitatively evaluating the connectivity among peer groups in the pure model.

Figure 3 shows the example of the performance evaluation model. This figure shows peer groups I to III formed by virtualized peers A to D as the minimum unit of autonomous distributed cooperation.

By peer B belonging to peer group I participates in peer group II, peer B participates in peer groups I and II simultaneously. As the same way, by peer C participates in peer groups II and III, peer A and peer D are connected via peer B and peer C. As the result of the connection among peer A, peer B, peer C and peer D, peer group I, peer group II, and peer group III are connected.

Thus, all peers independently participate in all peer groups with an arrival rate $\lambda$. Here, figure 3 shows the situation where peer B belonging to peer group I tries to participate in peer group II and peer group III with an arrival rate $\lambda$. As a result, it participates only in peer group II. We assume all peers in a peer group are always connected to each other. Therefore, we do not deal with the connection topology between peers in a peer group in this paper.

On the other hand, all peers independently depart from all peer groups to which they belong with a departure rate $\mu$. For example, in figure 3, if peer B departs from peer group II, a fragmentation occurs between peer groups II-III, and peer group I. In this paper, for the sake of simplicity, we assumed that the arrival rate of each peer is all the same. And the departure rate of each peer is also all the same. We will discuss the performance evaluation model having different arrival and departure rate on each peer in the future work.

We explain the parameter and the performance evaluation scale on the performance evaluation model.

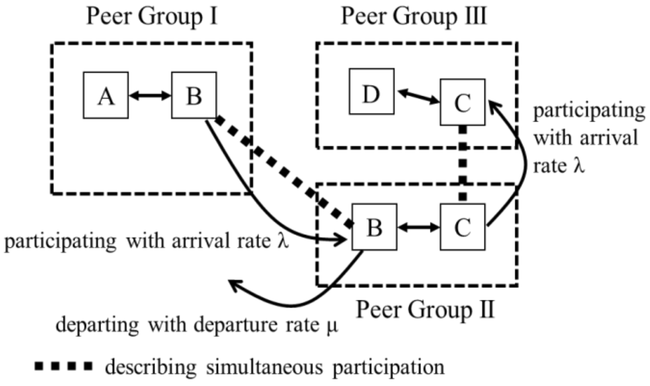

Peer Group I
Peer Group III
A
B
D
C
participating with arrival rate λ
B
C
participating with arrival rate λ
departing with departure rate μ
Peer Group II
▪ ▪ ▪ ▪ describing simultaneous participation

**Fig. 3.** Performance evaluation model

- The number of peer groups $m$: Total number of peer groups.
- The total number of peers $n$: The total number of peers, i.e., the sum of all peers participating in $m$ peer groups. For example, $n$=4 in figure 3.
- Arrival rate $\lambda$: The arrival rate of each peer in each peer group. The mean number of times a peer participates in each peer group per unit of time. It is equal to the reciprocal of the mean time of participation in each peer group.
- Departure rate $\mu$: The mean number of times that a peer departs from each peer group per unit of time. It is equal to the reciprocal of the mean time of sojourn in each peer group.
- Utilization rate $\rho$: Ratio of arrival rate to departure rate ($\rho = \lambda / \mu$)
- Maximum number of connected peers $L$: The maximum number of connected peers at a given point in time.
- Connectivity $S$: Percentage of peers that are connected among peer groups. $E\{L\}$ is the mean value of $L$, obtained by simulation. $S$ is calculated from the following formula.

$$S = \frac{E\{L\}}{n} \times 100(\%) \qquad (2)$$

- The number of simultaneously participated peer group on each peer $K_i$: The number of peer groups that the peer $p_i$ ($i$=1 to $n$) is participating at the same time at a given point in time.
- The number of simultaneously participated peer group $K$: The mean number of $K_i$ at a given point in time.
- The mean number of connected peer groups $E\{K\}$: The mean number of peer groups in which a peer participates at the same time. $E\{K\}$ is the mean value of $K$ over the observation period, calculated by simulation.
- Peer's simultaneous participation rate $R$: The number of peer groups in which a single peer participates at the same time.

$$R = \frac{E\{K\}}{m} \times 100\ (\%) \qquad (3)$$

For example, figure 4 shows a situation where there is a fragmentation among peer groups when $m$=3 and $n$=6. In peer group I, the number of connected peers is 2 because the $p_1$ and $p_2$ are connected. On the other hand, in peer group II and peer group III, the number of connected peers is 4

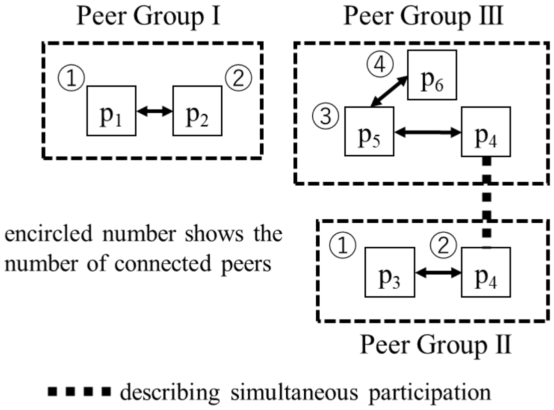

Peer Group I
Peer Group III
①
②
$p_1$
$p_2$
④
$p_6$
③
$p_5$
$p_4$
encircled number shows the number of connected peers
①
②
$p_3$
$p_4$
Peer Group II
▪ ▪ ▪ ▪ describing simultaneous participation

**Fig. 4.** Example of maximum number of connected peers (L) and number of connected peer groups (Ki).

because p3, p4, p5 and p6 are connected. Therefore, the maximum number of connected peers L is 4. Moreover, the number of simultaneously participated peer group on each peer Ki, which is the number of peer groups in which pi participates simultaneously, is K1 =1, K2 =1, K3 =1, K4 =2, K5 =1 and K6 =1.

## V. SIMULATION RESULTS AND DISCUSSION

In this section, we clarify that the connectivity $S$ is calculated without using the arrival and departure rates (i.e., participation and departure information) from simulation results.

The convergence condition of the simulation is set to be within 3% difference from the previous $E\{L\}$ value.

Therefore, the convergence condition is given by the following formula where the time until convergence is the observation period.

$$\frac{\left| E\{L\}^{(t+\Delta t)} - E\{L\}^{(t)} \right|}{E\{L\}^{(t)}} < \varepsilon \tag{4}$$

Where $t$ is any time in the simulation, $\Delta t$ is the minimum unit time, and ε is the threshold.

The simulation was based on the performance evaluation model shown in figure 3. The simulator is implemented in C language, and the specification of the computer used for the simulation is presented in Table 1. The time required to obtain a single plot (point) in the graph was approximately 10 seconds, and the confidence interval ±5% was obtained by running the simulation three times for each plot in 95% confidence interval.

To ensure sufficient coverage, the number of peer groups $m$ was set to $m$=10 and $m$=30, and the total number of peers $n$ was set from $n$=30 to $n$=1,500.

Figure 5 shows the simulation results for $m$=10 and $n$=30. For both the utilization rate $\rho$=0.4 and $\rho$=0.04, the connectivity $S$ does not depend on the departure rate $\mu$, but on $\rho$, i.e., the ratio of $\lambda$ to $\mu$. Therefore, once $\rho$ is determined, the connectivity $S$ is independent of the departure rate $\mu$. Therefore, figure 5 describes that robustness to $\rho$ holds for the connectivity $S$.

Next, figure 6 shows the relationship between the departure rate $\mu$ and the mean number of connected peer groups $E\{K\}$ for $m$=10 and $n$=30, where $E\{K\}$ is the mean number of peer groups in which a peer is participating at the same time. Figure 6 shows that $E\{K\}$ does not depend on the departure rate $\mu$, but on $\rho$, i.e., the ratio of $\lambda$ to $\mu$, for both the utilization rate $\rho$=0.4 and $\rho$=0.04. Therefore, once $\rho$ is determined, $E\{K\}$ is uniquely determined. Figure 6 describes that robustness with respect to $\rho$ holds for the mean number of connected peer groups $E\{K\}$.

Next, figure 7 and figure 8 show the simulation results of the departure rate $\mu$ and the peer's simultaneous participation rate $R$ for $n/m$=3 ($m$=10, $n$=30 and $m$=30, $n$=90) and $n/m$=50 ($m$=10, $n$=500 and $m$=30, $n$=1,500), respectively, for the utilization rate $\rho$ = 0.4 and $\rho$ = 0.04. From the respective figures, the peer's simultaneous participation rate $R$ is dependent on the utilization rate $\rho$, since the same utilization rate $\rho$ results in the same peer's simultaneous participation rate $R$.

**Table 1.** Simulation execution environment

| | |
|---|---|
| Computer | Mac Book Pro (Late2013) |
| OS | OS X version 10.9.4 |
| CPU | Intel Core i7 2.0 GHz |
| Memory | 8GB (DDR3 1,600 MHz) |
| HDD | 250 GB |

Figure 9 is constructed by combining figures 5 and 6. Figure 9 describes that even without obtaining $\lambda$ and $\mu$, the mean number of connected peer groups $E\{K\}$ provides the connectivity $S$. For example, if each peer participates in 3 peer groups on average, then the connectivity $S$ is approximately 100 %. This means that the derivation of $\lambda$ and $\mu$ (setting of input parameters) by actual measurement is not necessary.

Finally, figure 10 compares $n/m$=3 ($m$=10, $n$=30 and $m$=30, $n$=90) and $n/m$=50 ($m$=10, $n$=500 and $m$=30, $n$=1,500) in the relationship between the mean number of connected peer groups $E\{K\}$ and connectivity $S$. The figure shows that the connectivity $S$ depends on $n/m$, since the connectivity $S$ is the

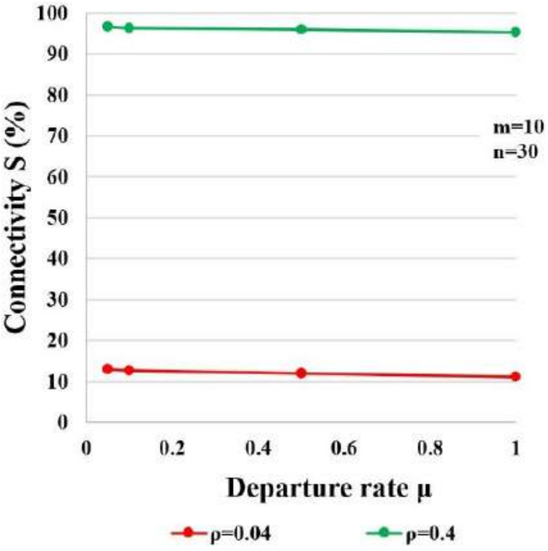


**Fig. 5.** Robustness to $\rho$ on the Connectivity $S$

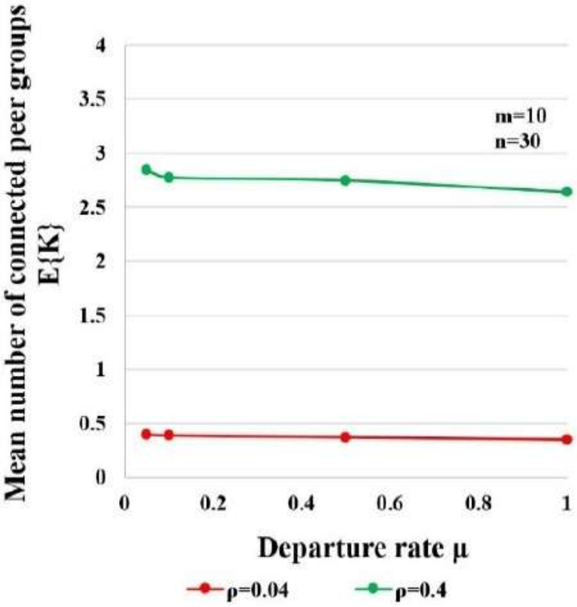


**Fig. 6.** Robustness to $\rho$ on the Mean number of connected peer groups $E\{K\}$

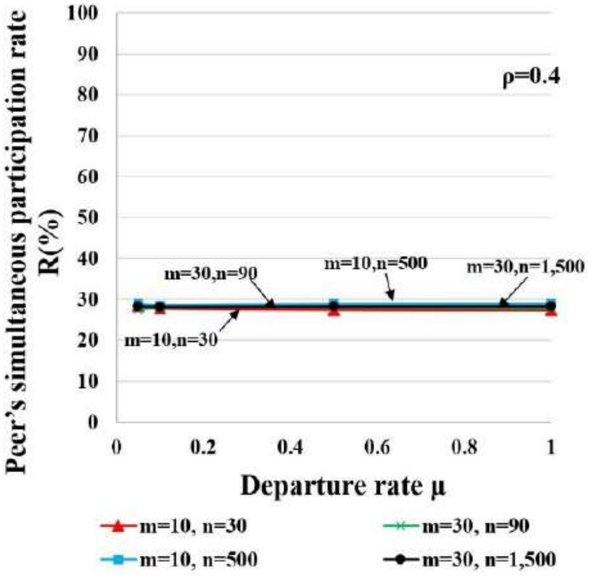


**Fig. 7.** Robustness to $\rho$ on the peer's simultaneous participation rate $R$ ($\rho$=0.4)

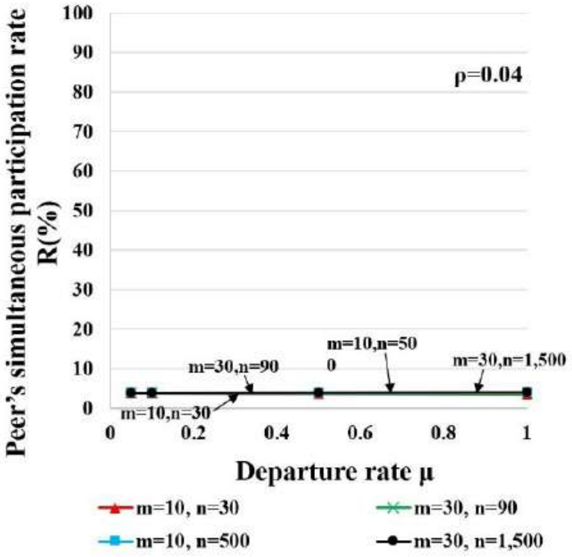


**Fig. 8.** Robustness to $\rho$ on the peer's simultaneous participation rate $R$ ($\rho$=0.04)

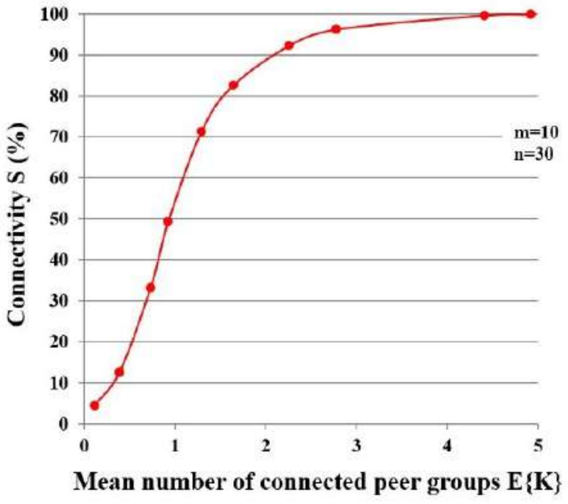


**Fig. 9.** Mean number of connected peer groups $E\{K\}$ vs. connectivity $S$

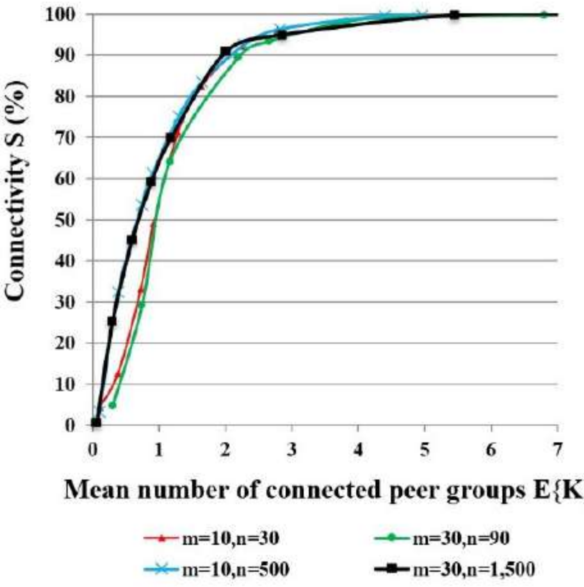


**Fig. 10.** Mean number of connected peer groups $E\{K\}$ vs. connectivity $S$ ($n/m$=3, $n/m$=50)

same when $n/m$ is the same. The larger $n/m$ is, the faster the convergence of the connectivity $S$ to the mean number of connected peer groups $E\{K\}$ becomes.

