# Measurement Based Evaluation and Mitigation of Flood Attacks on a LAN Test-Bed


Mohammed Nasereddin and Mert Nakıp
Institute of Theoretical & Applied Informatics
Polish Acad. Sci., 44-100 Gliwice, Poland
{mnasareddin,mnakip}@iitis.pl

Erol Gelenbe
Institute of Theoretical & Applied Informatics
Polish Acad. Sci., 44-100 Gliwice, Poland
& Yaşar University, Bornova, Turkey,
& Lab. I3S Université Côte d'Azur, Nice, France
seg@iitis.pl



***Abstract*—The IoT is vulnerable to network attacks, and Intrusion Detection Systems (IDS) can provide high attack detection accuracy and are easily installed in IoT Servers. However, IDS are seldom evaluated in operational conditions which are seriously impaired by attack overload. Thus a Local Area Network test-bed is used to evaluate the impact of UDP Flood Attacks on an IoT Server, whose first line of defence is an accurate IDS. We show that attacks overload the multi-core Server and paralyze its IDS. Thus a mitigation scheme that detects attacks rapidly, and drops packets within milli-seconds after the attack begins, is proposed and experimentally evaluated.**



***Index Terms*—Internet of Things, Local Area Networks, Cybersecurity, Random Neural Networks, G-Networks, UDP Flood Attacks, Intrusion Detection and Mitigation**


## I. INTRODUCTION

Denial of service (DoS) disables systems or networks by flooding them with huge streams of requests, causing reputational damage, with financial and productivity losses [1]. In the last year, a 150% increase in such attacks occurred worldwide [2], targeting the IoT, industrial control systems, power grids and transportation systems [3]–[6], with DoS and Botnet attacks spreading via their victims, who then become attackers [7]–[9]. Flood attacks [10] overwhelm networks with large numbers of forged-source address packets, causing delayed or lost data, inaccurate or incomplete readings and overload [11]. Thus these threats require effective Intrusion Detection Systems (IDS).

While much of the literature on IDS evaluates them under ideal conditions where attack traffic is treated as data, this paper compares "ideal" results about attack detection (AD) algorithms, with system performance measurements in a LAN environment. Section I-A reviews related work, Section II describes the experimental test-bed that we use and Section II-A briefly discusses the IDS and its performance under ideal and real-world conditions. Section III presents measurements during different UDP Flood attacks, and summarizes improvements resulting from simple attack mitigation. Finally, Section IV concludes the paper and outlines future work directions.

### A. Related Work

Using test-beds to evaluate IDSs was recommended in early work [12], and several test-beds for cyber-physical systems, industrial control and IoT environments have been described

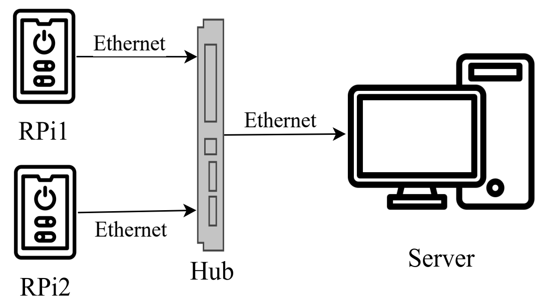


Fig. 1. Testing Environment using Ethernet for communications, with Raspberry Pi machines acting as forwarders of normal and attack traffic, and an Intel 8-Core Processer used as a Server to process incoming packet traffic and run the IDS algorithm.

[13], such as real-time test-bed for cyber-physical systems [14], power systems [15], and wind farms [16]. In [17], a semi-physical test-bed for ICSs was proposed, while in [18], a low-cost Smart Grid test-bed for SCIDS systems was evaluated for TCP flood attacks. Test-beds for SCADA systems are discussed in [19]–[21]. In [22], a test-bed using six NetFlow tools for collecting, analyzing, and displaying data was proposed for HTTP-GET flood attacks in a WAN. In [23], the impact of attack datasets on IoT systems is discussed and real-time data collection for DNS amplification is investigated, while DoS attacks on software defined networks are discussed in [24]. In [25], DoS attacks on an autonomous vehicle test-bed are described. Attack datasets are reviewed in [22], [23], while the present paper uses the MHDDoS repository [26] of real-world DoS attacks.

