Actively Detecting Multiscale Flooding Attacks & Attack Volumes in Resource-Constrained ICPS

Farzana Zahid, Matthew M.Y. Kuo, *Member, IEEE*, Roopak Sinha, *Senior Member, IEEE*, Gustavo Funchal, Tiago Pedrosa, and Paulo Leitao, *Senior Member, IEEE*

***Abstract*—The significant growth in modern communication technologies has led to an increase in zero-day vulnerabilities that degrade the performance of *industrial* cyber-physical systems (ICPS). Distributed denial of service (DDoS) attacks are one such threat that overwhelms a target with floods of packets, posing a severe risk to the normal operations of the ICPS. Current solutions to detect DDoS attacks are unsuitable for resource-constrained ICPS. This study proposes actively detecting multiscale flooding DDoS attacks in resource-constrained ICPS by analyzing network traffic in the frequency domain. A two-phased technique detects attack presence and attack volume. Both phases use a novel combination of light-weight and theoretically sound statistical methods. The effectiveness of the proposed technique is evaluated using mainstream metrics like true and false positive rates, accuracy, and precision using BOUN DDoS 2020 and CICDDoS 2019 datasets. An implementation of the proposed approach on a programmable logic controllers-based ICPS demonstrated improvements in resource usage and detection time compared to the existing state-of-the-art.**

***Index Terms*—Discrete Fourier transform, distributed denial of service (DDoS), Euclidean distance, fast-entropy, industrial cyber-physical system (ICPS), Jaccard similarity, resource-constrained.**

I. Introduction

DISTRIBUTED denial of service (DDoS) is an immense threat that disrupts or degrades some or all of the resources of *industrial* cyber-physical systems (ICPS) by preventing information distribution over networks [16], [17]. ICPS are highly distributed systems with numerous components interacting with each other and the physical environment. Inherently, ICPS, are resource-constrained with limited bandwidth, processing capacity, memory, and security capabilities [5]. Programmable logic controllers (PLCs) are industrially hardened but resource-constrained computers on which ICPS software is deployed to control physical processes. DDoS attacks can quickly overwhelm PLCs, causing substantial disruptions. Protecting PLCs from DDoS is challenging and has not received significant research attention [18]. To the best of our knowledge, no existing work detects flooding attacks within PLC-based ICPS.

DDoS attacks exploit resource limitations or vulnerabilities and cause delays, bandwidth depletion, buffer overflows, and crashes [19]. DDoS attacks are categorised based on attack rate (high-rate or flooding attack and low-rate DDoS) or their impact (resource-depletion or bandwidth depletion attacks) [17] or in terms of endpoint (devices/applications in the physical and cyber layers), and network (communication layer) of ICPS [20]. A recent analysis of DDoS attack trends shows that these attacks are evolving rapidly in terms of their types, volume [increased by 89% quarter-on-quarter (QoQ)], and rates, which have increased dramatically [21]. The in-time detection of DDoS attacks, particularly multiscale flooding attacks, is critical and challenging due to their destructive impact on performance and safety [12], [20]. "Multiscale" refers to varying aspects of the attack traffic, including predictable and unpredictable attack rates, attack periods, attack densities, peaks, and attack volume (magnitude/sheer number of attacks over a given time-interval) [21], [22].

This work explores the following research questions.

- RQ1 What are the existing DDoS flooding attack detection strategies suitable for ICPS applications?
- RQ2 What are the limitations of the works identified in RQ1 in resource-constrained ICPS applications?
- RQ3 How can a light-weight active security technique be provided in resource-constrained ICPS to deal with multiscale flooding attacks and attack volumes?
- RQ4 How can the technique proposed in answering RQ3 be evaluated for its feasibility and effectiveness with respect to the works identified in RQ1?

We answer RQ1 and RQ2 through a literature survey presented in Section II. RQ3 and RQ4 are answered (Sections III–V) through an adapted design science research methodology to build and test the light-weight technique for detecting multiscale flooding attacks in resource-constrained ICPS applications.

Our *survey results (Section II) indicate that the optimal solution for detecting multiscale flooding attacks in resource-constrained ICPS is still an open research problem* [8], [16]. Existing works use anomaly-based, signature-based, or

Manuscript received 8 September 2023; revised 15 December 2023; accepted 28 February 2024. Paper no. TII-23-3448. *(Corresponding author: Farzana Zahid.)*

Farzana Zahid is with the School of Computing and Mathematical Sciences, University of Waikato, Hamilton 3216, New Zealand (e-mail: farzana.zahid@waikato.ac.nz).

Matthew M.Y. Kuo is with the School of Engineering, Computer and Mathematical Sciences, Auckland University of Technology, Auckland 1010, New Zealand (e-mail: matthew.kuo@aut.ac.nz).

Roopak Sinha is with the School of Information Technology, Deakin University, Melbourne, VIC 3125, Australia (e-mail: roopak.sinha@deakin.edu.au).

Gustavo Funchal, Tiago Pedrosa, and Paulo Leitao are with the Research Centre in Digitalization and Intelligent Robotics (CeDRI), Instituto Politecnico de Braganca, 5300-253 Braganca, Portugal (e-mail: gustavofunchal@ipb.pt; pedrosa@ipb.pt; pleitao@ipb.pt).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2024.3383520.

Digital Object Identifier 10.1109/TII.2024.3383520

TABLE I
STATE-OF-THE-ART DDOS FLOODING ATTACKS DETECTION STRATEGIES BASED ON DETECTION METHODS, DETECTION FEATURES, ATTACK VOLUME (YES/NO), DOMAIN ANALYSIS (TIME/FREQUENCY), RESOURCE-CONSTRAINED (YES/NO), THRESHOLD [STATIC/DYNAMIC/NOT-APPLICABLE (NA)], DATASETS USED [NAME/NO (DATASET NOT USED)]