The simulation results provide the following findings.

- Even without investigating $\lambda$ and $\mu$ as input parameters, the connectivity $S$ is calculated from $E\{K\}$.
- The connectivity $S$ depends not only on $\rho$, but also on the ratio of $n$ to $m$ ($n/m$). Therefore, if $n/m$ is the same, the result will be the same.
- The convergence of the connectivity $S$ with respect to the mean number of connected peer groups $E\{K\}$ is faster when $n/m$ is large.
- Once $n$ and $m$ are determined, or once the ratio of $n$ to $m$ is determined, the mean number of connected peer groups $E\{K\}$, is obtained from the connectivity $S$. For the instance, figure 9 shows that the mean number of connected peer groups $E\{K\}$, is 2 to satisfy 90% of the connectivity $S$. In other words, a peer should participate in two peer groups in average.
- Once $m$, $S$, and $E\{K\}$ are determined, the required number of peers $n$ is derived.
- The peer's simultaneous participation rate $R$ depends on $\rho$.

## VI. CONCLUSION

In this paper, we clarified through simulations that the peer connectivity is calculated without using the arrival rate or the departure rate in the pure model. Specifically, we found that the connectivity can be obtained from the average number of peer groups in which peers participate at the same time. Since the conventional method of evaluating the connectivity requires a large amount of time and effort to derive the necessary arrival and departure rates, we conclude that we can derive connectivity in a relatively short time through the quantitative evaluation of the connectivity.

In the future, we plan to evaluate simulations with a performance evaluation model that considers combination of the peers and peer groups having various arrival and departure rates, i.e., the participation rate $\lambda_{ij}$ of peer$_i$ ($i$=1,2,3...$n$) in peer group$_j$ ($j$=1,2,3...$m$) and the departure rate $\mu_{ij}$.

## REFERENCES

[1] L. Gong, "JXTA: a network programming environment," *IEEE Internet Computing*, Vol. 5, No. 3, pp. 88-95, DOI: 10.1109/4236.935182, 2001.

[2] E. Halepovic, and R. Deters, "JXTA performance study," *2003 IEEE Pacific Rim Conference on Communications Computers and Signal Processing*, Vol. 1, pp. 149-154, DOI: 10.1109/PACRIM.2003.1235740, 2003.

[3] N. Yoshida, S. Urashita, Y. Hayashi et al., "SOBA framework: an application framework for broadband network environment," *The 2005 Symposium on Applications and the Internet*, pp. 296-303, DOI: 10.1109/SAINT.2005.59, 2005.

[4] SOBA Project Inc., "What is SOBA?," Available: https://www.soba-project.com.

[5] T. Hoshiai, K. Koyanagi, K. Sukhbaatar Birke, M. Kubota, H. Shibata and T. Sakai, "Semantic Information Network Architecture," *IEICE Transactions on Electronics, Information and Communication Engineers (B)*, Vol. J84-B, No. 3, pp. 411-424, 2001.

[6] T. Hoshiai, *Brokerless model and SIONet*, Telecommunications Association of Japan (Ohmsha), 2003.

[7] T. Hoshiai, "General Theory of P2P [I]: Challenge of Brokerless Model," *IEICE Journal*, Vol.87, No.9, pp.804-811, 2004.

[8] SIONet (NTT Information Communication Glossary), Available: https://www.ntt-review.jp/yougo/word.php?word_id=1928.

[9] J. Kishigami, S. Fujimura, D. Watanabe, M. Ohashi, and A. Nakahira, *Introduction to Blockchain Technology*, Morikita Publishing, 2017.

[10] Y. Kitahashi, Y. Hoshiai, H. Mitomo and T. Hoshiai: "A Proposal of Decentralized Collaboration Architecture on Brokerless Network and its Evaluation," *Transactions of Information Processing Society of Japan*, Vol. 47, No. 8, pp. 2669-2683, 2006.

[11] T. Hoshiai, Y. Kitahashi, Y. Hoshiai, T. Harada and H. Mitomo, "Performance Evaluation on Brokerless Networking Architecture," *IEICE Transactions on Electronics, Information and Communication Engineers (D)*, Vol. J88-D-I, No. 11, pp. 1608-1621, 2005.

[12] R. Cohen, K. Erez, D. Ben-Avraham and S. Havlin, "Resilience of the Internet to random breakdowns," *Phys. Rev. Lett.*, Vol. 85, No. 21, pp. 4626-4628, 2000.

[13] M. Nakauchi, "Facilitators of knowledge transfer among engineers: from the perspective of information acquirers," *Organization Science*, Vol. 48, No. 2, pp. 61-73, 2014.

[14] S. Saroiu, P. K. Gummadi and S. D. Gribble, "A measurement study of peer-to-peer sharing systems," *Proc. Multimed. Comput. Netw*, 2002 (MMCN '02), pp. 156-170, 2002.

[15] K. Leibnitz, T. Hossfeld, N. Wakamiya and M. Murata, "Peer-to-peer vs. client/server: Reliability and efficiency of a content distribution service," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 4516, pp. 1161-1172, 2007.

[16] S. G. Motlagh, J. Misic and V. B. Misic, "Impact of Node Churn in the Bitcoin Network," *IEEE Transactions on Network Science and Engineering*, Vol. 7, No. 3, pp. 2104-2113, 2020.

[17] M. A. Imtiaz, D. Starobinski, A. Trachtenberg and N. Younis, "Churn in the Bitcoin Network: Characterization and Impact," *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pp. 431-439, DOI: 10.1109/BLOC.2019.8751297, 2019.

[18] Y. Naito, C. Katsuki, N. Suehiro and T. Hoshiai, "Regional activation based on P2P network architecture," *2016 International Symposium on Nonlinear Theory and Its Applications*, pp. 423-426, 2016.

[19] Y. Naito, T. Hoshiai, K. Yoshimi, "Research of Evaluation on Regional Resource Network Model for Innovation Emergence," *Proc. of the 79th National Conference of Japan Society for Information and Management [Autumn]*, pp. 215-218, 2019.

[20] Y. Naito, T. Uemura and T. Hoshiai, "A Study on Connectivity Evaluation Among Peer Groups in Pure P2P Networks," *25th International Conference on Advanced Communication Technology (ICACT)*, pp. 298-302, DOI: 10.23919/ICACT56868.2023.10079579, 2023.

[21] Shinkin Central Bank, "Financial Research Information 2020-8 Trends in multitasking by sales branch staff in Shinkin Banks," *Management Strategy*, Vol. 32, pp. 3, 2020.

[22] Japan Tourism Agency, "Grandia Housen," *Case Studies on Productivity Improvement in the Lodging Industry*, Vol. 3, pp. 16, 2020. Available: http://www.shukuhaku-kaizen.com/wp-content/themes/shukuhaku_kaizen/img/case_studies_2020_01.pdf

[23] Japan Gerontological Evaluation and Research Institute, "Survey and Research Project on Outcome Indicators for Realization of Community Coexistence Society," *Examination of Process Evaluation for Establishment of Comprehensive Support System*, 2020.

[24] Nihon Keizai Shimbun, "Long Working Hours in Nursing Homes, 70% of Nursing Homes Have Two Shifts and Night Shifts Over 16 Hours," 2018. Available: https://www.nikkei.com/article/DGXMZO2943 1530W8A410C1CR8000/

**Yutaka Naito** was born in Japan, 1969. He received the Ph.D. degree in Engineering from Graduate School of Engineering, Sojo University, 2022. He is currently engaged in research on P2P network technology for regional revitalization as an assistant professor in the Faculty of Computer and Information Sciences, Sojo University.

**Takumi Uemura** was born in Japan, 1980. He received the Ph.D. degree in Engineering from Graduate School of Science and Technology, Kumamoto University, 2011. He is currently engaged in research on image processing and pattern recognition as an associate professor in the Faculty of Computer and Information Sciences, Sojo University.

**Takashige Hoshiai** was born in Japan, 1962. He received the Ph.D. degree in Engineering. He was a visiting researcher at Bell Telephone Laboratories from 1995 to 1997, and proposed the brokerless model in 1998, and invented the semantic information network architecture SIONet, which is the technology to realize the model. In 2011, he proposed Social Community Brand (SCB theory) that utilizes P2P for local revitalization. He is currently conducting research on regional revitalization and the emergence of regional innovation using SCB theory. In addition, he and his team invented a method of innovation emergence based on the concept of the board game GO. He is currently a president of Sojo University IoT/AI Center, a professor of the Faculty of Computer and Information Sciences, Sojo University, an invited researcher of Waseda University, a president of SCB Lab, and a principal of SCB Innovation Academy.

# Quick Blocking Operation of IDS/SDN Cooperative Firewall Systems by Reducing Communication Overhead

Akihiro Takai*, Yusei Katsura**, Nariyoshi Yamai*, Rei Nakagawa*, and Vasaka Visoottiviseth***

** Graduate School of Engineering, Tokyo University of Agriculture and Technology, Tokyo, Japan*

*** Graduate School of Science and Technology, Nara Institute of Science and Technology, Nara, Japan*

**** Faculty of Information and Communication Technology, Mahidol University, Nakhon Pathom, Thailand*

**s224164x@st.go.tuat.ac.jp, katsura.yusei.ky6@is.naist.jp, nyamai@cc.tuat.ac.jp, rnakagawa@go.tuat.ac.jp, vasaka.vis@mahidol.edu**

***Abstract*—An Intrusion Detection System (IDS) / Software Defined Networking (SDN) cooperative firewall system has attracted much attention recently because it has many advantages of dynamic network configuration with SDN and scalable IDS hosts. In the IDS/SDN cooperative firewall system, an SDN switch relays traffic between a client and a server and mirrors traffic from a client to an IDS host. The IDS host monitors the mirrored traffic and notifies the SDN switch to block malicious traffic according to the detection of the attack. At this point, malicious packets reach the server until the IDS detects the attack and notifies it. In this paper, we propose a method to speed up mirroring and notification by integrating IDS and SDN switch hosts as a method to shorten the blocking time and compare it with existing methods. The experimental system was constructed using Raspberry Pi3 B+ and 4B boards. As a result, it was confirmed that the proposed method completes the blocking operation faster than the existing method. We also investigated the breakdown of the blocking time to confirm the effect of the proposed method.**

***Keyword*— Firewall, Intrusion Detection System, OpenFlow, Software Defined Network**

## I. INTRODUCTION

To prevent computing devices of the organization from being compromised, a firewall, intrusion detection system (IDS), and intrusion prevention system (IPS) that can detect and block malicious files are essential. For quick and flexible network management and maintenance in large organization networks, the Software-Defined Network (SDN) should be used [1]. Upon integrating SDN with IDS, network administrators can deploy a flexible firewall system, namely the IDS/SDN cooperative firewall system [2], [3]. In this integration, the SDN switch that relays the bidirectional communication traffic of the target network mirrors the traffic to the IDS, and the IDS will detect anomaly packets that may be attacks from an outsider and inform the SDN controller. Once the SDN controller learns about attacks, it will create rules or policies to push to SDN switches to block malicious traffic coming into the network to be protected. In addition, multiple IDS servers can be used to provide load balance between IDS servers. Furthermore, by using the integration of SDN and IDS, not only a flexible firewall, but network administrators can also configure flexible network routes, for example, forwarding anomaly traffic to honeypots for further security analysis.

The large delay required for the period to mirror packets from the SDN switch to the IDS and notify attack detection from the IDS to the SDN controller (the blocking time) allows malicious packets to enter the protected network in the meantime. In the existing IDS / SDN cooperative firewall system, the OpenFlow, which is the leading implementation of the SDN concept, conventionally uses the REST API for notification from IDS [2]. However, because most IDS implementations do not support the REST API, the system must use the log monitoring tools to observe the change of IDS alert logs and configure the SDN controller to block this anomaly traffic via REST API, resulting in increasing the blocking time.

To reduce the blocking time, in the previous work, we have proposed a method to change the notification method from REST API to Syslog, which is faster because of reducing communication overhead [4]. To further reduce the blocking time along with using the fast Syslog notification method, this paper demonstrates a method of integrating IDS hosts and SDN switch hosts, which are detached physical hosts in existing systems, into a single physical host, so that mirroring and notification can be completed within a single physical host. Therefore, combining the integration method and the fast Syslog notification mitigates the communication overhead compared with the existing system of the separate physical hosts and reduces the blocking delay of malicious traffic. To evaluate the effectiveness of the system, we conducted an experiment to compare the blocking time of the existing system in which the IDS and SDN switch hosts are

Manuscript received December 8, 2022. This work is a follow-up of the invited journal to the accepted & presented paper entitled “Quick blocking operation of firewall system cooperating with IDS and SDN” of the 24th International Conference on Advanced Communication Technology (ICACT2022).

Akihiro Takai, he is a master’s student in the Graduate School of Engineering at Tokyo University of Agriculture and Technology, Japan. (e-mail: s224164x@st.go.tuat.ac.jp)

Yusei Katsura, he is a Ph.D. student in the Graduate School of Science and Technology at Nara Institute of Science and Technology, Japan. (e-mail: katsura.yusei.ky6@is.naist.jp)

Nariyoshi Yamai, he is a professor in the Graduate School of Engineering, Tokyo University of Agriculture and Technology, Japan. (corresponding author, phone: +81-42-388-7695, e-mail: nyamai@cc.tuat.ac.jp)

Rei Nakagawa, he is an assistant professor in the Graduate School of Engineering at Tokyo University of Agriculture and Technology, Japan. (e-mail: rnakagawa@go.tuat.ac.jp)

Vasaka Visoottiviseth, she is an associate professor in the Faculty of Information and Communication Technology at Mahidol University, Thailand. (e-mail: vasaka.vis@mahidol.edu)

detached and the proposed system in which the IDS and SDN switch hosts are integrated. In the experiment, two kinds of experimental systems, the existing method and the proposed method, were constructed with real devices, and a total of four blocking times were measured by the combination of two kinds of notification methods, REST API and Syslog. We make following contributions.

- Proposal of quick blocking operation of integrating IDS hosts and SDN switch hosts using the fast notification method.
- Implementation of the proposed method using Raspberry Pi 4B and 3B+ as devices, Suricata as IDS, and Open vSwitch (OvS) as SDN switch implementation.
- Experimental verification of the reduction of blocking time by host integration in combination with two types of notification methods, REST API and Syslog.