## II. EXPERIMENTAL SETUP

In this paper, a physical test-bed environment is constructed to evaluate IDS algorithms in more realistic conditions, with an arbitrary number of linked devices, multiple sources of normal and attack traffic, and a Server that supports the UDP Protocol for incoming traffic, runs the IDS algorithm, and processes the incoming packets' contents. The devices that generate benign or attack traffic, are embodied by Raspberry Pi 4 Model B Rev 1.2 machines (RPi1 and RPi2), each with a 1.5GHz ARM Cortex-A72 quad-core processor and 2GB LPDDR4 – 3200 SDRAM, running Raspbian GNU/Linux 11 (bullseye), a Debian-based operating system for the Raspberry Pi hardware. A Server with eight Intel Core i7 $-$ 8705G processors acts as the receiver of the packet traffic and is responsible for detecting the attack and storing the arriving packets. It has 16GB of RAM, a 500GB hard drive, and runs Linux $5.15.0-60-$ generic $66-$Ubuntu SMP, an Ubuntu-based operating system. Its cores run at 3.10GHz. UDP traffic is carried over Ethernet connections between all devices via the Hub in Figure 1, without ACKs or error recovery [27].

### *A. The IDS and its Ideal Performance*

The IDS used in this paper, based on the Deep Random Neural Network (DRNN) [28], is shown in Figure 2. It **learns from the** first 500 benign packets received by the Server, with metrics $x^i = [x_i^1, x_i^2, x_i^3]$ related to successive sets of packets. The IDS predicts the expected metrics: $\hat{x}_i = [\hat{x}_i^1, \hat{x}_i^2, \hat{x}_i^3]$, and the difference between its input and the prediction yields the decision variables $y_i$ (attack or non-attack), which are also used to update the algorithm's weights. The IDS uses the DRNN [29], a Random Neural Network [30] with soma-to-soma triggering between neurons. This IDS provides accurate detection with different datasets [31]–[34], with the excellent statistical performance shown in Figure 3, which reports the Accuracy, TPR, and TNR, for a 10 second attack. This IDS attains high accuracy both when a fixed threshold $\gamma = 0.3$ is used, and for the best threshold $\gamma = 0.3787$, exhibiting 99.7% Accuracy and TPR, while TNR is 98.48%. These results are not significantly different when the attack lasts for 60 seconds. Note that Figure 4 shows that, due to the decision delay, the IDS may raise an alarm just after an attack ceases.

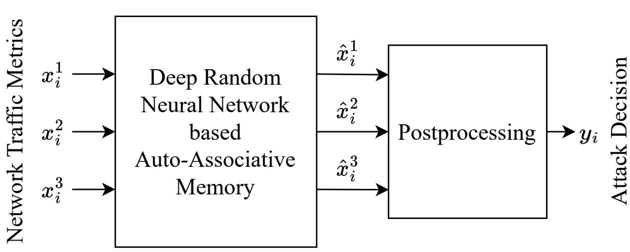


Fig. 2. The structure of the IDS system that computes the decision variable $y_i$ from the network traffic metrics $[x_i^1, x_i^2, x_i^3]$ with the DRNN based Auto-Associative Random Neural Network (AAD RNN) and the postprocessing module.

## III. Experiments with Normal and Attack Traffic

In Figure 5, the Server receives packets from linked devices on port 5555, which are then passed to the buffer manager by the network protocol, and queued for analysis at the IDS, whose decisions are based on the average of a batch of 10 successive packets, that are being classified as normal or attack traffic. Packets classified as "normal" are forwarded to the packet content processor for the rest of the Server's operations.

When there is no attack, each RPi device generates normal IP packet traffic containing the device's own CPU temperature and transmits it to the Server every (1) second using UDP.

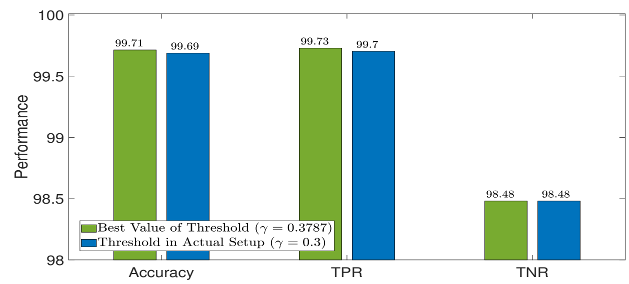


Fig. 3. The performance of the IDS with $\gamma = 0.3$, and compared with the best value of $\gamma = 0.3787$, is evaluated for Accuracy, TPR, and TNR, in an experiment where RPi2 starts a UDP Flood attack lasting 10 seconds.