Ref[1]	Detection Method(s)	Detection feature(s)	Vol[2]	Dom[3]	Res[4]	Th[5]	Datasets used
[1]	Hierarchical Bayesian network	Src[6], Dest IP[7], P[8], service types, duration	No	T[9]	No	NA	NSL-KDD
[2]	Krill herd optimization, bi-LSTM	flow and connection data	No	T	No	NA	NSL-KDD
[3]	Matrix multiplication	control data	No	T	No	NA	No
[4]	Shannon entropy	Src, Dest IP	No	T	No	D	No
[5]	QuickDFT	Timestamp	No	Freq[10]	Yes	NA	BOUN DDoS 2020
[6]	Federated learning, neural network	flow data	No	T	No	S	No
[7]	Entropy, sequential probability ratio	Dest IP	No	T	No	S[11]	DARPA98, DARPA2000, CI-CDDoS 2019
[8]	Renyi entropy, stochastic gradient	Src, Dest IP, Src, Dest ports, P, packet size	No	T	No	D[12]	MIT98, CAIDA2007
[9]	Fuzzy logic, neural network	flow rate, pressure	No	T	No	NA	No
[10]	Random conditional probability	Control data	No	T	No	S	No
[11]	Bayesian-based search, score technique	not mention	No	T	Yes	NA	No
[12]	Mean, variance	Src, Dest IP, Src, Dest ports, P	No	T	No	D	DARPA2000, DARPA98
[13]	DFT, sparse representation model	Timestamp	No	Freq	No	S	CAID2011
[14]	Generalised entropy, information distance	Src IP	No	T	No	S	MIT98,CAIDA2007, FIFA98
[15]	Shannon entropy	Src, Dest IP, Src, Dest ports, P	No	T	No	D	No
Our work	Fast-entropy, Jaccard similarity, Euclidean distance	*Timestamp*	*Yes*	*Freq*	*Yes*	*D*	*CICDDoS 2019, BOUN DDoS 2020*

[1]References, [2]Attack volume, [3]Domain analysis, [4]Resource-constrained, [5]Threshold, [6]Source, [7]Destination IP, [8]Protocol, [9]Time, [10]Frequency, [11]Static, [12]Dynamic

hybrid-based detection approaches [17], [23]. These security approaches are heavy-weight, not memory-efficient, require significant processing capacity, and have high false positive rates. These limitations make the existing works ineffective, difficult to determine the flooding attacks in time, and expensive to develop and maintain [4], especially in resource-constrained ICPS [16]. Moreover, attack volumes indicate attack intensity. A higher attack volume typically implies a more significant threat. Thus, *resource-constrained ICPS require light-weight mechanisms to actively (dynamically and programmability) detect multiscale flooding attacks and volumes early from the incoming traffic flow* [5].

This study aims to propose a light-weight active security technique to detect multiscale flooding attacks and volumes in resource-constrained ICPS to address the identified issues. The proposed technique analyzes the network traffic in the frequency domain. It is based on straightforward spectral analysis and statistical approaches, which allow fast and accurate detection of DDoS attacks [24]. Frequency domain analysis is a promising approach that provides a better understanding of network traffic, which is often not possible via time domain analysis and enables accurate and robust attack detection using the unique frequency signature of abnormal behavior [22].

The rest of this article is organized are as follows.

1) A systematic literature review of existing DDoS flooding attack detection strategies (Section II).
2) A novel two-phased light-weight active security technique to dynamically detect multiscale flooding attacks and attack volumes in resource-constrained ICPS (Section III).
3) A novel multimethod approach to dynamically detect the presence of flooding attacks in PLC-based ICPS (Section III).
4) Experimental validation of our proposed approach on publicly available datasets: Boğaziçi University distributed denial of service (BOUN DDoS) dataset [25] and CICDDoS 2019 [26] (Section V).
5) Performance (memory and CPU overhead) evaluation of the proposed approach (Section V). Finally, Section VI concludes this article.

II. RELATED WORKS

Traditional security mechanisms like firewalls, routers, or load balancers are ineffective in resource-constrained ICPS [16], [18] as they do not prioritize memory efficiency or resource optimization. These mechanisms rely on prior knowledge of attack signatures (specialized patterns signalling threats); consequently, new and unseen attacks go unnoticed [18]. Various DDoS attack detection surveys exist [17], [19], [20], [23], but none consider resource-constrained ICPS.

Our survey includes generic and ICPS-specific techniques for detecting flooding attacks. Table I shows a comparison of various state-of-the-art DDoS detection strategies.

Several works employ information entropy, statistical dispersion, similarity/dissimilarity, machine learning, hybrid and knowledge-based methods for attack detection. Information entropy-based detection mechanisms measure changes in the randomness of network traffic. However, most techniques are not ICPS-relevant, have low detection rates and/or have high

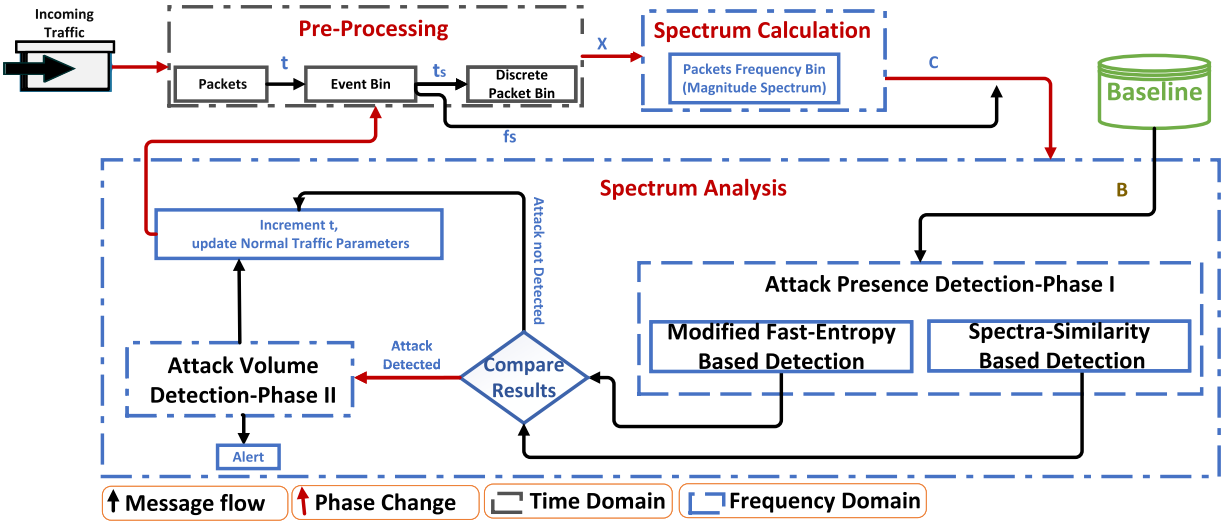

Fig. 1. Proposed model for multiscale flooding attacks and attack volumes detection.

computational costs [8]. Statistical dispersion methods become computationally time and resource-intensive when large numbers of network traffic detection features (inspection and analysis of both header and payload of packets) are considered [24]. Using static thresholds is computationally inexpensive, but such techniques cannot adapt to changing traffic characteristics and are ineffective in dealing with new attacks [4]. Spectral analysis can detect trends and irregularities in network traffic effectively [13], [22], but it has not been used in resource-constrained ICPS.

Existing techniques for detecting DDoS attacks in ICPS range from machine learning, signature-based, and knowledge-based approaches [3], [5], [6], [9], [10], [11]. These techniques require baseline information like historical data, models, forecasts or rules, and dictionaries of known attacks. These methods are not well-suited for resource-constrained ICPS because of high resource usage, and inability to detect previously unknown attacks. Another work [5] employs frequency analysis for DDoS attack detection in ICPS; however, detection relies on human observation, and attack detection time is high.