## II. Background and Existing Works

### *A. Software Defined Network*

In the traditional network, once any network policy changes, network administrators have to carefully configure all switches manually. It requires a great deal of network maintenance. To solve this problem, the SDN is introduced. In the SDN concept, network tasks are divided into the control plane and the data plane. The control plane, which is responsible for making decisions on how packets should be handled, is the task of the SDN controller. On the other hand, the data plane or forwarding plane, which is responsible for handling packets based on the instructions from the control plane, is the task of the SDN switch. Instructions from the SDN controller can be either forwarding or dropping packets on the SDN switch. It can manage or control the forwarding table residing on SDN switches. On the other hand, SDN switches can focus on accelerating the packet forwarding.

By using SDN, administrators can centrally maintain the network policy via the SDN controller. Instead of manually updating each switch, SDN enables the network administrators to distribute the policy evenly across multiple switches. This can simplify network management.

In SDN, there are Northbound and Southbound APIs that are APIs operating between data plane, control plane and application plane. The Northbound APIs are available on an SDN controller and allow applications or the application plane to interact with the controller, which is the control plane. Applications and services are, for example, load-balancers and firewalls. On the other hand, the Southbound APIs are available between the SDN controller in the control plane and other forwarding devices, e.g. switches, and routers, which are the data plane or forwarding plane. By using the Southbound APIs, administrators can adjust the network according to the change requirements.

If you want to submit your file with one column electronically, please do the following:

#### *1) OpenFlow*

OpenFlow is a well-known technology maintained by the Open Networking Forum (ONF) [5] that implements the SDN concept. The OpenFlow protocol is a set of specifications that provides the southbound interface and defines how the controller interacts with the data plane. The newest version of the OpenFlow Switch specification is version 1.5.1. Each OpenFlow switch will store information about the flow entry received from the controller into the flow table. Each flow entry contains an explicit action to handle each flow.

On an OpenFlow switch, there are two main components: the switch-agent and the data plane. The switch-agent uses the OpenFlow protocol to communicate with one or more controllers. Moreover, it communicates with the data plane using the internal protocol.

Every OpenFlow message begins with the same header structure containing version, type, length, and transaction id (xid). The OpenFlow message types can be, for example, FlowMod, PacketIn, FlowRemoved, PacketOut, and StatsReq. The FlowMod message is used to allow the controller to modify the flow entries on the OpenFlow switch. On the other hand, the PacketIn message type is used by the switch to send incoming packets to the OpenFlow controller. Normally, there are two cases where PacketIn message type is used: (1) there is an explicit action for this behavior specified in the flow entry and (2) the flow does not match any flow entry in the table. Moreover, to make the OpenFlow controller understand that the traffic comes from which switch, each switch will be assigned a Datapath ID.

There are many OpenFlow controller software, for example, Ryu, Treman, Opendaylight, etc. In this research, we selected Ryu as the SDN controller software.

#### *2) Ryu*

Ryu [6] is a framework that implements the OpenFlow controller and is developed in Python. Once the Ryu controller receives an OpenFlow message from a switch, it will trigger an event. An event handler contains an event class and the state of the switch. States of OpenFlow switch can be, for example, the HANDSHAKE_DISPATCHER, which is the initial state that exchanges the HELO message, and the MAIN_DISPATCHER, which is the normal state. When the controller receives a PacketIn message from the switch, the EventOFPPacketIn event class will be called. For example, with the API set_ev_class(ofp_event.EventOFPPacketIn, MAIN_DISPATCHER), we can define the process when the OpenFlow switch is in the normal operation state and receives a PacketIn message [7].

### *B. Related Works*

Here, we surveyed related works that utilize Snort IDS [8] and SDN together to dynamically filter anomaly traffic.

Nam *et al.* studied SDN security enhancement using open-source IDS/IPS Suricata [9][10]. They proposed requirements for implementing SDN security. They mentioned whether SDN solutions, Suricata IDS/IPS, automated intrusion prevention, and mirroring are necessary or not when they want to implement a firewall, network scan detection, abnormal traffic detection, intrusion detection, and intrusion prevention. They concluded that to implement intrusion prevention they need Suricata IDS/IPS, automated intrusion prevention, as well as mirroring because the SDN alone cannot handle this. However, no implementation and performance were shown in their paper.

Hendrawan *et al.* studied the performance degradation when integrating the SDN architecture with an IDS, either the signature-based Snort or the anomaly-based Bro [11]. The throughput, delay, and packet loss, which are required performance metrics of quality of service (QoS), are observed. The results of their experiment on the mininet virtual network [12] showed that integrating SDN with Bro IDS gave better performance than using Snort IDS in all

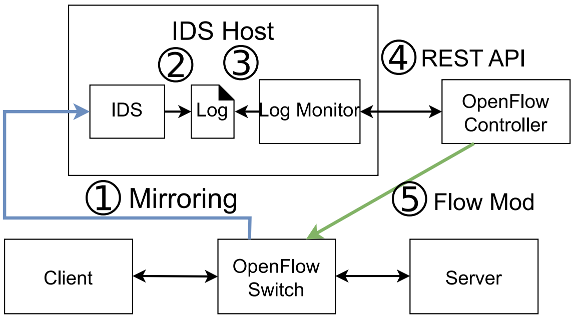


Fig. 1. System architecture of the Original IDS/SDN Cooperative Firewall System.

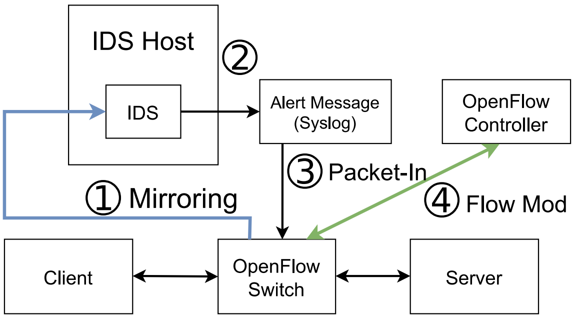


Fig. 2. System architecture of the Quick block operation by Syslog notification.

aspects: throughput, delay, and packet loss. However, the delay is not much different, while the CPU usage and memory usage of Snort are lower than those of Bro.

R. Sutton et al. designed and developed a system that utilized an SDN to divert traffic from some suspicious traffic multiple Snort IDS machines for inspection [13]. The authors developed 'PySnorter,' a Python tool for routing traffic to dedicated Snort machines. Once Snort alerts the malicious traffic, it pulls the information from the alert and generates a REST API command that is then sent to the SDN controller. For the experiments, they implemented a virtualized network using GNS3, which is the graphical version of Network Simulator-3, and used Open vSwitch. However, latency was not measured.

### C. *The Original IDS/SDN Cooperative Firewall System*

In firewall systems cooperating with IDS and SDN, the IDS generally do not have the capability to send alert messages directly to the SDN controller, but instead has the capability to send them to an alert handler as Syslog messages or record them into a log file. Typically, Syslog messages are in the common event format (CEF) or log event extended format (LEEF) to facilitate analysis. However, similar to what was proposed in the [12], in the case of a firewall system recording alert messages in a log file and using OpenFlow as an SDN platform, the typical behavior of the firewall system to block malicious packets consists of the following six steps, as illustrated in Fig. 1.

(1) When the OpenFlow switch receives packets from the client that are destined to the server, it will delay the forwarding process to the destination and copy the traffic to IDS via port mirroring.

(2) Once the IDS receives those packets, it will analyze and detect malicious packets. If the IDS finds malicious packets, it will alert and write the information to the log file.

(3) The log monitoring tool continuously monitors that log file.

(4) When the log monitoring tool detects a change in the log file, that is, a new alert message written in the log file, it will use the REST API to send the information of that malicious packet to the OpenFlow controller.

(5) The OpenFlow controller uses the received information to create a flow entry that determines how an OpenFlow switch must behave when it receives packets from that malicious flow. Then, the OpenFlow controller will use the FlowMod message to send that Flow entry to the OpenFlow switch.

(6) The OpenFlow switch updates the flow table based on the information specified in the FlowMod message. Then, it can process packets that are delayed in step 1.

In this case, because most IDS do not support REST APIs, they have to use log monitoring tools to monitor log change and use a command line tool such as 'curl' to transfer REST APIs. Therefore, a processing overhead will occur. Moreover, REST APIs use HTTP methods and run over TCP protocol.

### D. *Quick Block Operation by Syslog Notification*

To reduce communication overhead due to REST API, we have proposed the fast notification method for the IDS/SDN cooperative firewall system as illustrated in Fig. 2. There are five steps to perform.

(1) When the OpenFlow switch receives packets from the client that are destined to the server, it will delay the forwarding process to the destination and copy the traffic to IDS via port mirroring.

(2) If the packet matches with any signatures in the IDS, the IDS will send the alert message to the OpenFlow switch by using, for example, the Syslog message or the SNMP trap.

(3) When the OpenFlow switch receives the alert message sent via UDP, it will forward the message as the PacketIn message type to the OpenFlow controller.

(4) The OpenFlow controller uses the received information to create a flow entry that determines how an OpenFlow switch must behave when it receives packets from those malicious flows. Then, the OpenFlow controller will use the FlowMod message to send that Flow entry to the OpenFlow switch.

(5) The OpenFlow switch updates the flow table based on the information specified in the FlowMod message. Then, it can process packets that are delayed in step 1.

Note that this method can reduce one step compared with the existing method mentioned in Section II B, because we send the alerts as the Syslog message directly to OpenFlow switch, but the existing method needs to write the alerts in the log file first and use the log monitoring tools to detect the log change. The characteristics of REST API and Syslog as notification methods can be summarized as follows.

- REST API

The REST API submissions use an external program Swatchdog to monitor logs and curl to send packets. The POST method of the REST API is used to send notifications as well as host blocking. Since TCP is used for communication, it is expected that there will be overhead to establish the connection.

- Syslog

Most IDSs can notify anomalies using Syslog. Syslog can communicate via either UDP or TCP. In an IDS/SDN cooperative firewall system, it is considered better to use UDP for low latency communication.

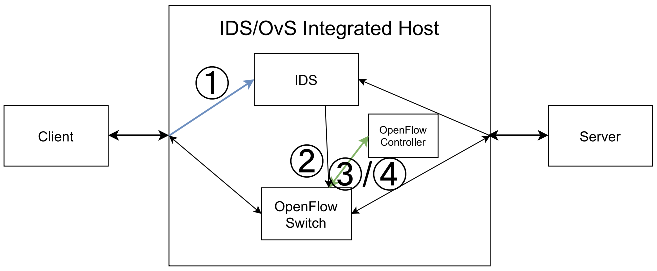


Fig. 3. System architecture of the proposed work.

TABLE I
SOFTWARE SPECIFICATION

| Device | Software |
|---|---|
| OpenFlow controller | Ryu 2.7 |
| OpenFlow switch | Open vSwitch 2.10.1 (OvS) |
| Intrusion Detection System | Suricata 6.0.6 |
| Log monitoring tools | swatch |

## III. PROPOSED WORK

To further reduce the blocking time, we adopt an approach to reduce the communication overhead between separated physical hosts of IDS and SDN, as well as that of existing notification method, REST API. Therefore, we propose a method to reduce the blocking time by running an IDS and an SDN switch implementation, such as Open vSwitch, on a single device, thereby reducing the communication overhead in the mirroring in step 1 and the notification in step 4, as illustrated in Fig. 3. In the IDS/SDN cooperative firewall system experimented in the previous study [4], the host acting as an SDN switch and the IDS host are different devices. Therefore, we propose a method to reduce the blocking time by running an IDS and an SDN switch implementation, such as Open vSwitch, on a single device, thereby reducing the communication overhead in the mirroring in step 1 and the notification in step 4. An IDS/SDN cooperative firewall system that integrates an IDS host and an SDN switch host in a single device is called an integrated system, while one that has different hosts for each is called a detached system.

In our implementation, we implement two approaches: (1) the existing detached system and (2) our proposed integrated system. Details of them are described below.

## IV. IMPLEMENTATION

For the implementation, as mentioned earlier in Section II, we select Ryu as the OpenFlow controller. For the specification of OpenFlow, we use OpenFlow version 1.3. For the IDS, we select Suricata, which is a famous Open Source Software (OSS). Suricata has a large community and has more than thousands of detection signatures available. Table I summarizes the software and its version used in our experiments.

Moreover, we used two Raspberry Pi 3B+ boards and a Raspberry Pi 4B board to emulate each device. First, Raspberry Pi 3B+ is used to implement a client host and a server host, while Raspberry Pi 4B is used for implementing IDS/OvS integrated host. In the detached system, we use Raspberry Pi 3B+ for the client host, server host and IDS host, while Raspberry Pi 4B is used for implementing OvS host.

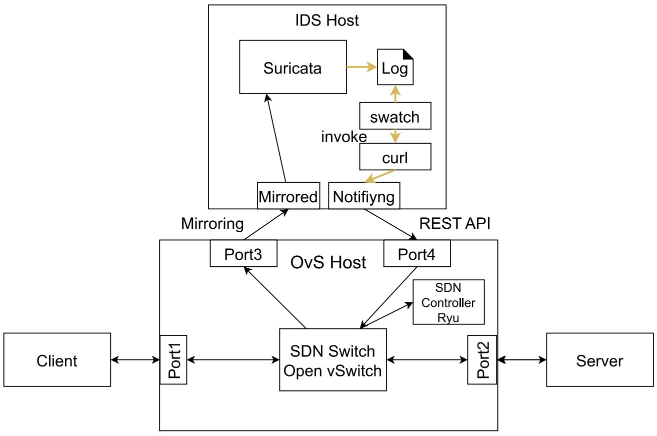


Fig. 4. Implementation of the detached system.

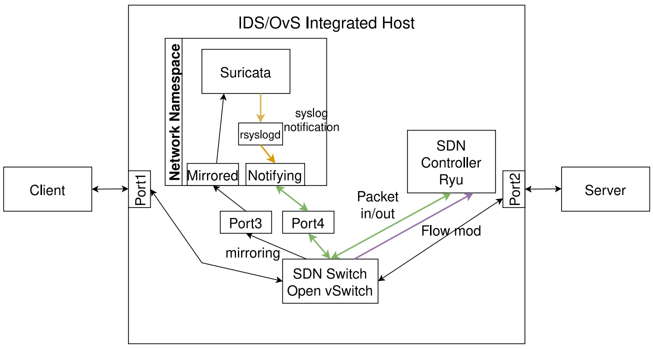


Fig. 5. Implementation of the integrated system.

### A. Implementation of the Detached System

First, we implement the existing approach that uses Suricata to write alerts to the log file and uses swatch to monitor the log file. As shown in Fig. 4, in this approach Suricata and the swatch are running on the same Raspberry Pi board. Moreover, the Raspberry Pi that emulates both Ryu controller and Open vSwitch is equipped with four ports: Port 1 connecting to the client, Port 2 connecting to the server, Port 3 which is a one-way directional port just to forward packets to Suricata to analyze, and Port 4 which is also a one-way directional port for Ryu to receive REST APIs over TCP from the swatch utility. Moreover, there are only FlowMod messages sent from Ryu to OvS when the Ryu controller wants to modify the flow table on OvS.