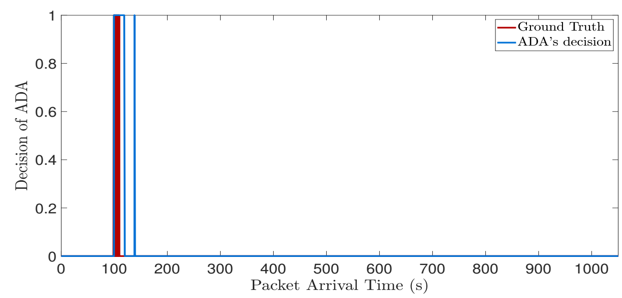


Fig. 4. The IDS's binary decisions are shown for $\gamma = 0.3$, when the RPi2 starts a UDP Flood attack lasting 10 seconds.

RPi2 is programmed to send both normal and attack traffic via random sampling, and generates attacks from the public repository MHDDoS [26]: each 1 second, it initiates a UDP Flood attack with a probability of 0.10, or sends normal traffic packet with a probability of 0.90. RPi1 only sends normal traffic.

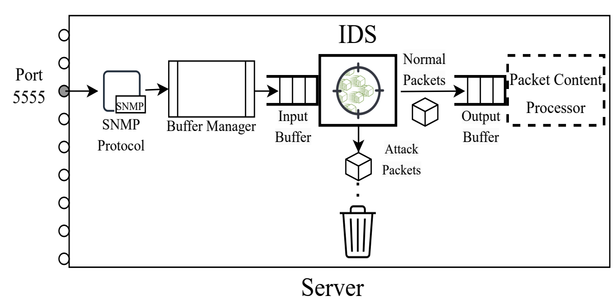


Fig. 5. Schematic organization of the Server that supports the IDS. Mitigation is based on triggering "packet drop" decisions for all packets in the IDS Input Buffer, when it detects a majority of attack packets among the most recent 20 packets. The IDS then resumes testing the incoming packets, and the decision and mitigatiion process is repeated.

Figure 6, displays an intense flow of 1032 byte attack packets during 10 seconds, while "normal" traffic typically consists of two small packets sent each second, for each IoT device. Figure 7 shows that the packet queue length at the Server rises sharply infront of the IDS, with a gradual decrease after the attack. Figure 8 shows (Above) the effect of a 60 second attack on the Server's intermittently paralyzed packet processing rate ($y$-axis in packets/sec), and (Below) the resulting huge input queue length. Figures 7 and 8 show that a 10 second attack floods the packet queue, and the IDS completes the analysis of the accumulated packets over a long 15 minute period, while for a 60 second attack, the IDS is intermittently paralyzed and its analysis can last 5.85 hours, since the Server's cores are busy handling the incoming traffic. Thus the effect of an attack typically lasts far longer than the attacker's activity.

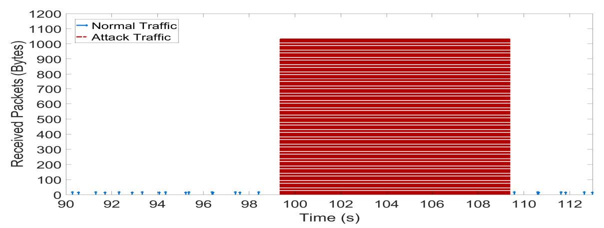


Fig. 6. The difference between the normal and attack traffic on the Server that is targeted by a UDP Flood attack.