Most current works are validated against outdated datasets, making them ineffective against more sophisticated flooding attacks [4]. Also, many methods use time domain analysis of network traffic and cannot detect patterns hidden in a noisy environment [22]. Moreover, researchers proposed a single method of detection that may have limited effectiveness in a changing environment. Table I shows that most works detect the presence of attacks in malicious traffic but do not identify attack volume. No current work optimises performance overheads and consequently feature high attack detection times.

In contrast to existing works, we propose a novel technique that combines multiple methods and optimisations to efficiently detect multiscale flooding attacks in resource-constrained ICPS.

III. Proposed Light-Weight Multiscale Flooding Attacks and Attack Volumes Detection Technique

The proposed attack detection technique comprises *three major components: preprocessing, spectrum calculation, and spectrum analysis*, as illustrated in Fig. 1.

A. Preprocessing and Spectrum Calculation

Preprocessing involves the transformation of raw network traffic into a meaningful form for accurate analysis of malicious behavior or to uncover the hidden patterns or trends in the traffic.

This work employs a computationally efficient *binning method* for traffic preprocessing. The binning method ensures that the traffic is appropriately processed for accurate analysis in later steps of the proposed techniques. Binning involves extracting discrete packets from the network over a time window and distributing them into contiguous and equally sized event bins. The flooding attacks exhibit specific temporal traits; thus, binning methods help to identify the deviations or trends over different time-intervals.

During traffic preprocessing, the first and last packet arrival times (t_{first} and t_{last}) and the total number of packets captured (p_{total}) are used to create equally sized bins x_i with the duration t_s (dynamic sampling time-period). $X = [x_1, x_2, \ldots x_n]$ is the array of all bins over a time-window t_w. The dynamic sampling time-period is computed by $t_s = 1/(2 \times f_s)$ where $f_s = p_{\text{total}}/(t_{\text{last}} - t_{\text{first}})$. $2 \times f_s$ represents the Nyquist sampling frequency. Each bin stores a count of the number of packets within the time period t_s.

Spectrum calculation uses a light-weight quick discrete Fourier transform (QuickDFT) approach to transform the bins of packet counts in the time domain into the frequency domain [5]. $C = \text{QuickDFT}(X)$ where $C = [c_1, c_2, \ldots, c_n]$ is the magnitude array of the frequency domain representation of the packets over the time-interval t_w [5].

Baseline data: Our detection technique detects anomalies or deviations from a baseline array ($B = [b_1, b_2, \ldots, b_n]$), representing normal traffic where the system is not under attack. The baseline array is an iteratively updated array storing the magnitude values of baseline traffic over the frequency domain. Further detail about baseline data is in Section IV-A.

B. Spectrum Analysis

The spectrum analysis component comprises two phases: *attack presence* and *attack volume detection*. Attack presence detection is further divided into two subphases, shown in Fig. 1.

Spectra-similarity-based detection uses the *Jaccard similarity metric* [27] to compare the similarity between normal and incoming traffic magnitude spectra with high accuracy. Jaccard similarity was selected due to its lower sensitivity to noise and based on the higher detection accuracy after extensive experimentation over various metrics, including intersection, Kulczynski, Soergel, Canberra, Czekanowski, Tanimoto, and Hellinger [27]. *Modified fast-entropy-based detection* uses the *fast-entropy method* [28] to monitor changes in entropy (uncertainty) of the magnitude spectrum of traffic through flooding attacks. Utilizing the multimethod approach in phase 1 improves the effectiveness of flooding attacks detection by providing distinct abnormal traffic behavior and increasing the confidence level in detection results. Both proposed methods contribute to identifying the flooding attack by analyzing the magnitude spectrum in the frequency domain, but their applications differ. Jaccard similarity indicates the potential attack based on the deviation of incoming traffic from the established baseline. The modified fast-entropy detection focuses on the randomness of incoming traffic during the attack based on the calculated fast-entropy value.

The second phase, *volume detection*, uses Euclidean distance (a dissimilarity metric) as the primary measure [27]. Euclidean distance is computationally simple and can analyze the correlation information more efficiently and effectively, even over a few features [27]. The following analysis is all performed over the frequency domain.

Spectra-similarity based detection: The Jaccard similarity is computed over the baseline B and incoming traffic spectra C using (1) where $n = \max(|C|, |B|)$ [27]

$$J = \frac{\sum_{k=1}^{n} c_k.b_k}{\sum_{k=1}^{n}(c_k)^2 + \sum_{k=1}^{n}(b_k)^2 - \sum_{k=1}^{n} c_k.b_k}. \tag{1}$$

Trailing zeros are padded to ensure the two arrays are of the same size. Perfect similarity is indicated by 1, while 0 means no similarity [27]. *An attack is present if the value of J lies within the range of 0 to 0.5*. The intention behind establishing this range is to provide a clear distinction between normal and malicious behavior and effectively capture the deviation of incoming network traffic from the baseline.

Modified fast-entropy method: The entropy value describes the dispersion or concentration of network traffic features [8]. The higher the concentration, the lower will be the entropy value. In flooding attacks, the number of packets increases significantly for a certain time-period, which results in a dominant peak that indicates a dramatic fall in fast-entropy value.

To reduce the computation time, we first create a subarray of C, denoted as Z, for the values greater than or equal to a lower control threshold β [calculated by using (8) discussed in Section IV-B]. Let $Z = [z_1, z_2, \ldots, z_k]$ where $\forall z \geq \beta$ be the subarray of C.

The fast-entropy values $H = [h_1, h_2, \ldots h_k]$ for each corresponding magnitude value in Z is calculated by (2) where h_j is the fast-entropy value of z_j

$$h_j = -\log \frac{z_j}{\sum_{i=1}^{|Z|} z_i} + \lambda_j$$

where

$$\lambda_j = \begin{cases} |\log \frac{z_j}{z_j + z_{j+1}}|, & \text{if } z_j \geq z_{j+1} \\ |\log \frac{z_j + z_{j+1}}{z_j}|, & \text{if } z_j < z_{j+1}. \end{cases} \tag{2}$$

λ_j is the fast-entropy calibration factor based on [28]. In our study, to increase the accuracy of the multiscale attack detection, we have modified the existing fast-entropy calibration factor (λ_j) empirically using Shannon's fundamental properties of information content [29]. *The modified fast-entropy method detects an attack if incoming traffic's fast-entropy rate (δ_Z) is less than the flooding attack confirmation threshold (Θ)*. Generally, the fast-entropy rate (δ) is the average of all entropy values H [15]

$$\delta = \frac{\sum_{k=1}^{|H|} h_k}{|H|}. \tag{3}$$

An attack is detected if δ_Z is less than a threshold Θ [see (9) in Section IV-B].