### B. Implementation of the Integrated System

Fig. 5 shows a diagram of the IDS and SDN switch. A Linux network namespace is running on the IDS/OvS integrated host. Using the Network Namespace function, IP-related processes can be divided into multiple processes within a single Linux unit. Suricata runs within a Network Namespace on the integrated host. This Network Namespace has two Ethernets: one corresponds to port 3 and is used to receive packets from the client and server ports (Ports 1 and 2), and the other corresponds to port 4 and is used by Suricata to notify the outside of Network Namespace.

First, packets arriving from the client are forwarded to the server and simultaneously mirrored to the Network Namespace which the IDS operates. Packets arriving from

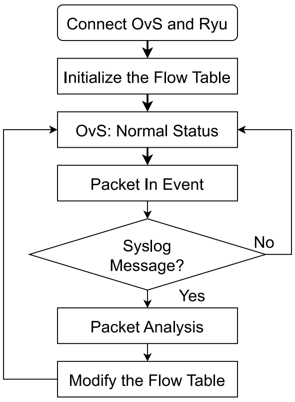


Fig. 6. Flowchart of the process inside the Ryu controller.

TABLE II
FLOW TABLE IN THE INITIAL STATE

| Match | Action |
|---|---|
| in_port 1 | OutPut (Port 2, Port 3) |
| in_port 2 | OutPut (Port 1, Port 3) |
| in_port 4 | OutPut (Controller) |

TABLE III
FLOW TABLE AFTER GOT ATTACKS

| Match | Action |
|---|---|
| in_port 1 | OutPut (Port 2, Port 3) |
| in_port 2 | OutPut (Port 1, Port 3) |
| in_port 4 | OutPut (Controller) |
| ip_src 192.168.10.21<br>tcp_dst 80<br>eth_type = 0x0800 | Drop |

the server are also forwarded to the client and mirrored to the network namespace. The IDS monitors the packets and notifies the SDN controller across the network namespace using Syslog if any malicious packet is detected. The SDN controller receives the notification and modifies the Open vSwitch flow table to drop the packets from the host that sent the packets deemed to be malicious. that sent the packets deemed to be malicious. This allows the mirroring and notification procedures of the blocking operation to be completed in a single device.

Fig. 6 shows the flowchart of the processes inside our Ryu controller. Once the OvS connects with the Ryu controller, it will initialize the flow table as shown in Table II and set the state of the OvS as normal. When the Ryu controller receives the PacketIn message, it will check whether the message is a Syslog message or not. If so, it will analyze the message and modify the flow table. An example of a flow table that is modified after receiving DDoS attacks is shown in Table III. A new flow entry is added to the flow table specifying the “Drop” action when the packet contains the header that matches the specified condition.

## V. EVALUATION RESULTS

For evaluation, we compare our proposed work with the existing approach and the proposed approach described in Section IV A and Section IV B, respectively. To measure the blocking time, we observe the packet timestamp using the tcpdump command. To investigate the breakdown of blocking time, we define the communication overhead (i.e., the blocking time) using the following periods of processing in the IDS/SDN cooperative firewall system.

TABLE IV
BLOCKING TIME OF EACH APPROACH

| | REST API | Syslog |
|---|---|---|
| Detached System | 75.6 ms | 9.6 ms |
| Integrated System | 67.3 ms | 4.6 ms |

1. Mirroring — the time taken to mirror packets from OvS to IDS.
2. Log monitoring — the time it takes for Swatch to detect IDS writes to the log.
3. Start script — time required for Swatch to start the REST API submission script.
4. Notification — the time taken to notify in REST API.
5. Syslog — the time taken for IDS to notify in Syslog.
6. Packet In/Flow Mod — the time it takes for OvS to packet in a Syslog packet and invoke Flow Mod processing.

Both existing and proposed approaches measure the blocking time from the time OvS host receives a malicious packet until SDN controller Ryu receives a Syslog or REST API Packet-in packet and calls the Flow Mod process. Table IV summarizes the blocking time for each method. First, we note the performance differences due to differences in the notification methods. When comparing Figs. 7 and 8, and Figs. 9 and 10, respectively, using Syslog as the notification method, the overhead of log monitoring, REST API sending script invocation, and notification processing required when using REST API was reduced, and the blocking operation could be performed at a speed of approximately 60 ms. In particular, we observed that it was taking a long time to send REST APIs using curl.

Next, we discuss the differences between the results of the detached and integrated systems. Comparing Figs. 7 and 9, and Figs. 8 and 10, respectively, the integrated system completed the blocking operation faster, and the breakdown suggests that this is due to faster mirroring, notification, Packet-in, and Packet-out. Furthermore, comparing the detached and integrated systems in the results using Syslog as the notification method, the time spent for communication was approximately 0.6 ms for the detached system and approximately 0.3 ms for the integrated system. When using the REST API, the time spent for mirroring was approximately 2.2 ms for the detached system and 0.27 ms for the integrated system. These results indicate that the proposed method reduces the communication overhead. Although there are differences in the boards on which the IDS is running, the fact that mirroring and notification are completed within a single host is one of the factors contributing to the reduction in blocking time. To more accurately observe the effect of the proposed method, it is necessary to conduct experiments on a unified board on which the IDS operates.

## VI. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a method to reduce blocking time by integrating IDS hosts and SDN switch hosts along with the fast notification method and we have confirmed the effectiveness of the proposed method through experiments.

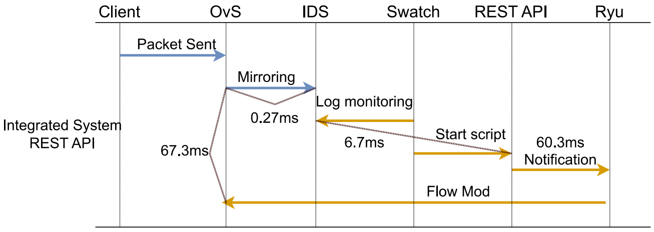


Fig. 7. Blocking time of Integrated System when using REST API.

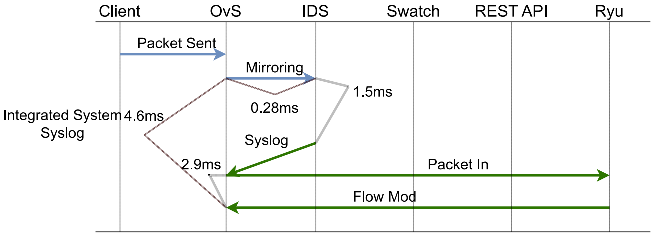


Fig. 8. Blocking time of Integrated System when using Syslog.

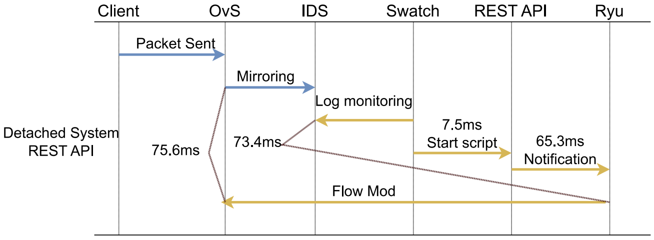


Fig. 9. Blocking time of Detached System when using REST API.

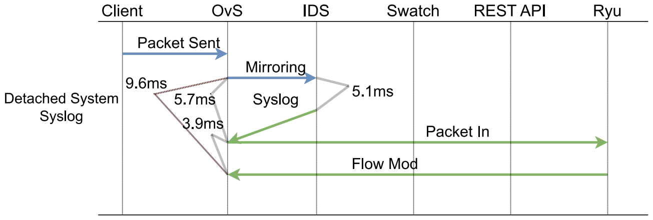


Fig. 10. Blocking time of Detached System when using Syslog.

In the experiments, the blocking time was measured by combining two types of notification methods and two types of IDS/SDN cooperative firewall systems: the integrated system proposed here and the detached system. The experimental results confirm that the notification method using Syslog is faster than the REST API because the Syslog notification method removes the process of monitoring IDS log for REST API. In addition, integrating IDS and SDN hosts reduced the communication overhead between the IDS and SDN, which were physically separated in conventional, and as a result, further reduced the blocking time at no load on the system. We also confirmed that the reduction of communication overhead, which is the goal of the proposed method, contributes to faster blocking operation by examining the breakdown of the blocking time. However, the proposed method and the existing method differ in the host on which the IDS operates, and we have not confirmed how the blocking times of the two systems change when a traffic load is added to them. Since the integrated system is more susceptible to the huge network traffic load than the detached system since IDS and OvS are running on a single host, the performance of the integrated system is expected to be degraded under high load. Therefore, for future work, an immediate verification method is required.

Moreover, the IDS/SDN cooperative firewall system using the IDS/OvS integrated host has an advantage over existing IDS/SDN cooperative firewall systems in terms of parallelization of OvS hosts. This is because the IDS and OvS hosts are integrated, which simplifies the network configuration. Parallelization of IDS and OvS hosts is effective to suppress the increase in shutdown time of IDS/SDN linked firewall system under load environment. In the future, we will construct an OvS host parallelized IDS/SDN cooperative firewall system using an IDS/OvS integrated host and compare its performance with existing parallelized configurations and examine its qualitative cost.

## REFERENCES

[1] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, “OpenFlow: enabling innovation in campus networks”, *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp 69–74, April 2008.

[2] P. Zanna, B. O'Neill, P. Radcliffe, S. Hosseini, and M. S. Ul Hoque, “Adaptive threat management through the integration of IDS into Software Defined Networks”, in *Proc. 2014 Int. Conf. and Workshop on the Network of the Future (NOF)*, pp. 1-5, 2014.

[3] Y. Katsura, H. Kimiyama, T. Tsutsumi, N. Yonezaki, J. Ichikawa, and M. Maruyama, “Proposal of real-time brute-force attack detection and blocking system using software switch”, *IEICE Technical Report*, vol. 118, no. 465, NS2018-272, pp. 461-464, 2019 (in Japanese).

[4] Y.Katsura, P.Sakarin, N.Yamai, H. Kimiyama, and V. Visoottiviseth: “Quick blocking operation of firewall system cooperating with IDS and SDN,” in *Proc. 24th Int. Conf. Advanced Communications Technology (ICACT 2022)*, pp.393-398, February, 2022.

[5] The Open Networking Foundation, “SDN Technical Specifications” [Online]. Available: https://opennetworking.org/software-defined-standards/specifications/

[6] Ryu SDN Framework Community, “Ryu SDN Framework” [Online]. Available: https://ryu-sdn.org/

[7] Nippon Telegraph and Telephone Corporation Revision d6cda4f4, “Ryu application API” [Online]. Available: https://ryu.readthedocs.io/en/latest/ryu_app_api.html

[8] Cisco and/or its affiliates, “Snort – Network Intrusion Detection & Protection System” [Online]. https://snort.org/

[9] The Open Information Security Foundation (OISF), “Home – Suricata” [Online]. Available: https://suricata.io/

[10] K. Nam and K. Kim, “A Study on SDN security enhancement using open source IDS/IPS Suricata,” in *Proc. 2018 Int. Conf. Information and Communication Technology Convergence (ICTC)*, pp. 1124-1126, Jeju, South Korea, 2018.

[11] H. Hendrawan, P. Sukarno, and M. A. Nugroho, “Quality of Service (QoS) Comparison Analysis of Snort IDS and Bro IDS Application in Software Define Network (SDN) Architecture,” in Proc. 2019 7th Int. Conf. Information and Communication Technology (ICoICT), pp. 1-7, Kuala Lumpur, Malaysia, 2019.

[12] Open Networking Foundation, “MININET - Open Networking Foundation” [Online]. Available: https://opennetworking.org/mininet/

[13] R. Sutton, R. Ludwiniak, N. Pitropakis, C. Chrysoulas and T. Dagiuklas, “Towards an SDN Assisted IDS,” in *Proc. 2021 11th IFIP Int. Conf. New Technologies, Mobility and Security (NTMS)*, pp. 1-5, Paris, France, 2021.

**Akihiro Takai** was born in Japan in 1999 and received his B.E. from Tokyo University of Agriculture and Technology (TUAT), Japan in 2022. Currently, he is a master's student at Tokyo University of Agriculture and Technology, Japan. His research interests include computer networks.

**Yusei Katsura** was born in Japan in 1996 and received his B. Info. Env. degree from Tokyo Denki University, Japan in 2019, and his M. S. degree in computer engineering from Nara Institute of Science and Technology (NAIST), Japan in 2022. He was a research student at Tokyo University of Agriculture and Technology (TUAT), Japan in 2019-2020. He is currently a Ph.D. student at Nara Institute of Science and Technology. His research interests include computer networks.

**Nariyoshi Yamai** was born in Japan in 1961 and received his B.E. and M.E. degrees in electronic engineering and Ph.D. degree in information and computer science from Osaka University, Japan in 1984, 1986, and 1993, respectively. Currently, he is a professor at the Institute of Engineering, Tokyo University of Agriculture and Technology (TUAT), Japan. His research interests include distributed systems, network architecture, network security, and the Internet.

**Rei Nakagawa** was born in Japan in 1993, received his B.S. and M.S. degrees from the Tokyo University of Science, Japan, in 2016, and 2018 respectively, and a Ph.D. degree in informatics and engineering from the University of Electro-Communications (UEC), Japan, in 2021. He has been an assistant professor at the Institute of Engineering, Tokyo University of Agriculture and Technology (TUAT), Japan since April 2021. His research interests include network architecture, video streaming technology, software defined network, and information centric networking.

**Vasaka Visoottiviseth** was born in Thailand in 1975, received her M.E. and B.E. degrees from Tokyo University of Agriculture and Technology (TUAT), Japan in 1999 and 1997, respectively, and received her Ph.D. degree in computer engineering from Nara Institute of Science and Technology (NAIST), Japan in 2003. Currently, she is an associate professor at Mahidol University, Thailand. Her current research interests include mobile and wireless computing, network security, and digital forensics.

# Automated Vulnerability Assessment Approach for Web API that Considers Requests and Responses

Yuki Ishida*, Masaki Hanada**, Atsushi Waseda**, and Moo Wan Kim***

* Graduate School of Informatics, Tokyo University of Information Sciences, Japan

** Department of Informatics, Tokyo University of Information Sciences, Japan

*** TA Tech., Japan

h22001iy@edu.tuis.ac.jp, mhanada@rsch.tuis.ac.jp, aw207189@rsch.tuis.ac.jp, ykim5jp@ybb.ne.jp

***Abstract*— In recent years, Web Application Programming Interfaces (Web APIs) have been extensively used in numerous web applications. However, the number of attacks exploiting Web API vulnerabilities has been rapidly increasing. The Open Web Application Security Project (OWASP) published guidelines known as the OWASP API Security Top 10 to mitigate the risks associated with these vulnerabilities. The guidelines identify the top 10 most critical security risks in Web APIs and provide remediation guidance to help developers. Although developers are required to address these vulnerabilities according to these guidelines, traditional vulnerability assessment tools may not perform adequately when used to assess Web API vulnerabilities. Manually addressing these is difficult because there are a large number of endpoints and parameters in Web APIs using traditional vulnerability assessment tools. To address this issue, we propose a method for automatically conducting Web API vulnerability assessments by utilizing references, requests, and responses for Web APIs. In the evaluation experiment, we showed that the proposed method can detect authorization-related vulnerabilities in the Web APIs of vulnerable testing environments and well-known Content Management Systems, such as Wordpress, Ghost CMS, and Joomla.**

***Keywords*— Web API, Vulnerability Assessment, Automation Analysis, Security**

## I. Introduction

In recent years, the widespread adoption and use of Web Application Programming Interfaces (Web APIs) have greatly enhanced the convenience of web systems. However, the vulnerabilities associated with Web APIs are increasingly subject to attacks. Akamai Inc. reported that more than 11 billion attacks occurred between January 2020 and June 2021 [1]. Specifically, in June 2021, 113.8 million attack traffic events were observed in a single day, and attacks targeting Web APIs are on the rise. The proactive elimination of Web API vulnerabilities is very important to guarantee a safe and secure cyberspace for general users.