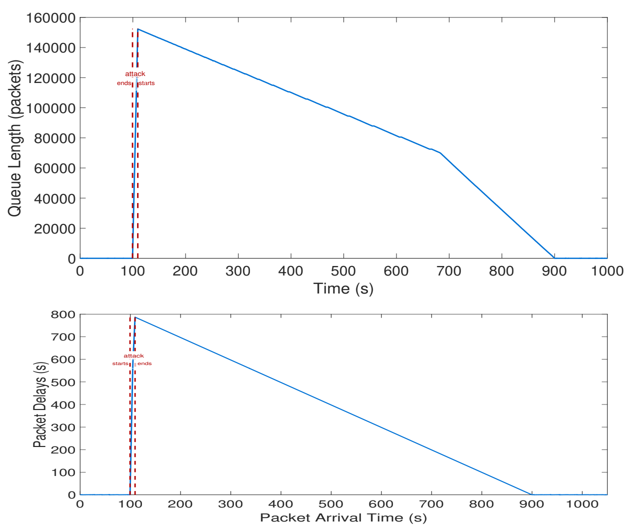


Fig. 7. The top figure shows the queue length infront of the IDS in a 10 second attack, and the vertical red dashed lines show the active duration of the attack originating in the compromised device RPi2. The bottom figure shows the delay before the packet is processed by the IDS.

We now report system measurements when a novel mitigation action occurs: if the IDS decides that the majority of the 20 most recent packets are attack packets, then the input buffer contents and all incoming packets within the next 30 seconds are dropped. This action is repeated at the end of the 30 second window. Figure 9 displays the queue length in the input buffer when the attack mitigation is performed during a UDP Flood attack which lasts 10 seconds: the queue length increases until the IDS processes 20 packets and decides to empty the buffer, and we observe that the attack is mitigated successfully.

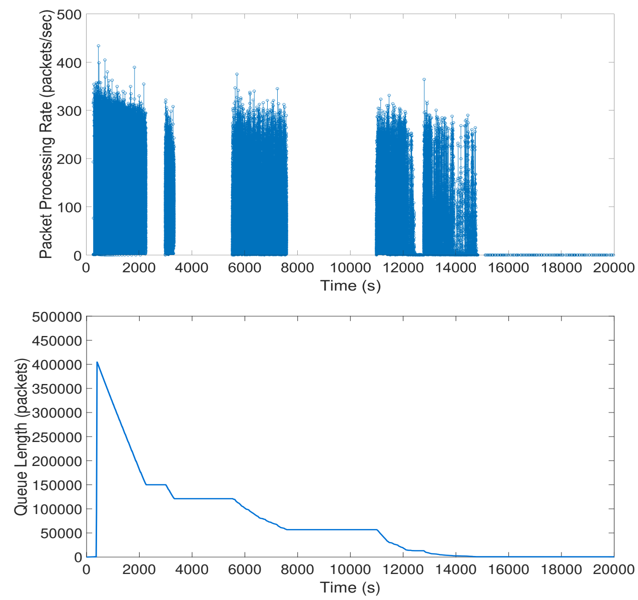


Fig. 8. At the top, the effect of a 60 second UDP Flood attack on the IDS traffic processing rate in packets per second, is shown when the attack duration is 60 seconds. The corresponding packet queue length infront of the IDS is shown at the bottom.

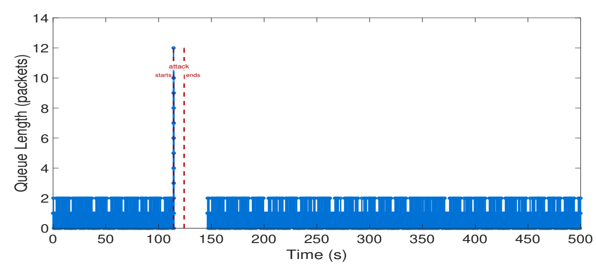


Fig. 9. During the 10 second attack, the decision to drop packets results in a very short packet queue length, avoiding Server and IDS paralysis.

Figure 10 shows the queue length when an attack that lasts 60 seconds is mitigated: the buffer length increases to 22 packets, which is small compared to the value without mitigation shown in Figure 8: the mitigation decision was taken twice, the second time between 162 to 192 seconds after the start of the experiment, and the Server could then operate normally without being paralyzed.

## IV. Conclusions and Future Work

Despite the high accuracy of an IDS installed on a Server that receives traffic from devices in a LAN network test-bed subjected to UDP Flood attacks, we observe that while short attacks are accurately detected, longer attacks may be detected with greater delay due to Server overload. Thus fast mitigation is proposed to discard attacking traffic with rapidly taken decisions. Future work will study mitigation to optimize the traffic that is discarded, the frequency with which the IDS analyzes incoming traffic, as well as the minimization of benign traffic loss, and the system's energy consumption [35].