Attack volume detection: Fig. 1 shows that the attack volume detection phase triggers based on the true output of the attack presence detection phase. Euclidean distance is used to determine the attack volume, i.e., one-time extreme peak (one dominant peak) or peak volume (multiple dominant peaks) by using the following:

$$d = |a_1 - a_2| \tag{4}$$

where a_1 and a_2 are two points on the real line in one dimension (either x-axis or y-axis). As we are only interested in the dominant peaks, we created a subarray called $\hat{H}$ from H where the values are above the upper control threshold (ξ) [calculated using (8) in Section IV-B]. Let $\hat{H} = [\hat{h}_1, \hat{h}_2, \ldots \hat{h}_m]$ where $\forall \hat{h}_j \geq \xi$.

The attack volume (A_{vol}) is calculated with the Euclidean distance d between each fast-entropy value in $\hat{H}$ and the dominant peak's fast-entropy value (ζ_Z) where ζ_Z=min(H). The peak volume is identified if the d values lie within the range from 0 to less than 0.5. The distance 0 means the peaks are close (have the same amplitude). An alert will be generated to give the information about the identified dominant peaks responsible for an attack. Equation (5) is performed for all values in $\hat{H}$

$$\text{alert} = |\zeta_Z - \hat{h}_j| < 0.5. \tag{5}$$

IV. Initialization of Baseline, Parameters, and Thresholds Computation

A. Initialization of Baseline and Parameters

The baseline is built using normal network traffic within the system. First, a list of magnitude arrays within each bin is collected, representing normal traffic across different time-intervals. The mean of each magnitude array is calculated, and the array with the maximum mean is selected as baseline (B). The maximum mean demonstrates the highest value (number of packets/magnitude) for legitimate traffic at a particular destination IP. Fig. 2(a) shows the illustration of the baseline.

This baseline magnitude array is used to calculate various parameters, such as baseline fast-entropy values (H_B) using (2), baseline fast-entropy rate δ_B using (3), and baseline standard deviation σ_B used to compute the flooding attack confirmation threshold (Θ) using (9) (see Section IV-B).

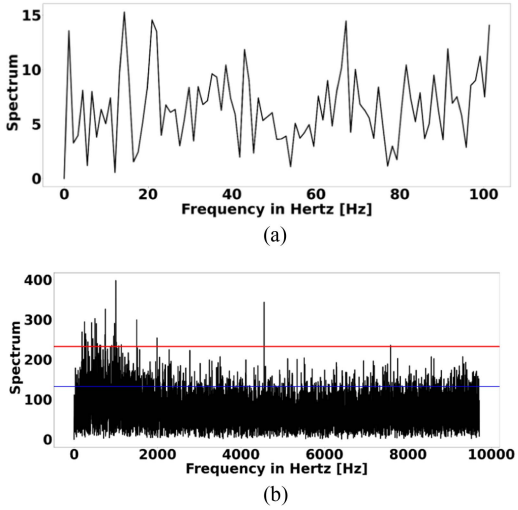

Fig. 2. Illustration of baseline and incoming traffic magnitude spectra. (a) Baseline. (b) Incoming traffic with β (blue line) and ξ (red line).

TABLE II
INFORMATION ABOUT USED DATASETS

Datasets	Attack Frequency	Attack Start-Time	Attack End-time	Destination IP Address
BOUN_1	1000 Hz	75.5356	105.685	10.50.199.99
BOUN_2	2000 Hz	280.594	303.162	10.50.199.99
BOUN_3	1500 Hz	180.942	203.552	10.50.199.99
CICDDoS	Variable	12:30:00	14:35:00	192.168.50.4
CICDDoS	Variable	10:30	13:30	192.168.50.1

The baseline is updated when no attack is detected, and the mean of the current magnitude array C is greater than that of the baseline. The baseline is then updated as $B = C$. The various parameters are also subsequently updated.

B. Thresholds Computation

For accurate attack detection, *threshold(s) identification and setting the values* are challenging tasks [15]. The thresholds used in this study are dynamic (change with the changes in network traffic flow) and determined by the traffic distribution.

Our approach uses a *lower control threshold (β), and flooding attack confirmation threshold (Θ)* in phase I during modified fast-entropy-based detection. To determine the peak volumes at phase II, an *upper control threshold (ξ)* is identified. In essence, the amplitudes of very high frequencies are the harmonics or noise which contain unimportant information [5]; therefore, we created β to remove unimportant information (noise/harmonics) and to eliminate the chances of false positives and negatives. The purpose of establishing an upper control threshold (ξ) is to identify attack volumes in resource-constrained environments while minimising the number of computations.

As our purpose is to make a robust and simple approach for resource-constrained ICPS, we have used Chebyshev's Inequality [30] to determine the upper and lower control thresholds. Chebyshev's Inequality is a valuable and flexible approach to assessing the proportion of observations that fall within a z (integer) range of standard deviations, as shown in the following [30]:

$$[\mu - z.\sigma, \mu + z.\sigma]. \tag{6}$$

During the flooding attack, the attack spectrum has higher amplitudes than the legitimate spectrum because attack network traffic contains more packets [5]. Therefore, we have used the upper bounds of the interval both for the lower control (β) and upper control (ξ) thresholds. In addition, to overcome the issue of false alarm rates, we have also introduced the threshold calibration factor called Freeman's index (v) with Chebyshev inequality. This factor is calculated by the following:

$$v = \frac{|1 - f_s|}{|C|}. \tag{7}$$

Thus, β and ξ are calculated for C by combining the upper bounds of (6) with (7), as shown in the following:

$$[\mu_C + z.\sigma_C] + v. \tag{8}$$

Our study detects an attack by comparing incoming traffic with the baseline distribution. Therefore, the modified fast-entropy method signals the presence of an attack by computing the flooding attack confirmation threshold (Θ) from the normal baseline behavior of the network. Θ is calculated from the multiple of standard deviation (σ_B) of B with baseline fast-entropy rate (δ_B) using the following:

$$\Theta = \delta_B * \sigma_B. \tag{9}$$

The selection of appropriate threshold values is essential for the accurate and precise detection of DDoS attacks in the resource-constrained ICPS. The sole purpose is to decrease the rate of false alarms. The selection of threshold values is discussed in Section V-C.

V. EXPERIMENTAL ANALYSIS AND DISCUSSION

A. Experimental Configurations and Datasets

Our proposed approach was implemented on Raspberry PLC 50RRA with 4 GB RAM. The proposed algorithm was codified in Python language and run many times on two different publicly available DDoS datasets: BOUN DDoS 2020 [25], and CICDDoS 2019 [26], which are used to detect different types of DDoS attacks. Note, we will use BOUN DDoS and CICDDoS to represent BOUN DDoS 2020 and CICDDoS 2019 datasets, respectively.

