Jordanian Journal of Computers and Information Technology

March 2021 | VOLUME 07 | NUMBER 01 | ISSN 2415 - 1076 (Online) ISSN 2413 - 9351 (Print)

JJCIT

www.jjcit.org jjcit@psut.edu.jo

An International Peer-Reviewed Scientific Journal
Financed by the Scientific Research Support Fund

Jordanian Journal of Computers and Information Technology (JJCIT)

The Jordanian Journal of Computers and Information Technology (JJCIT) is an international journal that publishes original, high-quality and cutting edge research papers on all aspects and technologies in ICT fields.

JJCIT is hosted by Princess Sumaya University for Technology (PSUT) and supported by the Scientific Research Support Fund in Jordan.

AIMS AND SCOPE

The JJCIT aims to publish the most current developments in the form of original articles as well as review articles in all areas of Telecommunications, Computer Engineering and Information Technology and make them available to researchers worldwide. The JJCIT focuses on topics including, but not limited to: Computer Engineering & Communication Networks, Computer Science & Information Systems and Information Technology and Applications.

INDEXING

JJCIT is indexed in:

EDITORIAL BOARD SUPPORT TEAM

JJCIT ADDRESS

WEBSITE: www.jjcit.org
EMAIL: jjcit@psut.edu.jo
ADDRESS: Princess Sumaya University for Technology, Khalil Saket Street, Al-Jubaiha
B.O. BOX: 1438 Amman 11941 Jordan
TELEPHONE: +962-6-5359949
FAX: +962-6-7295534

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 07, No. 01, March 2021.

5G HAIRPIN BANDPASS FILTER

Sahar Saleh[1], Widad Ismail[1], Intan Sorfina Zainal Abidin [1], Mohd Haizal Jamaluddin[2], Mohammed H. Bataineh[3] and Asem S. Alzoubi[3]

(Received: 15-Aug.-2020, Revised: 28-Sep.-2020, Accepted: 13-Oct.-2020)

ABSTRACT

In this paper, Hairpin Bandpass Filter (HPBF) is designed, simulated and fabricated at two 5G low-frequency bands: 3.7 GHz - 4.2 GHz and 5.975 GHz -7.125 GHz. This filter will be a part of our 5G narrowband/ Ultra Wide Band (UWB) reconfigurable antenna project that plays a significant role in the recent wireless networks, such as Cognitive Radios (CRs). Through the two frequency bands, the filter resulted in good matching and transmission responses with enhanced bandwidth. The measured reflection coefficient of the proposed HBPF, S_{11} is < -10 dB and < -11.66 dB through 3.45 GHz – 4.25 GHz and 5.62 GHz – 7.6 GHz, respectively. However, the transmission coefficient, S_{12} is around -1.5 dB and – 1.17 dB at the center frequencies F_C = 3.75 GHz and 6.61 GHz, respectively. In this paper, the High-Frequency Structure Simulator (HFSS) software is used to carry out the simulation. The full-wave simulation results are validated with the hardware measurements.

KEYWORDS

Hairpin bandpass filter (HPBF), Harmonics suppression, 5G, Reconfigurable antenna, Cognitive radios (CRs), HFSS.

1. INTRODUCTION

Filters play an important role in many RF/Microwave applications and are used to control the frequency responses (band-pass, band-stop, low-pass and high-pass) to overcome the limitation of the electromagnetic spectrum and facilitate the possibility to share it. HPBF is a compact structure bandpass filter which is simply constructed by folding the λ/2 resonators of the parallel-coupled filter, to get the U shape that eases its fabrication process, where no grounding *via* holes is needed [1]. HPBF has been recently used in many applications at different frequencies, such as Ku-band satellite communication [2]-[3], WiMAX [4]-[5], Wireless Local Area Network (WLAN) [6] and millimeter-wave applications [7]-[8]. In [9], X- band 9.3 GHz HPBF with a bandwidth (BW) of 0.28 GHz was designed for radar navigation; however, a 2 GHz – 4 GHz HPBF was designed in [10] for satellite application. For the 5th generation mobile communication system, authors in [11] designed three different 20 GHz HPBFs with different feeding techniques. It was found that the filter with input/output feed structure is better than the one with tap-type feed in terms of insertion and return losses. Authors in [12] designed HPBF with sharp frequency response suitable for narrowband communication, such as the uplink frequency in the band -3 eNodeB LTE (1.710 GHz-1.785 GHz). A 6-order unlicensed frequency band (2.2 GHz to 2.3 GHz) HPBF that meets electromagnetic interference (EMI) or electromagnetic compatibility (EMC) issues requirements was designed in [13]. In [14], the authors designed HPBF for 923 MHz RFID application. Second harmonics suppression for 1 GHz HPBF was achieved using modified Minkowski fractal shape in [15]. Besides, Defected Ground and Microstrip Structures (DGS) and (DMS) were used in HPBF for performance enhancement and size reduction. Superior harmonics suppression in the response of HPBF is obtained in [16] by adding different DMSs to the filter's resonators. The resulted suppressions were -25 dB and -40 dB for the second and the third harmonics, respectively. In [17], dumbbell-shaped DGS cells are etched at the input and output ports of a five-pole 2.5 GHz HPBF to enhance high-order harmonics suppressions. Extra harmonics suppression was also achieved by introducing an open stub at the feedline of the designed filter. Authors in [18] also used two pairs of dumbbell-shaped DGS cells at the feed lines of a 3-GHz HPBF used for microwave imaging application to enhance the BW in addition to get better S_{11}. In addition, to enhance the filter response, size reduction can also be achieved by adding square DGSs to HPBFs used for S-band [19]-[20] and X-band radar