To mitigate the risks associated with these vulnerabilities, the Open Web Application Security Project (OWASP) published guidelines known as the OWASP API Security Top 10 [2]. The guidelines identify the top 10 most critical security risks in Web APIs and provide remediation guidance to help developers. Although developers are required to address these vulnerabilities according to such guidelines, it is difficult to address them manually because of the large number of endpoints and parameters in Web APIs.

To solve these issues, traditional web applications have employed a variety of tools, such as OWASP ZAP (Zed Attack Proxy), for vulnerability assessment. However, because these tools are primarily designed for traditional web applications and content, they may not perform adequately when used to assess Web API vulnerabilities. Specifically, developers are required to manually configure many parameters of traditional tools (e.g., OWASP ZAP) for vulnerability assessment because traditional tools do not consider the logic and characteristics of Web APIs. A lack of understanding of the logic and characteristics of Web APIs causes omissions or mismatches in endpoints or parameters. Consequently, this issue leaves risks of Web API vulnerabilities.

In this study, we propose a method for automatically conducting a Web API vulnerability assessment by utilizing references, requests, and responses for Web APIs to detect authorization-related Web API vulnerabilities. The proposed method first generates endpoints and parameters using Web API references, and requests for validating the Web API are subsequently sent to the generated endpoints. Next, new endpoints and parameters are generated using the responses from the endpoints, and vulnerability assessment requests are sent to the generated endpoints. Finally, when the HTTP status codes of the responses from the endpoints satisfy certain conditions, the proposed method determines that they are valid endpoints and/or parameters and do not contain vulnerabilities.

Manuscript received on Sep. 27, 2023. This work is a follow-up on the accepted and presented paper titled ”An Automated Vulnerability Assessment Approach for WebAPI that Considers Requests and Responses,” at the 24th International Conference on Advanced Communication Technology (ICACT2022).

Yuki Ishida, He is a Ph.D. student in the Graduate School of Informatics, Tokyo University of Information Sciences, 4-1, Onaridai, Wakaba, Chiba, Japan (e-mail: h22001iy@edu.tuis.ac.jp.)

Masaki Hanada is with Department of Informatics, Tokyo University of Information Sciences, 4-1, Onaridai, Wakaba, Chiba, Japan (corresponding author, phone: +81-43-236-1224, e-mail: mhanada@rsch.tuis.ac.jp).

Atsushi Wasedais is with Department of Informatics, Tokyo University of Information Sciences, 4-1 Onaridai, Wakaba-ku, Chiba, Japan (e-mail: aw207189@rsch.tuis.ac.jp)

Moo Wan Kim is with TA-Tech, Japan (e-mail: ykim5jp@ybb.ne.jp)

ISSN: 2288-0003 (Online)

TABLE I
Example of RPC API and REST API

| Operation | Type of API | HTTP Method | Endpoint | Request Parameters (JSON Format) |
|---|---|---|---|---|
| Read | RPC | POST | /getItem | {”id”:”1”} |
| | REST | GET | /items/1 | None |
| Create | RPC | POST | /createItem | {”name”: ”Pen”, ”price”: 100} |
| | REST | POST | /Items | {”name”: ”Pen”, ”price”: 100} |

In the evaluation experiment, we showed that the proposed method can detect authorization-related vulnerabilities in the Web APIs of vulnerable testing environments and Content Management Systems (CMSs).

The remainder of this paper is organized as follows. Section II presents an overview of Web APIs. Section III presents the OWASP API Security Top 10. Traditional vulnerability assessment tools are presented in Section IV. Section V describes the details of the proposed method. Section VI presents the experimental environment. Section VII presents the experimental results. Finally, Section VIII concludes the paper.

## II. Web APIs

A Web APIs is an API that is accessed using the HTTP protocol and typically uses either Extensible Markup Language (XML) or JavaScript Object Notation (JSON) formats to encode data. Web APIs are extensively utilized to enhance the communication between web systems and a variety of web applications. Currently, despite the lack of a universal standardized format for Web APIs, Remote Procedure Call API (RPC API) and Representational State Transfer API (REST API) have been widely used as representative architectural styles in Web API design.

### *A. RPC API*

The RPC API is based on the remote procedure call (RPC). The representative implementations of the RPC API are XML-RPC and JSON-RPC; whereas JSON-RPC uses a lightweight JSON format to encode data, XML-RPC uses a heavier XML format.

### *B. REST API*

The REST API is a simple method for accessing Web resources. The REST API endpoint is a URL that utilizes HTTP methods such as POST, GET, POST, PUT, and DELETE to execute the CRUD (Create, Read, Update, and Delete) operations. It primarily focuses on providing resources from the server to clients. Similar to the RPC API, the REST API can use either XML or JSON to encode data.

Table I presents an example of HTTP requests to perform read and create operations in RPC API and REST API.

In this study, we targeted the REST API because it has become a mainstream Web API, and the Web APIs of many CMSs are provided by the REST API.

TABLE II
List of OWASP API Security Top 10

| Risk ID | Name of Vulnerability items | Category |
|---|---|---|
| API 1:2019 | Broken Object Level Authorization | Category 1 |
| API 2:2019 | Broken User Authentication | Category 1 |
| API 3:2019 | Excessive Data Exposure | Category 2 |
| API 4:2019 | Lack of Resources & Rate Limiting | Category 3 |
| API 5:2019 | Broken Function Level Authorization | Category 1 |
| API 6:2019 | Mass Assignment | Category 2 |
| API 7:2019 | Security Misconfiguration | Category 3 |
| API 8:2019 | Injection | Category 4 |
| API 9:2019 | Improper Assets Management | Category 3 |
| API 10:2019 | Insufficient Logging & Monitoring | Category 4 |

## III. OWASP API Security Top 10

OWASP is a nonprofit foundation that works to improve the security of web applications. OWASP published guidelines known as OWASP API Security Top 10 to mitigate the risks associated with Web API vulnerabilities; these guidelines aimed to provide guidance on the most important security risks to consider when developing and exposing APIs. The guidelines include recommended countermeasures and outline scenarios in which Web API vulnerabilities may occur.

Table II lists the OWASP API Security Top 10, which can be classified into four primary categories. The *category 1* in Table II pertains to access control vulnerabilities, such as the unauthorized use of Web APIs by general users, owing to inadequate authorization settings intended for administrators. The *category 2* includes vulnerabilities related to data handling. For example, a Web API server may transmit data to the client while assuming that filtering will occur on the Web application side, or it may include undefined data within the request body. The *category 3* highlights vulnerabilities that may occur during the development and operational phases; this includes exposure to problematic APIs utilized during development and issues related to the security configurations of the Web API server. The *category 4* encompasses vulnerabilities typical of traditional web threats, such as injection attacks and incorrect logging configurations.

This study focuses on authorization-related vulnerabilities as outlined in *API 1:2019* and *API 5:2019*, both of which hold significant importance on this list. Additionally, we investigated the endpoints associated with *API 9:2019*. In this vulnerability, endpoints not defined in the API references are exposed because of a lack of proper authentication and authorization.

### *A. API 1:2019 Broken Object Level Authorization*

*API 1:2019* is an authorization-related vulnerability. Traditional web applications utilize a mechanism called *session* that stores users' authentication status for both web servers and clients. However, a Web API does not typically manage users' authentication status (i.e., stateless), and user management is often achieved by embedding an identifying flag at the endpoint. For example, the endpoint for accessing the information of a user with ID 1 is */user/1*. In such cases, if a Web API server fails to properly authorize access to user information, a malicious user can manipulate the user ID at the endpoint to access other users' information.

### *B. API 5:2019 Broken Function Level Authorization*

*API 5:2019* is also a vulnerability associated with the authorization process. Vulnerabilities occur when proper authorization is not set for each endpoint in the Web API server used by general and privileged users. Malicious users can attack a system by exploiting vulnerabilities in privileged accounts.

### *C. API 9:2019 Improper Assets Management*

*API 9:2019* is associated with improper data handling. Exposed debug endpoints and deprecated API versions can increase potential security risks. For example, if web systems have been updated to use a new API and the old endpoint remains operational, it could provide an accessible point for a malicious user.

## IV. Vulnerability Assessment Tool

Several tools are available for detecting Web API vulnerabilities. However, these vulnerability assessment tools cannot detect the ten vulnerabilities described in the OWASP API Security Top 10. The reason for this failure is that they attempted to detect vulnerabilities by sending predetermined requests and could not specify the authentication information for each target Web API. In addition, various request body data (hereinafter referred to as ”parameters”) used for vulnerability assessment are set by a user who uses these tools, and if the user cannot set appropriate parameters, the assessment becomes difficult. The features of the three representative vulnerability assessment tools and the vulnerability items of the OWASP API TOP 10 that can be assessed are described below.

### *A. Automatic API Attack Tool*

*Automatic API Attack Tool* [3] can flexibly generate vulnerability detection test cases according to the Web APIs to be vulnerability-tested by loading the Web API reference in JSON or YAML formats. For example, for an endpoint that uses a parameter of ”INT type” as ”id,” this tool will change the value of ”id” to a numeric value of another type, such as long or double, and send a vulnerability detection request. This tool can detect vulnerabilities in *API 1:2019*, *API 5:2019*, and *API 9:2019* of the OWASP API Security Top 10. However, Web API references in JSON or YAML formats describing the Web API information are required, and vulnerability assessments may require considerable work and time.

### *B. Vooki, Rest API Scanner*

*Vooki, Rest API Scanner* [4] is a vulnerability assessment tool that sends requests to detect vulnerabilities in *API 3:2019*, *API 7:2019*, and *API 8:2019* of the OWASP API Security Top 10. However, all endpoints, headers, and parameters used in the Web API must be configured by the user of this tool. If the necessary information for vulnerability assessment cannot be set, vulnerabilities may not be detected, even if they exist.

### *C. OWASP ZAP*

The *OWASP Zed Attack Proxy (OWASP ZAP)* [5] is one of the most representative tools used for vulnerability scanning. Primarily, OWASP ZAP is designed to scan vulnerabilities in web systems. However, it can read references written in the OpenAPI format and conduct vulnerability scans of Web APIs by incorporating an add-on or utilizing a command-line tool.

### *D. Problem of Traditional Tools*

One of the major issues with these traditional vulnerability assessment tools is that the endpoints must be explicitly defined by developers. If any endpoint is omitted, this may lead to inaccurate vulnerability assessments. Therefore, developers must perform frequent maintenance and ensure that all endpoints are correctly and comprehensively defined.

## V. Proposed Method

As we mentioned in Section IV, these traditional vulnerability assessment tools require developers to provide endpoint information to the vulnerability tools, which then scan the given endpoint. Since there are many endpoints and parameters in Web APIs, it is difficult for developers to configure them manually. In addition, the endpoints and parameters continue to change because web systems undergo continuous improvements after completion.

In this study, we propose a method for automatically conducting a Web API vulnerability assessment by utilizing references, requests, and responses for Web APIs to detect authorization-related Web API vulnerabilities. First, in the proposed method, if authentication information is required, developers (or vulnerability diagnosticians) set the authentication information to the HTTP request (e.g., HTTP header) according to the API references, and each validation check starts.

The proposed method was executed in three steps.

**Step 1:** Obtaining References

We collected information about Web API references to automatically generate request queries for vulnerability assessment.

**Step 2:** Validation Check of API References

We obtained the response information by constructing a request query using the Web API information obtained from **Step 1**. The constructed request query is sent to the endpoints to validate the Web APIs and the response (i.e., JSON data) is stored.

**Step 3:** Vulnerability Detection

To generate a query for vulnerability assessment, we define the string to be used as a *candidate key* and extract the values of the key, as well as from the JSON data stored in **Step 2**. Vulnerability assessment queries are generated from the *candidate key*, and the vulnerability detection process begins. If the HTTP status code corresponding to each vulnerability assessment query is 200, the proposed method indicates the existence of a vulnerability.

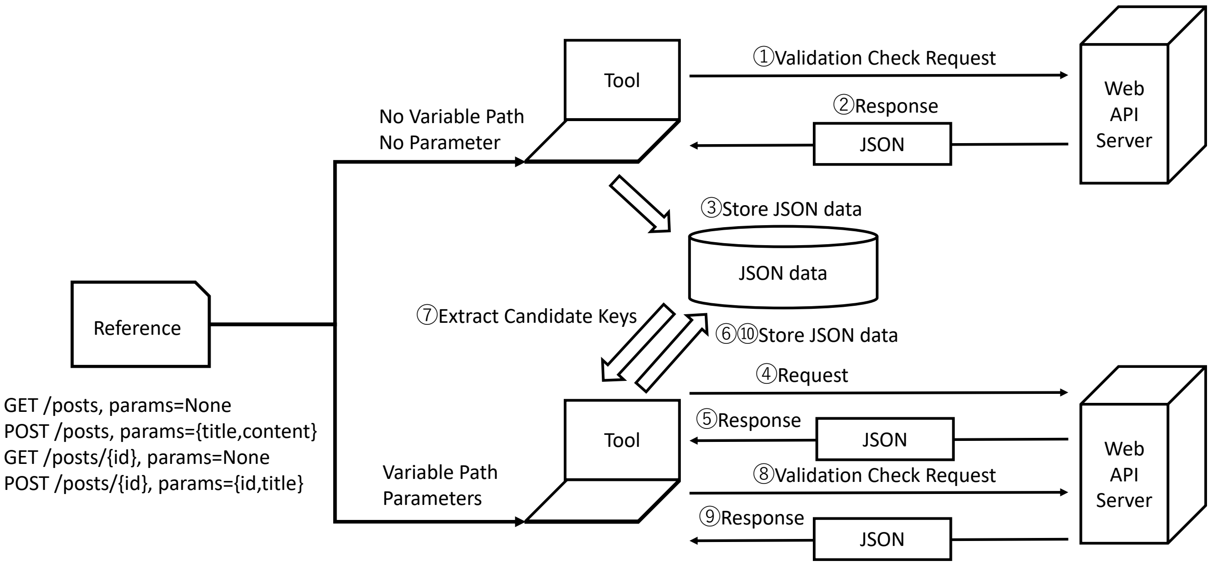


Fig. 1. API Validation Flow

TABLE III
Example of Web API References

| No | HTTP Method | Endpoint | Request Parameters | | Authentication |
|---|---|---|---|---|---|
| | | | Name | Type | |
| 1 | GET | /posts | None | | Non-required |
| 2 | POST | /posts | title<br>content | string<br>string | Non-required |
| 3 | GET | /posts/{id} | None | | Non-required |
| 4 | POST | /posts/{id} | id<br>title | integer<br>string | Non-required |

### *A. Step 1: Obtaining References*

**Step 1** is to collect information about Web API references, such as the HTTP method, endpoints, parameters, and their types using **Method 1** and **Method 2**. Examples of the information about the Web API references are listed in Table III. In Table III, $\{id\}$ is variable which can be replaced. Hereafter, the */posts* is referred to as the *basic path* and $\{id\}$ is the *variable path*. These references were obtained using **Method 1** and **Method 2**.

**Method 1:** This method uses SwaggerHub [6], a resource-rich platform for Web API developers. SwaggerHub provides a wealth of references for Web APIs available in either JSON or YAML formats.