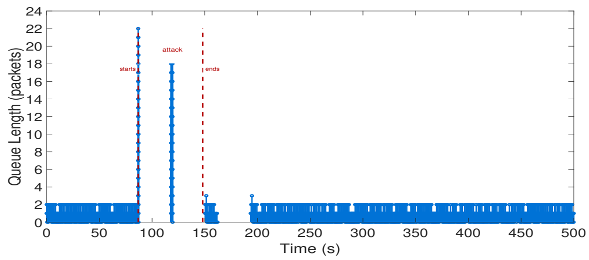


Fig. 10. The figure shows that during the attack's 60 seconds, the mitigation decision occurs twice, with the second mitigation occurring after a detection that takes place between 162 and 192 seconds.

**Acknowledgment** Support from the European Commission's H2020 IoTAC Project GA No. 952684, and H2020 DOSS Project GA No. 101120270, is gratefully acknowledged.

## REFERENCES


[1] E. Gelenbe, P. Campegiani, T. Czachórski, S. K. Katsikas, I. Komnios, L. Romano, and D. Tzovaras, Eds., *Security in Computer and Information Sciences, First International ISCIS Security Workshop, Euro-CYBERSEC 2018, London, February 26-27, 2018*. Springer, Cham, 2018. [Online]. Available: http://library.oapen.org/handle/20.500.12657/23295

[2] S. Staff, "Organizations fought an average of 29.3 attacks daily in late 2022," Feb 2023. [Online]. Available: https://www.securitymagazine.com/articles/98958-organizations-fought-an-average-of-293-attacks-daily-in-late-2022

[3] G. Öke, et al., "Detecting denial of service attacks with bayesian classifiers and the random neural network," in *IEEE International Fuzzy Systems Conf.* IEEE, 2007, pp. 1–6.

[4] S. Evmorfos, et al, "Neural network architectures for the detection of SYN flood attacks in IoT systems," in *Proceedings of the 13th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, 2020, pp. 1–4.

[5] L. Rajesh and P. Satyanarayana, "Detecting flooding attacks in communication protocol of industrial control systems," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 1, pp. 396–401, 2020.

[6] Y. Al-Hadhrami and F. K. Hussain, "Ddos attacks in iot networks: a comprehensive systematic literature review," *World Wide Web*, vol. 24, no. 3, pp. 971–1001, 2021.

[7] N. Statt, "How an army of vulnerable gadgets took down the web today," October 2016. [Online]. Available: https://www.theverge.com/2016/10/21/13362354/dyn-dns-ddos-attack-cause-outage-status-explained

[8] B. Tushir, H. Sehgal, R. Nair, B. Dezfouli, and Y. Liu, "The impact of DoS attacks on resource-constrained IoT Devices: A study on the Mirai attack," *arXiv preprint arXiv:2104.09041*, 2021.

[9] H. Sinanović and S. Mrdovic, "Analysis of Mirai malicious software," in *2017 25th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. IEEE, 2017, pp. 1–5.

[10] Cloudflare. [Online]. Available: https://www.cloudflare.com/learning/ddos/udp-flood-ddos-attack/

[11] J. Mirkovic, et al., "Accurately measuring denial of service in simulation and testbed experiments," *IEEE Transactions on Dependable and Secure Computing*, vol. 6, no. 2, pp. 81–95, 2008.

[12] J. Mirkovic, S. Fahmy, P. Reiher, and R. K. Thomas, "How to test dos defenses," in *2009 Cybersecurity App. & Technology Conf.* IEEE, 2009, pp. 103–117.

[13] O. A. Waraga, M. Bettayeb, Q. Nasir, and M. A. Talib, "Design and implementation of automated iot security testbed," *Computers & Security*, vol. 88, p. 101648, 2020.

[14] C. B. Vellaithurai, S. S. Biswas, and A. K. Srivastava, "Development and application of a real-time test bed for cyber-physical system," *IEEE Systems Journal*, vol. 11, no. 4, pp. 2192–2203, 2015.

[15] A. Ashok, et al., "Testbed-based performance evaluation of attack resilient control for agc," in *2016 Resilience Week (RWS)*. IEEE, 2016, pp. 125–129.

[16] V. K. Singh, R. Sharma, and M. Govindarasu, "Testbed-based performance evaluation of attack resilient control for wind farm scada system," in *2020 IEEE Power & Energy Society General Meeting (PESGM)*. IEEE, 2020, pp. 1–5.