The BOUN DDoS and CICDDoS datasets include the recent DDoS attack vectors or types. Several studies have used them for real-world intrusion detection [5], [7]. The BOUN DDoS dataset was generated to detect flooding DDoS attacks like TCP-SYN and UDP flooding. In contrast, the CICDDoS dataset was used to detect exploitation and reflection-based DDoS attacks. For our work, shown in Table II, we have categorized the BOUN DDoS dataset as BOUN_1 for attack period 1 (75.535–105.685 s) with attack frequency 1000 Hz, BOUN_2 for attack period 3 (280.594–303.162 s) having an attack frequency 2000 Hz and BOUN_3 for attack period 2 (180.942–203.552 s) having an attack frequency 1500 Hz. Similarly, the CICDDoS dataset contains the SYN flooding attack samples captured at 12:30:00–14:35:00 on 12th January and 10:30–13:30 on 11th March.

Moreover, most of the existing works merged two different datasets to differentiate the legitimate and attack traffic, which

TABLE III
SIMULATION RESULTS FOR UNPREDICTABLE ATTACKS IN TEN BINS UTILIZING CICDDOS DATASET

Results	Bin 1	Bin 2	Bin 3	Bin 4	Bin 5	Bin 6	Bin 7	Bin 8	Bin 9	Bin 10
J	0.1	0.25	0.2	0.1	0.01	0.3	0.25	0.05	0.1	0.2
C_{max}	422.4	328.3	339.8	330	353.5	364.6	439.1	403.4	355.1	397.7
ζ_Z	8.95	9.35	9.31	9.34	9.25	9.16	8.85	9.01	9.20	9.04
δ_Z	6.99	6.96	6.9	6.96	6.93	6.97	6.89	6.99	6.93	6.96
Θ	11.12	11.12	11.12	11.12	11.12	11.12	11.12	11.12	11.12	11.12
No of dominant peaks	7	12	15	17	12	10	3	5	9	9

TABLE IV
SIMULATION RESULTS FOR PREDICTABLE ATTACKS IN TEN BINS UTILIZING BOUN_2 DATASET

Results	Bin 1	Bin 2	Bin 3	Bin 4	Bin 5	Bin 6	Bin 7	Bin 8	Bin 9	Bin 10
J	0	0.1	0.3	0	0.2	0	0.2	0.3	0.1	0.3
C_{max}	772.6	662.9	1115.1	724.06	900.5	703.9	793.3	799.6	886.1	865.3
ζ_Z	3.17	3.5	2.5	3.34	2.8	3.35	3.11	3.04	2.90	2.95
δ_Z	6.0	6.4	6.16	6.24	6.39	6.23	6.03	6.25	6.11	6.22
Θ	15.08	15.08	15.08	15.08	15.08	15.08	15.08	15.08	15.08	15.08
No of dominant peaks	1	2	1	2	1	2	3	1	2	2

TABLE V
DETERMINATION OF VALUES FOR CICDDOS DATASET

Value of z_β	Value of z_ξ	Accuracy	Detection time
1	2	40%	0.8 s
	3	75%	0.75 s
	4	90%	0.65 s
	5	95.5%	0.6 s
	6	5%	0.6 s
2	3	85%	0.7 s
	4	93%	0.63 s
	5	100%	0.6 s
	6	3%	0.59 s
3	4	60%	0.7 s

influence their detection results [4]. However, we have used the same datasets for both normal and attack traffics to analyze our proposed work.

B. Simulation Scenarios and Results

To show the applicability and flexibility of our proposed technique, we experimented with multiscale traffic characteristics and considered three scenarios.

1) Scenario 1- Attack Rates: The packet transmission characteristics determine the attack rates. There are two types of attack rates: predictable and nonpredictable attack rates. The predictable attack rate involves sending the attack packets to the victim according to a predictable pattern. In contrast, the nonpredictable attack rate involves sending the attack packets randomly or at a variable rate to avoid being discovered. In our study, we have used the attack rates of the BOUN DDoS dataset as the predictable DDoS attack rates, and the CICDDoS dataset is used to validate our approach for nonpredictable attack rates. The simulation results of our detection technique for attack rates scenarios are shown in Table VII.

2) Scenario 2- Attack Density: An attack density is a ratio or percentage between the number of attack packets compared to the number of normal packets at the time of attack [25]. The performance of the proposed detection technique for different attack densities is shown in Table VII.

TABLE VI
DETERMINATION OF VALUES FOR BOUN DDOS DATASET

Value of z_β	Value of z_ξ	Accuracy	Detection time
1	2	20%	0.45 s
	3	45%	0.47 s
	4	50%	0.43 s
	5	55%	0.39 s
	6	55%	0.44 s
2	3	82%	0.39 s
	4	95%	0.35 s
	5	100%	0.25 s
	6	80%	0.4 s
3	4	70%	0.38 s
	5	75%	0.38 s

3) Scenario 3- Attack Periods: We have performed experiments with different attack periods, such as 10, 15, 60, 120 s. Table VII depicts the results of different periods for each dataset.

The first step in our experiments is to set the baseline by following the procedure mentioned in Section IV-A. For illustration, the normal magnitude spectrum with the highest mean (μ=6.5) was selected as a baseline in the CICDDoS dataset, as shown in Fig. 2(a). This baseline is used to compute the flooding attack confirmation threshold ($\Theta = 11.12$).

Using the *CICDDoS* dataset, we conducted a simulation of the flooding attack on a victim's destination IP and used our technique to detect the attack. The purpose was to determine the presence of an attack in the incoming traffic. For example, using the steps discussed in Section III-A, we simulate an attack with a period of 60 s and set the window size (t_w) to 1 s. During this attack period, 1 582 112 packets were captured with a 70% attack density, and 60 bins of 1 s were observed. Table III shows the simulation results of 10 bins generated within first 10 s and Fig. 2(b) shows the incoming traffic spectrum for Bin 10. Jaccard similarity between the baseline and incoming traffic is $J = 0.2$, which shows significant variance. Therefore, the system is considered under flooding attack. To further confirm the presence of the flooding attack within the incoming traffic, as shown in Fig. 2(b), the modified fast-entropy was calculated for amplitude values greater than or equal to the lower threshold value $\beta = 121.8$, represented by a blue line in Fig. 2(b). It is evident from Table III that incoming traffic potentially indicates a flooding attack as δ_Z (6.96) is less than Θ (11.12).