**Method 2:** This method employs the use of official websites. Automatic extraction of information from the HTML source becomes necessary because Web API references are provided in HTML format. In this study, we extracted information regarding Web API references from webpages written in HTML by adapting the methodology employed in a previous study [7].

If a reference does not exist in SwaggerHub (i.e., **Method 1**), the proposed method obtains the reference from the official websites of the Web APIs (i.e., **Method 2**).

### *B. Step 2: Validation Check of API References*

**Step 2** validates the Web APIs (i.e., endpoints and parameters) described in the API references. Request queries for validation are generated based on information about the API references and sent to the endpoints of the Web APIs. An overview of the validation process is presented in Figure 1. Table III shows an example of the information regarding API references.

The validation process for the Web APIs is as follows:

1) The validation check for endpoints without *variable path* and request parameters is first conducted. In Table III, the No.1 API, which has */posts* endpoint, is first selected, and the request query without request parameters is sent to the */posts* endpoint (① in Figure 1). After confirming the validation (i.e., HTTP Status Code is 200), response parameters (i.e., JSON data) that are returned from the endpoint are obtained and stored (② and ③ in Figure 1).
2) The validation check for endpoints including *variable path* without request parameters is conducted. In Table III, the No.3 API, which has */posts/*$\{id\}$ endpoint, is selected. The request query is sent to the endpoint */posts* which is an endpoint one level up from */posts/*$\{id\}$ (④ in Figure 1), Response parameters (i.e., JSON data) that are returned from the endpoint are obtained and stored (⑤ and ⑥ in Figure 1).
   Next, the proposed method searches for a parameter with the key *id* in the response parameters. When the keys *id* are found in the response parameters (⑦ in Figure 1), the *variable path* $\{id\}$ is replaced with the corresponding value for the first key *id*.
   For example, if the response parameters displayed in Figure 2 are obtained, the endpoint of the next request query is */posts/38*, and this query is sent to the */posts/38* endpoint (⑧ in Figure 1). If there is no endpoint one level up or the same key, the *variable path* $\{id\}$ is

```
[
    {”id”: 38, ”title”: ”ArticleName1”},
    {”id”: 1, ”title”: ”ArticleName2”}
]
```

Fig. 2. Example of response parameters from /posts endpoint in No. 3 API

```
{
    {”title”: ”ArticleName1”, ”content”: ”Content1”},
    {”title”: ”ArticleName2”, ”content”: ”Content2”}
}
```

Fig. 3. Example of response parameters from /posts endpoint in No. 2 API

replaced with the fixed value *1* (i.e., default value).
After confirming the validation (i.e., HTTP Status Code is 200), response parameters (i.e., JSON data) that are returned from the endpoint are obtained and stored (⑨ and ⑩ in Figure 1).

3) The validation check for endpoints, including request parameters without *variable path*, is conducted. In Table III, the No.2 API, which has */posts* endpoint and the request parameter with the keys *title* and *content* is selected, and the request query without request parameters is sent to the */posts* endpoint using HTTP GET Method (④ in Figure 1). Response parameters (i.e., JSON data) that are returned from the endpoint are obtained and stored (⑤ and ⑥ in Figure 1).
When the key *title* and *content* are found in the response parameters (⑦ in Figure 1), the keys *title* and *content* of the request parameter are set to the corresponding value for the keys *title* and *content*.
For example, if the response parameters displayed in Figure 3 are obtained, the endpoint and parameter are */posts* and {*”title”: ”ArticleName1”, ”Content1”: ‘”content1”*} respectively (⑦ in Figure 1). This query is sent to the */posts* endpoint (⑧ in Figure 1). If there is not the same key of the request parameters, the keys *title* and *content* of the request parameter are set to the fixed value *”a”* (i.e., default value).
After confirming the validation (i.e., HTTP Status Code is 200), response parameters (i.e., JSON data) that are returned from the endpoint are obtained and stored (⑨ and ⑩ in Figure 1).

4) The validation check for endpoints including *variable path* and request parameters is conducted. In Table III, the No.4 API, which has */posts/*{*id*} endpoint and the request parameter with the keys *id* and *title*, is selected. Similar to the above *process 2)*, when the keys *id* is found in the response parameters, the *variable path* {*id*} is replaced with the corresponding value for the first key *id* (④ – ⑦ in Figure 1). Additionally, similar to the above *process 3)*, when the key *title* is found in the reponse parameters, the key *title* of the request parameter is set to the corresponding value for the key *title* (④ – ⑦ in Figure 1).
For example, if the response parameters displayed in Figure 2 are obtained, the endpoint and parameter are */posts/38* and {*”id”: 38, ”title”: ”ArticleName1”*} respectively (⑦ in Figure 1). This query is sent to the */posts/38* endpoint (⑧ in Figure 1). If there is no endpoint one level up or the same key in the request parameters, the *variable path* {*id*} is replaced with the fixed value *1* (i.e., */posts/1*), and the keys *id* and *title* of the request parameter are set to the fixed values *1* and *”a”* (i.e., {*”id”: 1, ”title”: ”a”*}). After confirming the validation (i.e., HTTP Status Code is 200), response parameters (i.e., JSON data) that are returned from the endpoint are obtained and stored (⑨ and ⑩ in Figure 1).

The above four processes (*process 1) - process 4)*) are conducted in order from the *process 1)*.

After completing the validation check for all endpoints, the vulnerability detection process detailed in **Step 3** begins.

### *C. Step 3: Vulnerability Detection*

The objective of **Step 3** is to detect authorization-related Web API vulnerabilities. This step generates two types of vulnerability assessment queries.

- The first query aims to detect the existence of endpoints unintended for exposure by the developers. This query is basically for API 1:2019 in the OWASP API Security Top 10.
- The second query aims to detect vulnerabilities within known endpoints by replacing the *variable path* and changing parameters using the JSON data stored in **Step 2**. This query is basically for API 5:2019 in the OWASP API Security Top 10.

An overview of the vulnerability assessment query generation process is shown in Figure 4. To generate vulnerability assessment queries, we must first extract strings to replace the *variable path* and change the parameters from the JSON data stored in **Step 2**.

*Candidate keys* comprise the following two types of words:

- Words that are used as the key names and values in the JSON data.
- Words that are included in the endpoints provided by the API references.

When strings that are used as the key names and values in the JSON data are provided as text, we separate the texts into words using morphological analysis. For example, the words ”id,” ”1,” ”title,” and ”ArticleA” are extracted from the JSON data in ④ of Figure 4.

*1) Queries for Detection of Endpoints Including Unintended Exposure:* To detect endpoints, including unintended exposure, we need to estimate unintended endpoints for developers. The *candidate keys* are further classified into two types based on morpheme analysis. One is all noun words except numerals, and the other is numerals (e.g., 0, 1, ....). For example, in ⑤ in Figure 4, the noun words ”id,” ”title,” and ”Article” are extracted from the words ”id,” ”1,” ”title,”

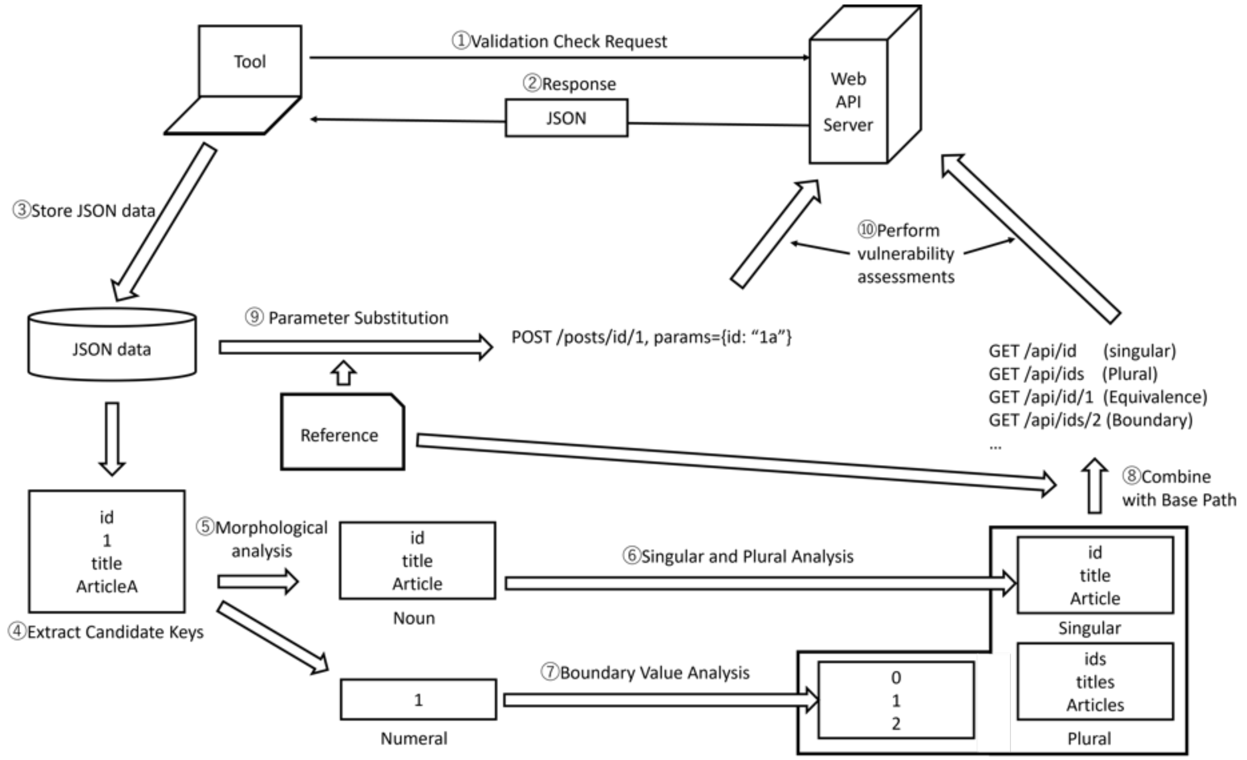


Fig. 4. Vulnerability Assessment Query Generation Flow

and "ArticleA." Additionally, the numeral *1* is extracted from the words "id," "1," "title," and "ArticleA."

For nouns, the proposed method adapts both singular and plural forms of words to detect endpoints, including unintended exposures. If only a plural form of the word is extracted, the proposed method transforms it into a singular form, and vice versa. The words "id," "ids," "title," "titles," "Article," and "Articles" are transformed from the words "id," "1," "title," and "Article" in ⑥ in Figure 4.

In the case of a numeral, given a set of numerals, the maximum value, the maximum value ± 1, the minimum value, and the minimum value ± 1 are adapted according to the boundary value analysis to detect endpoints, including unintended exposure. For example, numerals *0*, *1*, and *2* are generated from numeral *1* in ⑦ of Figure 4.

Finally, by the above-mentioned processes, we can obtain the numerals *0*, *1* and *2* and the noun words "id," "ids," "title," "titles," "Article," and "Articles" are generated in ⑥ and ⑦ of Figure 4.

The vulnerability assessment queries were constructed from these numerals and nouns. In the proposed method, the four combinations of vulnerability assessment queries (⑧ in Figure 4) are as follows:

- <API entry point>/<plural noun word>
- <API entry point>/<singular noun word>
- <API entry point>/<plural noun word>/<number>
- <API entry point>/<singular noun word>/<number>

The *API entry point* represents the base path obtained from API references. For example, in Table IV, */api* is defined as *API entry point*. The *API entry point* is the */api*, which is one level away from the */api/*{*id*} if the *variable path* is included.

TABLE IV
Example of Web API References (Non-parameter)

| No | HTTP Method | Endpoint | Request Parameters | | Authentication |
|---|---|---|---|---|---|
| | | | Name | Type | |
| 1 | GET | /api | None | | Non-required |
| 2 | GET | /api/{id} | None | | Non-required |

Finally, if the numerals *1* and *2*, and the noun words "id" and "ids" are given, the endpoints of the vulnerability assessment queries are generated as follows:

- /api/id
- /api/ids
- /api/id/1
- /api/ids/1
- /api/id/2
- /api/ids/2

The vulnerability assessment queries which has these endpoints are illustrated in No.1 – No.6 of Table VI.

*2) Queries for Detection of Vulnerabilities within Known Endpoints Using Replacement and Changing:* To detect vulnerabilities in terms of parameters within known endpoints, we must estimate the resources that are not managed at the configuration or code levels. The proposed method first uses the values for the keys of the request parameters using JSON data, which are stored in **Step 2**, and the API references. Next, the proposed method changes the type of parameters and generates new values for the keys of the request parameters for vulnerability assessment using the JSON data and API

TABLE V
Example of Web API References (Parameters)

| No | HTTP Method | Endpoint | Request Parameters |  | Authentication |
|---|---|---|---|---|---|
|  |  |  | Name | Type |  |
| 1 | POST | /posts | id | integer | Non-required |
| 2 | POST | /posts | status | string | Non-required |

```
{{”id”: 5}}
{{”status”: ”publish”}}
```

Fig. 5. Example of JSON data

references (⑨ of Figure 4).

We explain the generation of parameters for vulnerabilities using the API references listed in Table V and the JSON data illustrated in Figure 5.

If a key has a numeric type in the API references, three values for the key are generated: first, the same value as that of the numeric type is set; second, the fixed value ”a” as that of the string type is set; third, the proposed method changes the numeric type to the string type keeping the numeral, and concatenates the numeral of the string type and the fixed value ”a” of the string type. For example, if the key ”id” is 5 (i.e., {”id”:5}), the proposed method first uses {”id”:5}. Second, the proposed method generates {”id”:”a”} where ”a” is used as the fixed value of string type in this study. Third, the proposed method generates {”id”:”5a”} after concatenating ”5” and ”a” of the string type.

Finally, the request parameters {”id”:5}, {”id”:”a”} and {”id”:”5a”} are generated from {”id”:5}. The vulnerability assessment queries which has these parameters are illustrated in No.7 – No.9 of Table VI.

If a key has a string type in the API references, the following three values for the key are generated. First, the same value of string type is set. Second, when the value of the string type is a numeral, the numeral is converted to a string type; otherwise, the value ”1” is used as the fixed value of the string type in this study. Third, the proposed method concatenates the original value of the string type and the fixed value ”1” of the string type. For example, if the key ”status” is ”publish” (i.e., {”status”:”publish”}), the proposed method first uses {”status”: ”publish”}. Second, the proposal method replaces ”publish” with the fixed value ”1” (i.e., {”status”:”1”}). Third, the proposal method replaces ”publish” with ”publish1” (i.e., {”status”:”publish1”}).

Finally, the request parameters {”status”:”publish”}, {”status”:”1”} and {”status”:”publish1”} are generated from {”status”:”publish”}. The vulnerability assessment queries which has these parameters are illustrated in No.10 – No.12 of Table VI.

*3) Vulnerability Detection:* The above-mentioned vulnerability assessment queries (Table VI) were generated, and the vulnerability detection process was initiated.

The HTTP status code of the response is used to determine the vulnerability assessment results. If the HTTP status code corresponding to each vulnerability assessment query is 200, the proposed method indicates the existence of a vulnerability. Additionally, if the HTTP status code is 501, the proposed method indicates the existence of a vulnerability because the API server attempts to perform certain processes. When the HTTP status codes are not equal to 200 and 501, the proposed method indicates the absence of vulnerabilities.