[17] M. Kaouk, F.-X. Morgand, and J.-M. Flaus, "A testbed for cybersecurity assessment of industrial and iot-based control systems," in *Lambda Mu 2018-21è Congrès de Maîtrise des Risques et Sûreté de Fonctionnement*, 2018.

[18] M. Annor-Asante and B. Pranggono, "Development of smart grid testbed with low-cost hardware and software for cybersecurity research and education," *Wireless Personal Communications*, vol. 101, pp. 1357–1377, 2018.

[19] A. Ghaleb, S. Zhioua, and A. Almulhem, "Scada-sst: a scada security testbed," in *2016 World Congress on Industrial Control Systems Security (WCICSS)*. IEEE, 2016, pp. 1–6.

[20] A. Tesfahun and D. L. Bhaskari, "A scada testbed for investigating cyber security vulnerabilities in critical infrastructures," *Automatic Control and Computer Sciences*, vol. 50, pp. 54–62, 2016.

[21] B. Reutimann and I. Ray, "Simulating measurement attacks in a scada system testbed," in *Critical Infrastructure Protection XV: 15th IFIP WG 11.10 International Conference, ICCIP 2021, Virtual Event, March 15–16, 2021, Revised Selected Papers 15*. Springer, 2022, pp. 135–153.

[22] S.-U. Park and S.-M. Hwang, "Test bed construction for apt attack detection," *International Journal of Control and Automation*, vol. 11, no. 4, pp. 175–186, 2018.

[23] R. Arthi and S. Krishnaveni, "Design and development of iot testbed with ddos attack for cyber security research," in *2021 3rd International Conference on Signal Processing and Communication (ICPSC)*. IEEE, 2021, pp. 586–590.

[24] A. P. Wright and N. Ghani, "A testbed for the evaluation of denial of service attacks in software-defined networks," in *2019 SoutheastCon*. IEEE, 2019, pp. 1–6.

[25] P. V. Sontakke and N. B. Chopade, "Impact and analysis of denial-of-service attack on an autonomous vehicle test bed setup," in *Proceedings of Third International Conference on Intelligent Computing, Information and Control Systems: ICICCS 2021*. Springer, 2022, pp. 221–236.

[26] "MHDDoS - DDoS Attack Script With 56 Methods," Online, May 2022, accessed: 2023-02-22. [Online]. Available: https://github.com/MatrixTM/MHDDoS

[27] S. Kumar and S. Rai, "Survey on transport layer protocols: Tcp & udp," *International Journal of Computer App.*, vol. 46, no. 7, pp. 20–25, 2012.

[28] O. Brun, et al., "Iot attack detection with deep learning," in *Security in Computer and Information Sciences, First International ISCIS Security Workshop, Euro-CYBERSEC 2018, London, February 26-27, 2018*. Springer, Cham, 2018. [Online]. Available: http://library.oapen.org/handle/20.500.12657/23295

[29] E. Gelenbe and Y. Yin, "Deep learning with random neural networks," in *2016 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2016, pp. 1633–1638.

[30] E. Gelenbe, "Random neural networks with negative and positive signals and product form solution," *Neural computation*, vol. 1, no. 4, pp. 502–510, 1989.

[31] M. Nakip and E. Gelenbe, "MIRAI Botnet attack detection with Auto-Associative Dense Random Neural Network," in *IEEE Global Communications Conference (Globecom)*, 2021, pp. 1–6.

[32] M. Nakıp and E. Gelenbe, "Botnet attack detection with incremental online learning," in *Security in Computer and Information Sciences: Second International Symposium, EuroCybersec 2021, Nice, France, October 25–26, 2021, Revised Selected Papers*, E. Gelenbe, et al., Ed. Springer, 2022, pp. 51–60.

[33] E. Gelenbe and M. Nakıp, "Traffic based sequential learning during botnet attacks to identify compromised iot devices," *IEEE Access*, vol. 10, pp. 126 536–126 549, 2022.

[34] E. Gelenbe and M. Nakıp, "G-networks can detect different types of cyberattacks," in *MASCOTS'22: 30th International Symposium on the Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, IEEE Computer Society*, pp. 1–6.

[35] B. Pernici, et al., "What IS can do for environmental sustainability: a report from caise'11 panel on green and sustainable IS," *Comm. Association for Information Systems*, vol. 30, no. 1, p. 18, 2012.