After confirming attack presence, attack volume is identified for the values greater than or equal to the upper control threshold value $\xi = 235$, represented by a red line. The dominant peak's fast-entropy value ($\zeta_Z = 9.04$) for the incoming traffic is compared with other fast-entropy values within Bin 10. Finally, an alert will be generated when our proposed approach identifies the number of dominant peaks that are contributing to the intensity of an attack.

Similarly, Table IV shows the simulation results for BOUN_2 dataset for 10 s where attack density was 40% and attack rate was 2000 Hz (predictable attacks). We also applied our approach on BOUN_1 and BOUN_3 datasets and complete simulation results are available online.

C. Threshold Computation

Determining the appropriate values for the lower and upper control thresholds (β and ξ) requires identifying the standard deviation to consider. A range of [-6,+6] is commonly used

TABLE VII
RESULTS FOR DIFFERENT ATTACK RATES, ATTACK DENSITIES, AND ATTACK PERIODS

Scenarios	Attack Rates				Attack Density				Attack Period			
Datasets	BOUN_1	BOUN_2	BOUN_3	CICDDoS	BOUN_1	BOUN_2	BOUN_3	CICDDoS	BOUN_1	BOUN_2	BOUN_3	CICDDoS
	1000	2000	1500	Variable	95 (28%)	120.31 (40%)	55 (9%)	350.10 (70%)	20	15	20	60
TPR	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
FPR	0	0	0	0	0	0	0	0	0	0	0	0
Ac	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
P	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%

in practice. Tables V and VI show empirical simulation results with different values for z_β and z_ξ for the two datasets we used. Optimal selection is based on accuracy. Table V shows that the careful selection of z (z_β and z_ξ) plays an essential role in the reduction of false rates and computation time for the early detection of flooding attacks. If we select a lower value for z, the system becomes more sensitive to minor, regular variations in traffic and generates a false alert, resulting in less accuracy. Similarly, setting a higher value results in an alert only when there is a substantial deviation from the baseline, making it challenging to sense more subtle deviations.

Algorithm 1 shows the steps to determine z_β and z_ξ representing ranges to compute the thresholds z_ξ for computing ξ. The lower threshold β is used to remove noise, and its computation is limited to 3 standard deviations or 3-sigma (line 3). Beyond this value, there was a noticeable increase in the false rates, as shown in the Tables V and VI. The upper threshold ξ is used to select the amplitude values representing the attack intensity in terms of dominant peaks. Determining ξ is limited to 6 standard deviations or 6-sigma (line 10).

Note that threshold computation is sensitive only to the characteristics of the dataset, and independent of our detection scheme; this indicates that our technique can be used on other datasets with negligible changes.

Algorithm 1: Lower and Upper Thresholds Calculations.

Input : Incoming traffic spectrum C, maximum bin frequency f_s, flooding attack confirmation threshold Θ, Jaccard similarity J, fast-entropy rate δ_Z
Output: lower control threshold β, upper control threshold ξ

1 Initialise variables opt_Ac, opt_z_β, opt_z_ξ, opt_β, opt_ξ
2 Calculate mean μ_c & standard deviation σ_c of C
3 Calculate freeman's index v: $v = |1 - f_s|/|C|$ `// using Eq. 7`
4 Initialise $z_\beta = 1$ `// index for calculating` β
5 **while** $z_\beta \leq 3$ **do**
6 Calculate $\beta = [\mu_C + z_\beta * \sigma_C] + v$
7 Calculate Z (sub-array of C)
8 **if** $\delta_Z < \Theta$ & $0 \geq J < 0.5$ **then**
9 Set $z_\xi = z_\beta + 1$ `// index for calculating` ξ
10 **while** $z_\xi \leq 6$ **do**
11 Calculate $\xi = [\mu_C + z_\xi * \sigma_C] + v$
12 Determine attack volume A_{vol} using Eq. 5 and accuracy Ac using Eq. 12.
13 **if** $Ac > Optimal\_Ac$ **then**
14 Set $opt\_Ac = Ac$
15 Set $opt\_z_\beta = z_\beta$
16 Set $opt\_z_\xi = z_\xi$
17 Set $opt\_\beta = \beta$
18 Set $opt\_\xi = \xi$
19 $z_\xi = z_\xi + 1$
20 $z_\beta = z_\beta + 1$
21 return opt_β, opt_ξ

D. Complexity Analysis

Let n be the total number of packets within an attack period t_w. Our detection approach monitors k samples ($k < n$) in time-interval t_s. In other words, we obtained the detection result within t_s for k samples. Let the runtime of computing these samples be defined as α, i.e, $\alpha = t_s \times k$. The spectrum calculation is based on the QuickDFT, which has a time complexity of $n\log n$ [5], but in our case (as mentioned above), the time complexity is O(αlog α). The spectrum analysis includes the time complexity of Jaccard similarity method $O(\alpha)$, modified fast-entropy method ($O(\alpha$ log$\alpha)$), and Euclidean distance calculation $O(\alpha)$. Now by combining all these time complexities we get: $O(\alpha) + O(\alpha) + O(\alpha\log\ \alpha) + O(\alpha\ \log\alpha) = 2(O(\alpha)) + 2(O(\alpha\log\ \alpha)) = O(\alpha\ \log\alpha)$. This time complexity indicates that our technique is an efficient solution for detecting multiscale flooding attacks in resource-constrained ICPS.

E. Performance Evaluation

To assess the performance of the proposed detection technique, we have used four performance evaluation criteria: false positive rate (FPR), true positive rate (TPR), accuracy (Ac), and precision (P). FPR is the percentage of normal instances that are incorrectly classified as an attack and is calculated by the following:

$$\text{FPR} = \frac{\text{FP}}{\text{TN} + \text{FP}}. \tag{10}$$

TPR is a ratio of appropriately identified attack instances to the total number of attack instances in the dataset. It is also called sensitivity and is computed by using the following:

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{11}$$

Accuracy (Ac) is a metric used to describe how correctly the technique detects attacks, and it is computed by the following:

$$\text{Ac} = \frac{\text{TP+TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{12}$$

TABLE VIII
COMPARISON WITH THE EXISTING WORKS

Reference	Domain Analysis	Vol[1]	A.Ac[2] %	Dataset	Detect[2] time $'sec'$	Memory usage	CPU usage
[4]	Time	No	100	No	9.7	NG	NG
[5]	Freq	No	NG	BOUN DDoS 2020	576	NG	NG
[3]	Time	No	NG	No	28	NG	NG
[7]	Time	No	98	CICDDoS 2019	NG	NG	NG
Our	Freq	Yes	100, 100	CICDDoS 2019, BOUN DDoS 2020	0.6, 0.25	3MB, <1KB	10%, 4%

[1] Attack Volume, [2] Average Accuracy, [3]Detection time

Precision is a measure of how well a system can detect attacks or normal behavior. It is calculated by using the following:

$$P = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{13}$$

In the abovementioned performance measures, TP means true positive (result indicating correct identification of flooding attack traffic), where FP means false positive (result showing incorrect identification of normal traffic as an attack traffic), where TN means true negative (correct identification of normal traffic), where FN means false negative (incorrect identification of attack traffic as normal traffic).