TABLE VI
Example of the vulnerability assessment queries

| No | HTTP Method | Endpoint | Request Parameters |
|---|---|---|---|
| 1 | GET | /api/id | None |
| 2 | GET | /api/ids | None |
| 3 | GET | /api/id/1 | None |
| 4 | GET | /api/ids/1 | None |
| 5 | GET | /api/id/2 | None |
| 6 | GET | /api/ids/2 | None |
| 7 | POST | /posts | {”id” : 5} |
| 8 | POST | /posts | {”id” : ”a”} |
| 9 | POST | /posts | {”id” : ”5a”} |
| 10 | POST | /posts | {”status” : ”publish”} |
| 11 | POST | /posts | {”status” : ”1”} |
| 12 | POST | /posts | {”status” : ”publish1”} |

## VI. Experiment Environment

We conducted a preliminary experiment using actual environment to evaluate whether the proposed method can detect the OWASP API Security Top 10 vulnerabilities. We used the well-known CMSs as the experimental environments. In our selection criteria, we focus on whether the CMSs include known vulnerabilities and whether they have higher market shares [8]. Consequently, we selected three well-known CMSs and a vulnerable training environment.

- vAPI [9]
- WordPress [10]
- Ghost CMS [11]
- Joomla [12]

### A. vAPI

vAPI was developed to test the vulnerabilities in the OWASP API Security Top 10. It replicates the vulnerabilities described in the OWASP API Security Top 10, serving as an application that enables users to experience attacks.

### B. WordPress

WordPress is a leading CMS with a market share of 63.7% as of January 2023, according to a report by W3Techs [13]. It is utilized by global organizations, such as Microsoft, Mozilla, and Apache. However, it has been targeted by many cyber attackers owing to its widespread use. WordPress versions 4.7.0 and 4.7.1 contain a vulnerability that allows article tampering by bypassing authentication [14]. This risk was categorized as *API 1:2019* in the OWASP API Security Top 10.

### *C. Ghost CMS*

Ghost CMS has a relatively modest market share of 0.08% compared with the other CMSs. However, it has steadily gained traction as an emerging CMS [15]. Additionally, it has been adopted by well-known websites such as Cloudflare and Duolingo. Ghost CMS versions from 4.0.0 to 4.9.4 contain vulnerabilities that allow an undisclosed endpoint to access information that typically requires administrative privileges [16]. This vulnerability was categorized as *API 5:2019* in the OWASP API Security Top 10.

### *D. Joomla*

Joomla has a small market share of 2.7% as of January 2023, according to the report of W3Techs [17]. Despite its small market share, it is renowned for its strong emphasis on security and is frequently used on government websites.

## VII. Experiment Results

We conducted a preliminary experiment using well-known CMSs, such as WordPress, Ghost CMS, and Joomla. We also used a deliberately vulnerable testing environment called vAPI. For vAPI, the proposed method could detect vulnerabilities (*API 1:2019* and *API 5:2019* of the OWASP API Security Top 10) because vAPI was developed to test vulnerabilities in the OWASP API Security Top 10. WordPress has a known vulnerability (*API 1:2019* of the OWASP API Security Top 10), and Ghost CMS has a known vulnerability (*API 5:2019*). Therefore, in this study, we first evaluate whether these known vulnerabilities can be detected. Additionally, the proposed method attempts to detect vulnerabilities that are not known (i.e., vulnerabilities where CVE ID are not assigned). CVE (Common Vulnerabilities and Exposures) is a list of publicly disclosed vulnerabilities and each vulnerability is assigned a CVE ID number. Joomla does not have a known vulnerability described in the OWASP API Security Top 10, and the proposed method attempts to detect vulnerabilities that are not known (i.e., vulnerabilities where CVE ID are not assigned).

TABLE VII
Web API References (*API 1:2019* of vAPI)

| No | HTTP Method | Endpoint | Request Parameters Name | Request Parameters Type | Authentication |
|---|---|---|---|---|---|
| 1 | GET | /vapi/api1/user/{api1_id} | None | | Required |
| 2 | POST | /vapi/api1/user | username<br>name<br>course<br>password | string<br>string<br>string<br>string | Required |
| 3 | PUT | /vapi/api1/user/{api1_id} | username<br>name<br>course<br>password | string<br>string<br>string<br>string | Required |

TABLE VIII
Number of Candidate Keys for *API 1:2019* of vAPI

| Item | Value |
|---|---|
| The Number of String Candidate Keys | 18 |
| The Number of Numerical Candidate Keys | 5 |

TABLE IX
Part of Candidate Keys for *API 1:2019* of vAPI

| String Candidate Key | Numerical Candidate Key |
|---|---|
| site | 0 |
| id | 1 |
| entry | 9 |
| user | 237235 |
| error | 2324 |
| username | |

```
GET http://[Server IP Address]/vapi/api1/user/1
Content−Type: application/json
Authorization−Token: (** mask **)
```

Fig. 6. Request for *API 1:2019* of vAPI

```
{
  ”id”: 1,
  ”username”: ”michaels”,
  ”name”: ”Michael Scott”,
  ”course”: ”flag{api1_(** mask **)}”
}
```

Fig. 7. Response for *API 1:2019* of vAPI

### *A. Vulnerability Detection for vAPI*

First, we evaluated whether the proposed method could detect *API 1:2019* and *API 5:2019* from the OWASP API Security Top 10. Given that vAPI provides a webpage with API references for each endpoint, we used this information to create endpoint-specific settings.

*1) Detection for API 1:2019: API 1:2019* of the vAPI contains a vulnerability in the authorization process. This vulnerability allows an authenticated user to access information regarding other users. First, in the proposed method, developers (or vulnerability diagnosticians) set the authentication information to the HTTP request (e.g., HTTP header) according to the API references, and **Step 1** begins. In **Step 1**, we can obtain the API references listed in Table VII. The number of APIs in *API 1:2019* of the vAPI was 3. In **Step 2**, the No.1 API which has the */vapi/api1/user/*{*api1_id*} endpoint, is first selected. {*api1_id*} is not included in the response parameter from */vapi/api1/user*, which is one level up from */vapi/api1/user/*{*api1_id*}. Therefore, the *variable path* {*api1_id*} is replaced by the default value of *1*, and a request with authorization information is sent to the */vapi/api1/user/1* endpoint. Similarly, the No.2 API and 3 API are selected, and a request with authorization information was sent to the endpoints. After completing **Step 2**, the number of *candidate keys* is shown in Table VIII is obtained from the JSON data. Table IX presents part of the *candidate keys*. In **Step 3**, the request with the authorization information shown in Figure 6 is sent to the */vapi/api1/user/1* endpoint, which is constructed using *the candidate keys* of the *user* and *1*, and the response shown in Figure 7 is obtained. Because the vAPI issues a flag when an attack is successful and an authenticated user can access information about another user (i.e., id = 1), the proposed method indicates the existence of the *API 1:2019* vulnerability. In this step, 2714 vulnerability assessment queries were sent to the vAPI server.

```
GET /vapi/api5/users
Content−Type: application/json
Authorization−Token: (∗∗ mask ∗∗)
```

Fig. 8. Request for *API 5:2019* of vAPI

```
{
    ”id”: 1,
    ”username”: ”admin”,
    ”name”: ”Admin User”,
    ”address”: ”flag{api5_(∗∗ mask ∗∗)}”,
    ”mobileno”: ”8080808080”
}
```

Fig. 9. Response for *API 5:2019* of vAPI

TABLE X
Web API References (*API 5:2019* of vAPI)

| No | HTTP Method | Endpoint | Request Parameters | | Authentication |
|---|---|---|---|---|---|
| | | | Name | Type | |
| 1 | GET | /vapi/api5/user/{api5_id} | None | | Required |
| 2 | POST | /vapi/api5/user | username<br>password<br>name<br>address<br>mobileno | string<br>string<br>string<br>string<br>string | Required |

TABLE XI
Number of Candidate Keys for *API 5:2019* of vAPI

| Item | Value |
|---|---|
| The Number of String Candidate Keys | 6 |
| The Number of Numerical Candidate Keys | 0 |

TABLE XII
Part of Candidate Keys for *API 5:2019* of vAPI

| String Candidate Key |
|---|
| u |
| false |
| s |
| cause |
| user |

*2) Detection for API 5:2019: API 5:2019* implementation of the vAPI also exposes a vulnerability in the authorization process. This vulnerability permits general users to access confidential endpoints at which administrator privileges are required. If this endpoint is accurately deduced, a list of users with administrative privileges can be procured. First, in the proposed method, developers (or vulnerability diagnosticians) set the authentication information to the HTTP request (e.g., HTTP header) according to the API references, and **Step 1** begins. In **Step 1**, we can obtain the API references listed in Table X. The number of APIs in *API 5:2019* of the vAPI was 2. In **Step 2**, the No.1 API which has the */vapi/api5/user/*{*api5_id*} endpoint, is first selected. {*api5_id*} is not included in the response parameter from */vapi/api5/user*, which is one level up from */vapi/api5/user/*{*api5_id*}. Therefore, the *variable path* {*api5_id*} is replaced by the default value of *1*, and a request with authorization information is sent to the */vapi/api5/user/1* endpoint. Similarly, the No.2 API are selected, and a request with authorization information was sent to the endpoints. After completing **Step 2**, the number of *candidate keys* shown in Table XI is obtained from the JSON data. Table XII presents part of the *candidate keys*. In **Step 3**, the request with the authorization information shown in Figure 8 is sent to the */vapi/api5/users* endpoint, which is constructed by *the candidate keys* of *users*, which is a plural of the singular form *user*, as shown in Figure 8. Because the vAPI issues a flag when an attack is successful, and an general user can access the confidential endpoint for which administrator privileges are needed, the proposed method indicates the existence of the *API 5:2019* vulnerability. In this step, 15586 vulnerability assessment queries were sent to the vAPI server. The number of queries was relatively large because the number of generated queries rapidly increased as the number of parameters increased.

### *B. Vulnerability Detection for WordPress*

The 4.7.0 version of WordPress has the privilege vulnerability identified as CVE-2017-1001000. This vulnerability prevails in the authorization process of the */posts/*{*id*} endpoint. If a request parameter including an invalid content identifier (i.e., a string type value composed of a numeral and string) is sent to the */posts/*{*id*} endpoint, it becomes possible to overwrite the content. This vulnerability was categorized as *API 1:2019* and *API 2:2019* in the OWAPS API Security Top 10. We evaluated whether this vulnerability and other unknown vulnerabilities could be detected by implementing the proposed method.

First, in the proposed method, developers (or vulnerability diagnosticians) set the authentication information to the HTTP request (e.g., HTTP header) according to the API references, and **Step 1** begins. In **Step 1**, we obtain the API references listed in Table XIII. The number of APIs that required authorization was 47. Part of these APIs are shown in Table XIII. In **Step 2**, the No.1 API which has the */wp-json/wp/v2/posts/*{*post_id*} endpoint, is selected. {*post_id*} is not included in the response parameter from */wp-json/wp/v2/posts*, which is one level up from */wp-json/wp/v2/posts/*{*post_id*}. Therefore, the *variable path* {*post_id*} is replaced by the default value of *1*, and a request with authorization information is sent to the */wp-json/wp/v2/posts/1* endpoint. Similarly, other APIs are selected, and a request with authorization information is sent to the endpoints. After completing **Step 2**, the number of *candidate keys* shown in Table XIV is obtained from the JSON data. Table XV presents an example of the *candidate keys*. In **Step 3**, the request with authorization information shown in Figure 10 with the parameter {*”id”:”1a”,”title”:”a”*} is sent to the */wp-json/wp/v2/posts/1* endpoint and the response shown in Figure 11 is obtained. The method for generating the parameter {*”id”:”1a”,”title”:”a”*} is described in Section V-C2. As illustrated in Figure 11, the value of the ”title” are tampered to ”a.” This result is considered to be the vulnerability identified as CVE-2017-1001000. Additionally, other unknown vulnerabilities were not detected using the

TABLE XIII
Part of Web API References (WordPress)

| No | HTTP Method | Endpoint | Request Parameters | | Authentication |
|---|---|---|---|---|---|
| | | | Name | Type | |
| 1 | GET | /wp-json/wp/v2/posts/{post_id} | None | | Required |
| 2 | POST | /wp-json/wp/v2/posts | id<br>title<br>content<br>status | integer<br>string<br>string<br>string | Required |
| 3 | GET | /wp-json/wp/v2/posts/{parent}/revisions/{id} | None | | Required |
| .. | .. | .. | .. | | .. |

TABLE XIV
Number of Candidate Keys for WordPress

| Item | Value |
|---|---|
| The Number of String Candidate Keys | 88 |
| The Number of Numerical Candidate Keys | 29 |

TABLE XV
Part of Candidate Keys for WordPress

| String Candidate Key | Numerical Candidate Key |
|---|---|
| site | 0 |
| id | 1 |
| http | 27079 |
| message | 27080 |
| world | 27081 |
| setting | |

```
POST /wp−json/wp/v2/posts/1
Content−Type: application/json

{”id”: ”1a”, ”title”: ”a”}
```

Fig. 10. Request for WordPress

```
{
    ”id”: 1,
    ”date”: ”2023−07−07T20:11:50”,
    ”modified”: ”2023−07−18T22:27:16”,
    ”slug”: ”hello−world”,
    ”type”: ”post”,
    ”title”: {
        ”raw”: ”a”,
        ”rendered”: ”a”
    },
}
```

Fig. 11. Response for WordPress

proposed method. In this step, 5288 vulnerability assessment queries were sent to the WordPress server.

### C. Vulnerability Detection for Ghost CMS

Ghost CMS versions from 4.0.0 to 4.9.4 contain a vulnerability pertinent to inadequate authorization management. This susceptibility permits the extraction of an administrator API key, including a response to a specific API, thereby enabling privilege escalation for any authenticated user. This vulnerability was categorized under *API 5:2019* of the OWAPS API Security Top 10.

First, in the proposed method, developers (or vulnerability diagnosticians) set the authentication information to the HTTP request (e.g., HTTP header) according to the API references, and **Step 1** begins. In **Step 1**, we can obtain the API references listed in Table XVI. The number of APIs that required authorization was 29. Part of these APIs are shown in Table XIII. In this experiment, we used version 2 of the API of Ghost CMS, and the *variable path* {*version*} was replaced with *v2*. In **Step 2**, the No.1 API, which contains the */ghost/api/v2/admin/posts* endpoint, is selected, and a request with authorization information is sent to the */ghost/api/v2/admin/posts* endpoint. Similarly, other APIs are selected, and a request with authorization information is sent to the endpoints. After completing **Step 2**, the number of *candidate keys* shown in Table XVII is obtained from the JSON data. Table XVIII presents part of the *candidate keys*. In **Step 3**, a request with authorization information, as shown in Figure 12 is sent to the */ghost/api/v2/admin/users* endpoint, and the response shown in Figure 13 is obtained. Consequently, 11 endpoints were identified as vulnerabilities. The 11 endpoints are listed in Table XIX. The list of users was procured using the administrator endpoint. Among the 11 endpoints, the */admin/site/* endpoint was defined as accessible by general users in the API references. Therefore, 10 endpoints were identified as vulnerabilities. In particular, the */admin/session/* and */admin/integrations/* endpoints were not described in the API references. The result was considered a vulnerability, identified as CVE-2021-39192. In this step, 2108 vulnerability assessment queries were sent to the Ghost CMS server.

### D. Vulnerability Detection for Joomla

In the Joomla Web API, no obvious vulnerabilities associated with the OWASP API Security Top 10 were reported. However, the proposed method attempted to detect vulnerabilities that are not known (i.e., vulnerabilities where CVE ID are not assigned).