1. S. Saleh, W. Ismail and I. S. Z. Abidin are with School of Electrical and Electronic Engineering, Universiti Sains Malaysia, Penang, Malaysia. Emails: sahar_saleh@student.usm.my, eewidad@usm.my and intan.sorfina@usm.my

2. M. H. Jamaliddin is with Wireless Communication Centre, Universiti Teknologi Malaysia, Johor, Malaysia. Email: haizal@utm.my

3. M. Bataineh and A. Alzoubi are with Hijjwai Faculty for Engineering Technology, Yarmouk University, Irbid, Jordan. Emails: mohbat@yu.edu.jo and asem@yu.edu.jo

applications [21]. However, in [22], dumbbell and stepped hairpin resonator (SHPR) DGS shapes are used for second harmonics suppression below -30 dB for a 2.4-GHz HPBF with 29% FBW. The design in [19] was enhanced using the dumbbell DGS [23]. Tapped feed lines are also used for harmonics suppression as in [24] by controlling the two transmission zeros. The designed 8.75-GHz HPBF provided 1 to 8 GHz and 9 to 15 GHz harmonics suppression. Furthermore, Plackett-Burman Design of Experiment methodology (DOE) was applied in designing a 2.4-GHz HPBF for further optimization. Resulted insertion and return loss of this filter were improved by 61% and 15 % from the designed one with Gensys software [25]. Many techniques were developed to reduce the size of HPBF, such as adding ground holes, a high dielectric substrate, multilayer structure, nonuniform coupled line resonators, aperture coupled line resonators, metamaterial complementary split ring resonators, Inkjet Printing and integrated passive device (IPD) technologies. A 37 % of size reduction was achieved in designing a 923-MHz HPBF with 7.5 MHz BW *via* hole grounding [14]. However, an 11% size reduction with spurious harmonics suppressions was obtained for a 1-GHz HPBF in [26] using novel fractal shaped lines (FSLs). For low-cost and high-performance integrated circuits, Inkjet Printing Technology (IPT) is used in [7] to design a 30-GHz millimeter wave HPBF on Liquid Crystal Polymer (LCP) substrate. Conventional Uniform Transmission Lines (UTLs) of 2.06 GHz HPBF resonators were replaced by artificial left-handed and right-handed transmission lines (LHRHTLs) in [27] for size reduction. For further compactness, these artificial lines were implemented in the form of multilayered structures by using liquid crystal polymer technology (LCPT). Compact size with spurious harmonic response suppression was obtained by using nonuniform coupled lines (NCLs) resonators to design a 34-GHz HPBF in [8]. A five-pole ISM band HPBF with approximately 40 % size reduction was designed in [28] using two stacked microstrip layers. In this filter, two apertures are introduced in the common ground planes to provide coupling between resonators at the two layers. A high dielectric substrate (Al_2O_3 ceramic) was used to design a compact C-band HPBF in [29]. A compact HPBF with a selective 100 MHz WLAN BW was proposed in [6] using two short-circuited comb-lines coupled with a Rectangular Shaped Loop Resonator (RSLR). For the first time, integrated passive device (IPD) technology was used in [30] to design a compact W-band 95.5-GHz HPBF with 8 GHz BW. A GaAs substrate with a thickness of 0.100 mm and two metal layers were employed to design the filter using PID technology. In [5], a compact multi-band (3.5 GHz and 5.5 GHz) HPBF was proposed using metamaterial complementary split ring resonators. Multilayer techniques are used in [2] for significant size reduction and enhanced Ku-band HPBF performance. An HPBF with a tunable center frequency of 650 MHz to 920MHz and a BW of 25MHz and 85 MHz was designed in [31] using Screen Printed Ferroelectric Varactors. However, in [32], a barium strontium titanate (BST) thin film was used for continuous control on center frequency (from 900MHz to 1.1 GHz) and BW (from 40MHz to 80MHz) of the designed HPBF. A $\lambda/4$ stub resonator was added to the filter to increase the stop attenuation and band isolation by controlling the introduced transmission zero at the upper frequency.