The results illustrated in Table VII show that the proposed technique has 100% results for the BOUN DDoS and CICDDoS datasets during different attack scenarios.

F. Comparison With Existing Methods

A general comparison of the proposed technique with the existing literature was conducted in Section II. Since many approaches exist in the literature, conducting a fair comparison with all available works is difficult. Instead, the results of the proposed work are compared with the approaches from [3], [4], [5] and[7], as shown in Table VIII.

Overall, the results demonstrated that our approach exhibits robust performance, maintaining optimal accuracy under multiscale flooding attacks without requiring complex or expensive implementations compared to current works. We have implemented our solution on PLCs that shows reduced performance overheads (memory and CPU usage) and attack detection times compared to existing approaches. The results also demonstrate that multimethod attack detection proves better than single-method detection. For example, Jaccard similarity alone was unreliable for detecting attacks, but when combined with the modified fast-entropy method, we could achieve 100% accuracy. Lastly, unlike existing approaches, considering attack volumes adds a valuable dimension to better understanding of attack intensity and enabling precise and timely actions against potential high-rate attacks.

G. Limitation of the Proposed Technique

The primary purpose of the proposed technique is to detect flooding attacks where incoming traffic exhibits predictable or unpredictable high-rate traffic. These attacks include various attack densities and generate substantial volume of traffic. Selecting appropriate dynamic threshold values plays a vital role in accurate attack detection. The performance of the proposed work can deteriorate for low-rate attacks based on the selected thresholds when the background baseline traffic is similar to the attack traffic.

Also, an imbalanced dataset with an unequal distribution of benign and traffic samples could impact the detection's decision. In the future, we will address this factor to optimise our proposed approach further.

VI. CONCLUSION

Resource-constrained ICPS needed light-weight mechanisms to actively (dynamically and programmability) detect the multiscale flooding attacks and attack volumes early in the incoming traffic flow. In this study, we used statistically robust and low computational overhead methods to detect multiscale flooding attacks in the resource-constrained ICPS. We had experimented our work on publicly available datasets. Furthermore, identifying attack volumes to determine the attack intensity introduced a significant aspect integral to our proposed approach. The implementation of our proposed technique on PLC-based ICPS and the experimental results indicated that our proposed technique was light-weight and surpasses current methods' performances by having an attack detection time of less than 1 s.

Considering the limitations of our work, future work is devoted to developing a fast and efficient online learning model to detect and mitigate the slow-rate DDoS attacks in resource-constrained ICPS.

REFERENCES

[1] X. Ma, L. Almutairi, A. M. Alwakeel, and M. H. Alhameed, "Cyber physical system for distributed network using DoS based Hierarchical Bayesian network," *J. Grid Comput.*, vol. 21, no. 2, 2023, Art. no. 27.
[2] S. Sivamohan, S. Sridhar, and S. Krishnaveni, "TEA-EKHO-IDS: An intrusion detection system for industrial CPS with trustworthy explainable AI and enhanced krill herd optimization," *Peer-to-Peer Netw. Appl.*, vol. 16, no. 4, pp. 1–29, 2023.
[3] B. Liu, J. Chen, and Y. Hu, "Mode division-based anomaly detection against integrity and availability attacks in industrial cyber-physical systems," *Comput. Ind.*, vol. 137, 2022, Art. no. 103609.
[4] L. D. Tsobdjou, S. Pierre, and A. Quintero, "An online entropy-based DDoS flooding attack detection system with dynamic threshold," *IEEE Trans. Netw. Service Manag.*, vol. 19, no. 2, pp. 79–89, Jun. 2022.
[5] F. Zahid, M. M. Kuo, and R. Sinha, "Light-weight active security for detecting DDoS attacks in containerised ICPS," in *Proc. IEEE 18th Int. Conf. Privacy, Secur. Trust*, 2021, pp. 1–5.
[6] B. Li, Y. Wu, J. Song, R. Lu, T. Li, and L. Zhao, "DeepFed: Federated deep learning for intrusion detection in industrial cyber-physical systems," *IEEE Trans. Ind. Inform.*, vol. 17, no. 8, pp. 15–24, Aug. 2021.
[7] B. H. Ali, N. Sulaiman, S. A. R. Al-Haddad, R. Atan, S. L. M. Hassan, and M. Alghrairi, "Identification of distributed denial of services anomalies by using combination of entropy and sequential probabilities ratio test methods," *Sensors*, vol. 21, no. 19, 2021, Art. no. 6453.
[8] J. David and C. Thomas, "Detection of distributed denial of service attacks based on information theoretic approach in time series models," *J. Inf. Secur. Appl.*, vol. 55, 2020, Art. no. 102621.
[9] M. Kordestani, A. Chaibakhsh, and M. Saif, "SMS–A security management system for steam turbines using a multisensor array," *IEEE Syst. J.*, vol. 14, no. 3, pp. 13–24, Sep. 2020.
[10] M. S. Mahmoud, M. M. Hamdan, and U. A. Baroudi, "Secure control of cyber physical systems subject to stochastic distributed DoS and deception attacks," *Int. J. Syst. Sci.*, vol. 51, no. 9, pp. 53–68, 2020.
[11] M. Raiyat Aliabadi, M. Seltzer, M. Vahidi-Asl, and R. Ghavamizadeh, "ARTINALI#: An efficient intrusion detection technique for resource-constrained cyber-physical systems," *Int. J. Crit. Infrastructure Protection*, vol. 33, no. 1, 2021, Art. no. 100430.
[12] J. David and C. Thomas, "Efficient DDoS flood attack detection using dynamic thresholding on flow-based network traffic," *Comput. Secur.*, vol. 82, pp. 284–295, 2019.