First, in the proposed method, developers (or vulnerability diagnosticians) set the authentication information to the HTTP request (e.g., HTTP header) according to the API references, and **Step 1** begins. Similar to the detections for vAPI, WordPress, and Ghost CMS, **Steps 1, 2, and 3** were processed. After completing **Step 2**, the number of *candidate keys* shown in Table XX were obtained from the JSON data. Table XXI presents part of the *candidate keys*. In **Step 3**, a request with the authorization information shown in Figure 14 is sent to the */api/index.php/v1/extensions*

TABLE XVI
Part of Web API References (Ghost CMS)

| No | HTTP Method | Endpoint | Request Parameters<br>Name | <br>Type | Authentication |
|---|---|---|---|---|---|
| 1 | GET | /ghost/api/{version}/admin/posts | None | | Required |
| 2 | GET | /ghost/api/{version}/admin/posts/{id} | None | | Required |
| 3 | POST | /ghost/api/{version}/admin/posts | title | string | Required |
| 4 | GET | /ghost/api/{version}/admin/settings | None | | Required |
| 5 | GET | /ghost/api/{version}/admin/posts/{parent}/revisions/{id} | None | | Required |

```
GET /ghost/api/v2/admin/users/
Content−Type: application/json
Authorization: Ghost (∗∗ mask ∗∗)
```

Fig. 12. Request for Ghost CMS

```
{
  ”users”: [ {
        ”id”: ”1”,
        ”name”: ”Yuki Ishida”,
        ”slug”: ”yuki”,
        ”email”: ”(∗∗ mask ∗∗)”,
        ”profile_image”: null,
        ”cover_image”: null,
        ”bio”: null,
        ”website”: null,
        ”location”: null,
        ”facebook”: null,
        ”twitter”: null,
        ”accessibility”: null,
        ”status”: ”active”,
        ”meta_title”: null,
        ”meta_description”: null,
        ”tour”: null,
        ”last_seen”: ”2023−07−22T10:53:11.000Z”,
        ”created_at”: ”2023−05−06T08:14:32.000Z”,
        ”updated_at”: ”2023−07−22T10:53:11.000Z”,
        ”url”: ”http://[Server IP Address]/404/”
  }, ... ],
}
```

Fig. 13. Response for Ghost CMS

```
POST /api/index.php/v1/extensions
Content−Type: application/json
Authorization: Bearer (∗∗ mask ∗∗)
```

Fig. 14. Request for Joomla

endpoint, and the response shown in Figure 15 was obtained.

Consequently, 3 unpublished endpoints were identified as vulnerabilities because the endpoints were not described in the API references. The 3 unpublished endpoints are listed in Table XXII. This vulnerability was categorized under *API 9:2019* of the OWAPS API Security Top 10. In particular, the */api/index.php/v1/extensions* endpoint illustrated in Figure 15 shows a catalog of extended functions installed within Joomla. This vulnerability implies the potential risk of exploiting the system configuration information if privilege escalation is perpetrated owing to other vulnerabilities.

TABLE XVII
Number of Candidate Keys for Ghost CMS

| Item | Value |
|---|---|
| The Number of String Candidate Keys | 974 |
| The Number of Numerical Candidate Keys | 887 |

TABLE XVIII
Part of Candidate Keys for Ghost CMS

| String Candidate Key | Numerical Candidate Key |
|---|---|
| sell | 0 |
| pages | 1 |
| tags | 40633578 |
| upload | 40633579 |
| site | 40633580 |
| session | |

TABLE XIX
Vulnerability Assessment Results for Ghost CMS

| Endpoint | HTTP StatusCode |
|---|---|
| /admin/pages/ | 200 |
| /admin/tags/ | 200 |
| /admin/posts/ | 200 |
| /admin/roles/ | 200 |
| /admin/users/ | 200 |
| /admin/settings/ | 200 |
| /admin/site/ | 200 |
| /admin/users/1/ | 200 |
| /admin/themes/ | 200 |
| /admin/session/ | 200 |
| /admin/integrations/ | 200 |

### *E. System Impact of Vulnerability Assessment using the Proposed Method*

We evaluated the impact of each CMS in terms of system load. Table XXIII lists the number of endpoints recorded in the API reference for each CMS, the number of *candidate keys* extracted from JSON data, and the number of vulnerability assessment queries.

Compared with the number of queries between WordPress and Ghost CMS, although the number of endpoints in WordPress is greater than that in Ghost CMS, the number of *candidate keys* in WordPress is lower than that in Ghost CMS. One reason for this is that there are more types of strings in keys and values in WordPress is more than that in Ghost CMS. However, although the number of *candidate keys* in WordPress was less than that in Ghost CMS, the number of vulnerability queries in WordPress was greater than that in Ghost CMS. One of the reasons for this is that the number of keys and values in WordPress is greater than that in Ghost CMS. Specifically, although the number of endpoints and *candidate keys* in the

```
{
   ”links”: {
      ”self”: ”http://[Server IP Address]/api/index.php/v1/
            extensions”,
      ”next”: ”http://[Server IP Address]/api/index.php/v1/
            extensions?page%5Boffset%5D=20&page%5
            Blimit%5D=20”,
      ”last”: ”http://[Server IP Address]/api/index.php/v1/
            extensions?page%5Boffset%5D=220&page%5
            Blimit%5D=20”
   },
   ”data”: [{
         ”type”: ”manage”,
         ”id”: ”91”,
         ”attributes”: {
            ”name”: ”Authentication − Joomla”,
            ”type”: ”plugin”,
            ”folder”: ”authentication”,
            ”client_id”: 0,
            ”status”: 2,
            ”version”: ”3.0.0”,
            ”id”: 91
         }
   }, ... ]
}
```

Fig. 15. Response for Joomla

TABLE XX
Number of Candidate Keys for Joomla

| **Item** | **Value** |
|---|---|
| The Number of String Candidate Keys | 134 |
| The Number of Numerical Candidate Keys | 57 |

TABLE XXI
Part of Candidate Keys for Joomla

| **String Candidate Key** | **Numerical Candidate Key** |
|---|---|
| site | 0 |
| featured | 1 |
| next | 25079 |
| page | 25080 |
| privacy | 25081 |
| category | |

TABLE XXII
Vulnerability Assessment Results for Joomla

| Endpoint | HTTP StatusCode |
|---|---|
| /api/index.php/v1/redirects | 200 |
| /api/index.php/v1/contacts | 200 |
| /api/index.php/v1/extensions | 200 |

vAPI(API 5) was the lowest, the number of vulnerability queries was the highest. Although this number is considered tolerable during the development phase, there will be a need to devise an approach that does not generate obvious invalid vulnerability requests, such as reductions in *candidate keys*.

## VIII. Conclusion

In this study, we proposed a vulnerability assessment method for addressing *API 1:2019*, *API 5:2019*, and *API 9:2019* vulnerabilities in the OWASP API Security Top 10. The proposed method conducts a dynamic vulnerability assessment using requests to validate the Web APIs according to API references and vulnerability requests that take advantage of their corresponding responses.

TABLE XXIII
Vulnerability Assessment Results in terms of System Load

| | vAPI1 | vAPI5 | WordPress |
|---|---|---|---|
| The number of endpoints | 3 | 2 | 47 |
| The number of candidate keys | 23 | 6 | 116 |
| The number of queries | 2714 | 15586 | 5288 |
| The number of http status 200 | 5 | 15554 | 184 |
| The number of http status 500 | 2704 | 22 | 0 |
| The number of invalid requests | 5 | 10 | 5104 |

| | Ghost | Joomla |
|---|---|---|
| The number of endpoints | 29 | 47 |
| The number of candidate keys | 1861 | 191 |
| The number of queries | 2108 | 657 |
| The number of http status 200 | 11 | 12 |
| The number of http status 500 | 0 | 46 |
| The number of invalid requests | 2097 | 599 |

In this experiment, we confirmed the following:

- *API 1:2019* and *API 5:2019* vulnerabilities in the OWASP API Security Top 10 can be exactly detected using the vAPI.
- In WordPress, the known vulnerabilities of *API 1:2019* and *API 5:2019* (i.e., CVE-2017-1001000) can be exactly detected.
- In Ghost CMS, the known vulnerabilities of *API 5:2019* (i.e., CVE-2021-39192) can be exactly detected. Additionally, 10 endpoints are identified as vulnerabilities. Specifically, the */admin/session/* and */admin/integrations/* endpoints are not described in the API reference.
- In Joomla, the unknown vulnerabilities of *API 9:2019* can be detected. Additionally, 3 endpoints are identified as unpublished endpoints.

In addition, we evaluated the impact on each CMS in terms of system load. The number of assessment queries of the proposed method depends on the type of strings in the keys and values and the number of keys. Therefore, there is a need to develop an approach that does not generate obvious invalid vulnerability requests, such as a reduction in the number of *candidate keys*.

With the application of the proposed method, we anticipate an effective vulnerability assessment of Web APIs during the developmental phase and prior to production.

## References

[1] Akamai, “State of the internet / report — api: The attack surface that connects us all — akamai,” https://www.akamai.com/site/en/documents/state-of-the-internet/soti-security-api-the-attack-surface-that-connects-us-all.pdf, (Accessed on 07/10/2023).

[2] Open Worldwide Application Security Project, “Owasp top 10 api security risks – 2019 - owasp api security top 10,” https://owasp.org/API-Security/editions/2019/en/0x11-t10/, (Accessed on 07/10/2023).

[3] imperva, “imperva/automatic-api-attack-tool: Imperva’s customizable api attack tool takes an api specification as an input, generates and runs attacks that are based on it as an output.” https://github.com/imperva/automatic-api-attack-tool, (Accessed on 07/13/2023).

[4] VegaBird Technologies, “Vooki - web application and api vulnerability scanner — vooki infosec,” https://www.vegabird.com/vooki/, (Accessed on 07/13/2023).

[5] Open Worldwide Application Security Project, "OWASP ZAP," https://www.zaproxy.org/, (Accessed on 07/25/2023).

[6] SwaggerHUB, "SwaggerHUB," https://app.swaggerhub.com/search, (Accessed on 07/25/2023).

[7] Takai Masanari and Sakaguchi Tetsuo, "Automatic generation of program libraries for accessing web apis,," *IPSJ Information Fundamentals and Access Technologies (IFAT)*, vol. 2012-IFAT-108, no. 1, pp. 1–8, 09 2012, (in Japanese).

[8] Q-Success: World Wide Web Technology Surveys, "Market share yearly trends for content management systems, july 2023," https://w3techs.com/technologies/overview/content_management, (Accessed on 07/15/2023).

[9] roottusk, "roottusk/vapi: vapi is vulnerable adversely programmed interface which is self-hostable api that mimics owasp api top 10 scenarios through exercises." https://github.com/roottusk/vapi, (Accessed on 07/16/2023).

[10] WordPress Foundation, "Blog tool, publishing platform, and cms – wordpress.org," https://wordpress.org/, (Accessed on 07/16/2023).

[11] Ghost Foundation, "Ghost: The creator economy platform," https://ghost.org/, (Accessed on 07/16/2023).

[12] Open Source Matters, Inc., "Joomla content management system (cms) - try it! it's free!" https://www.joomla.org/, (Accessed on 07/16/2023).

[13] Q-Success: World Wide Web Technology Surveys, "Usage statistics and market share of wordpress, july 2023," https://w3techs.com/technologies/details/cm-wordpress, (Accessed on 07/15/2023).

[14] National Institute of Standards and Technology, "Nvd - cve-2017-1001000," https://nvd.nist.gov/vuln/detail/CVE-2017-1001000, (Accessed on 07/17/2023).

[15] Q-Success: World Wide Web Technology Surveys, "Usage statistics and market share of ghost, july 2023," https://w3techs.com/technologies/details/cm-ghost, (Accessed on 07/15/2023).

[16] National Institute of Standards and Technology, "Nvd - cve-2021-39192," https://nvd.nist.gov/vuln/detail/CVE-2021-39192, (Accessed on 07/17/2023).

[17] Q-Success: World Wide Web Technology Surveys, "Usage statistics and market share of joomla, july 2023," https://w3techs.com/technologies/details/cm-joomla, (Accessed on 07/15/2023).

**Yuki Ishida** was born in Japan 1991, received the B.E. and M.S. degrees in Informatics from Tokyo University of Information Sciences, Japan, in 2014 and 2016, respectively. Upon graduation in 2016, he joined Digital Arts, Inc., where he contributed to the development of security products. In 2019, he transitioned to SecureBrain Corporation, also in Japan, with a primary focus on research and development in the field of cybersecurity. Since 2022, he has been concurrently enrolled in the doctoral program at the Graduate School of Informatics at Tokyo University of Information Sciences in Japan. His research interests encompass cybersecurity and network quality control. He holds memberships in IEEE, IEICE, and IPSJ.

**Masaki Hanada** was born in Japan 1973, received the B.E. degree in resources engineering from Waseda University in 1996, the M.S. degree in information science from Japan Advanced Institute of Science and Technology (JAIST) in 1999, and the M.S. and D.S. degrees in global information and telecommunication studies from Waseda University in 2003 and 2007, respectively. He worked at Waseda University and Tokyo University of Science. After joining Tokyo University of Information Sciences as an Assistant Professor in 2011, he has been a Professor in the Department of Information Systems, Tokyo University of Information Sciences, since 2019. His research interests include network QoS control, network resource control and management, and network security. He is a member of the IEEE, IEICE and IPSJ.

**Atsushi Waseda** was born in Japan 1977, received the B.E. degree in communication engineering from the University of Electro-Communications in 2000. He received his M.S. and Ph.D. in information science from Japan Advanced Institute of Science and Technology (JAIST) in 2002, and 2007, respectively. He worked at the National Institute of Information and Communications Technology and KDDI research inc. After joining Tokyo University of Information Sciences as an Assistant Professor in 2019. His research interests include information security, quantum security and privacy protection. He is a member of the IEICE and IPSJ.

**Moo Wan Kim** was born in Korea 1951, received B.E., M.E. and Ph.D degree in electronic engineering from Osaka University, Osaka, Japan in 1974, 1977 and 1980, respectively. He joined Fujitsu Lab. in 1980 and had been engaged in research and development on multimedia communication systems, Intelligent Network, ATM switching system and operating system. In 1998 he joined Motorola Japan and had been engaged in research and development on CDMA2000 system. In 2000 he joined Lucent Japan and had been engaged in research and development on W-CDMA system, IMS and Parlay. In 2005 he joined Tokyo University of Information Sciences and had been engaged in research on Ubiquitous Network. In 2022 he joined TA Tech. as a lecture.

Articles in this publication may be cited in other publications. In order to facilitate access to the original publication source, the following form for the citation is suggested:

Name of Author(s), "Title of Paper," Transactions on Advanced Communications Technology, (TACT-ICACT 2024), page numbers

IEEE Catalog Number: CFP24561-ART
ISSN 2288-0003

GIRI
Global IT Research Institute

Supported By

IEEE Communications Society, Gangwon Convention & Visitors Bureau, National Information Society Agency, Electronic and Telecommunications Research Institute, Korea Institute of Communication Sciences, IEEK Communications Society, Korean Institute of Information Scientists and Engineers, Open Standards and Internet Association, Korean Institute of Information Security and Cryptology, Information Technology Institute of Vietnam National University

Global IT Research Institute, All Rights Reserved. Contact: office@icact.org Tel:+82-70-4146-4991
1713 Obelisk, 216 Seohyunno, Bundang-gu, Sungnam Kyunggi-do, Republic of Korea 13591