[13] R. Fouladi, O. Ermis, and E. Anarim, "Anomaly-based DDoS attack detection by using sparse coding and frequency domain," in *Proc. IEEE 30th Annu. Int. Symp. Pers., Indoor Mobile Radio Commun.*, 2019, pp. 1–6.
[14] S. Behal, K. Kumar, and M. Sachdeva, "D-FACE: An anomaly based distributed approach for early detection of DDoS attacks and flash events," *J. Netw. Comput. Appl.*, vol. 111, pp. 49–63, 2018.
[15] K. Singh, K. S. Dhindsa, and B. Bhushan, "Threshold-based distributed DDoS attack detection in ISP networks," *Turkish J. Elect. Eng. Comput. Sci.*, vol. 26, no. 4, pp. 1796–1811, 2018.
[16] N. Agrawal and R. Kumar, "Security perspective analysis of industrial cyber physical systems (I-CPS): A decade-wide survey," *ISA Trans.*, vol. 130, pp. 10–24, 2022.
[17] D. Zhang, Q.-G. Wang, G. Feng, Y. Shi, and A. V. Vasilakos, "A survey on attack detection, estimation and control of industrial cyber–physical systems," *ISA Trans.*, vol. 116, pp. 1–16, 2021.
[18] P. Verma, M. P. De Leon, J. G. Breslin, and D. O'Shea, "FedTIU: Securing virtualized PLCs against DDoS attacks using a federated learning enabled threat intelligence unit," in *Proc. IEEE Int. Conf. Smart Comput.*, 2023, pp. 233–236.
[19] S. Bhatia, S. Behal, and I. Ahmed, "Distributed denial of service attacks and defense mechanisms: Current landscape and future directions," *Versatile Cybersecurity*, vol. 72, pp. 55–97, 2018.
[20] F. Zahid, G. Funchal, V. Melo, M. M. Y. Kuo, P. Leitao, and R. Sinha, "DDoS attacks on smart manufacturing systems: A cross-domain taxonomy and attack vectors," in *Proc. IEEE 20th Int. Conf. Ind. Inform.*, 2022, pp. 14–19.
[21] Y. Omer, "DDoS attack trends for Q1 2023," Accessed: Aug. 25, 2023. [Online]. Available: https://radar.cloudflare.com/reports/ddos-2023-q1
[22] X. He, C. Papadopoulos, J. Heidemann, U. Mitra, U. Riaz, and A. Hussain, "Spectral analysis of bottleneck traffic," Dept. Comput. Sci., Univ. Southern California, Los Angeles, CA, USA, Tech. Rep. USC-CSDTR-05-854, 2005.
[23] S. Tan, J. M. Guerrero, P. Xie, R. Han, and J. C. Vasquez, "Brief survey on attack detection methods for cyber-physical systems," *IEEE Syst. J.*, vol. 14, no. 4, pp. 5329–5339, Dec. 2020.
[24] M. Nooribakhsh and M. Mollamotalebi, "A review on statistical approaches for anomaly detection in DDoS attacks," *Inf. Secur. J.: A Glob. Perspective*, vol. 29, no. 3, pp. 118–133, 2020.
[25] D. Erhan and E. Anarim, "Boğaziçi university distributed denial of service dataset," *Data Brief*, vol. 32, 2020, Art. no. 106187.
[26] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy," in *Proc. IEEE Int. Carnahan Conf. Secur. Technol.*, 2019, pp. 1–8.
[27] S.-H. Cha, "Comprehensive survey on distance/similarity measures between probability density functions," *City*, vol. 1, no. 2, pp. 300–307, 2007.
[28] J. David and C. Thomas, "DDoS attack detection using fast entropy approach on flow-based network traffic," *Procedia Comput. Sci.*, vol. 50, pp. 30–36, 2015.
[29] M. Cover Thomas and A. Thomas Joy, *Elements of Information Theory*. New York, NY, USA: Wiley, 1991, vol. 3, pp. 37–38.
[30] B. G. Amidan, T. A. Ferryman, and S. K. Cooley, "Data outlier detection using the Chebyshev theorem," in *Proc. IEEE Aerosp. Conf.*, 2005, pp. 3814–3819.

Farzana Zahid received the B.S. (Hons.) degree in computer science and the M.S. degree in system and software engineering from Mohammed Ali Jinnah University, Karachi, Pakistan, and the Ph.D. degree in cybersecurity from Auckland University of Technology, Auckland, New Zealand.

She is currently a Lecturer with the Department of Computer Science, University of Waikato, Hamilton, New Zealand. Her research interests include active security, industrial cyber-physical systems, software engineering, and machine learning. Her research is dedicated to enhancing the self-protection of resource-constrained devices by applying finely tuned statistical and machine-learning methodologies.

Matthew M.Y. Kuo (Member, IEEE) received the B.E. (Hons.) and Ph.D. degrees in electrical and computer systems engineering from the University of Auckland, Auckland, New Zealand, in 2008 and 2015, respectively.

He is currently a Senior Lecturer with the School of Engineering, Computer and Mathematical Sciences, Auckland University of Technology, Auckland, New Zealand. His current research interests include cyber-physical embedded systems, Internet-of-Things, robotics, precision-timed systems, industrial automation, and safety-critical systems.

Roopak Sinha (Senior Member) received the master's degree in commercialisation and entrepreneurship, and the Ph.D. degree in electrical and electronics engineering from the University of Auckland, Auckland, New Zealand, in 2009 and 2016, respectively.

He is a Professor of software engineering with the School of Information Technology, Deakin University, Geelong, Australia. He has previously held research positions with Auckland University of Technology and The University of Auckland, New Zealand, and INRIA, France. He is an internationally recognized expert in systematically designing safe, secure, and maintainable industrial software, with interests in requirements engineering, design and architecture, code generation, formal methods, research commercialization, and industrial standards.

Gustavo Funchal received the B.Eng. degree in control and automation engineering from the Federal University of Technology, Curitiba, Brazil, in 2020, and the M.Eng. degree in industrial engineering from Polytechnic Institute of Bragança (IPB), Bragança, Portugal, in 2020. He is currently working toward the Ph.D. degree in computer engineering with the University of Salamanca, Salamanca, Spain.

His main research interests include artificial intelligence with focus on data analysis, Internet of Things (IoT), cybersecurity, cyber-physical systems, and multiagent systems.

Tiago Pedrosa received the bachelor's degree in informatics engineering from the Polytechnic Institute of Bragança, Bragança, Portugal, in 2006, and the Ph.D. degree in computer science from the Universities of Minho, Aveiro, and Porto, Portugal, in 2013.

He is an Adjunct Professor, Researcher, and Consultant in the field of cybersecurity and information security with the Informatics and Communications Department, School of Technology and Management of the Polytechnic Institute of Bragança, Bragança Portugal. His research interests include applied computer security, integrates projects, technology transfer, and consultancy.

Paulo Leitao (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Porto, Porto, Portugal, in 2004.

He is currently a Full Professor with the Department of Electrical Engineering, Polytechnic Institute of Bragança, Bragança, Portugal, and the Coordinator of the Research Centre in Digitalization and Intelligent Robotics (CeDRI). His research interests include intelligent and reconfigurable systems, industrial cyber-physical systems, multiagent systems, digital twin, and factory automation.

Dr. Leitao is a Senior Member of the IEEE Industrial Electronics Society (IES) and Systems, Man and Cybernetics Society (SMCS), past Chair of the IEEE IES Technical Committee on Industrial Agents, and the Chair of the established IEEE 2660.1-2020 Standard.