5G is a revolutionary technology that aims to enhance the communication link data rate [33]-[34], reduce the latency and increase the reliability [35]-[37] to meet the growing demand of mobile users and to be suitable for (Internet of Things) IoT networking services [38]. On October 24, 2018, the Federal Communication Commission (FCC) has proposed two frequency bands below 24 GHz: licensed (C-Band: 3.7 GHz-4.2 GHz) and unlicensed (5.925 GHz -6.425 GHz, 6.525 GHz -6.875 GHz and 6.875 GHz – 7.125 GHz bands (totally 5.925 GHz–7.125 GHz)) spectrum for 5G technology [39]. In this work, as a contribution to 5G technology, an HPBF is designed at these two frequency bands a wider resulting BW as compared to other filters in the literature. For recent wireless communication system

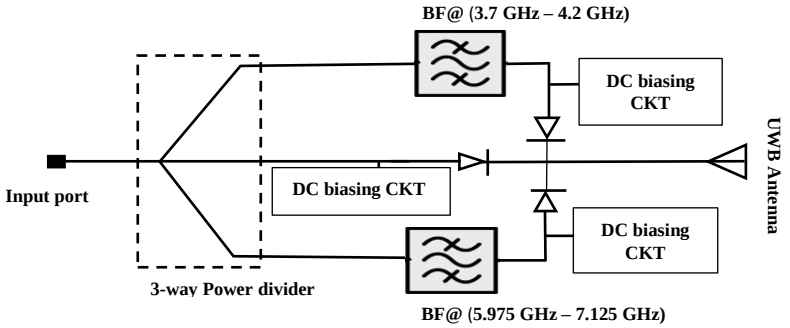
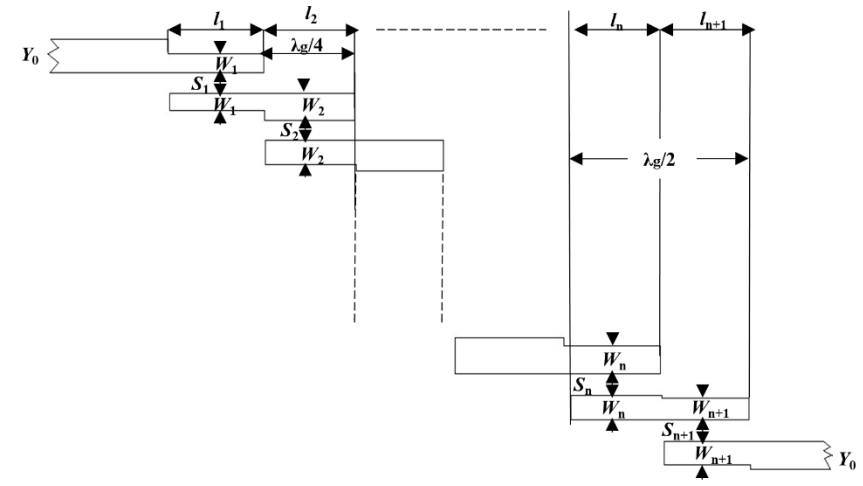


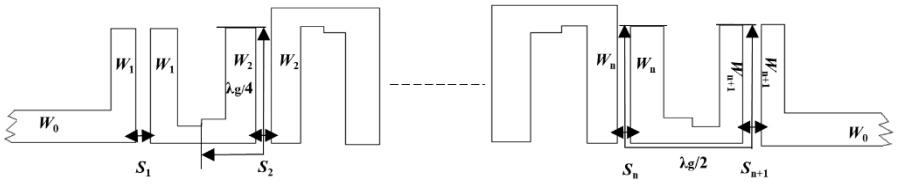
Figure 1. Future project: 5G narrowband / UWB reconfigurable antenna.

2. HAIRPIN BANDPASS FILTER

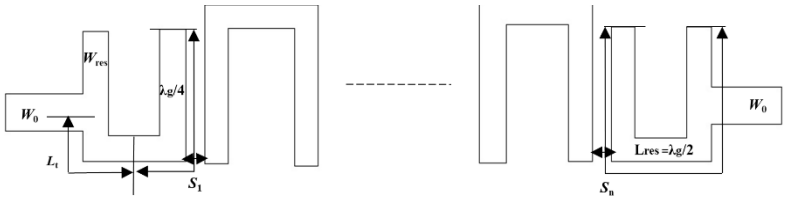
To reduce the size of the parallel-coupled λ/2 resonator filter, its resonators as shown in Figure 2a can be folded to get the U shape as in Figure 2b, forming a new type of bandpass filter called Hairpin Bandpass Filter (HPBF). In addition to HPBF compact structure, it is preferable due to its ease of fabrication, where no grounding *via* holes is needed [1].



(a)



(b)



(c)

Figure 2. N-order (a) Parallel-coupled λ/2 resonator filter into (b) and (c) HPBF.

The characteristics impedance of each resonator of n[th]-order HPBF can be found in [1]:

$$(Z_0)_{j,j+1} = \sqrt{(Z_{0e}^2)_{j,j+1} + (Z_{0o}^2)_{j,j+1}} \tag{1}$$

where

$$(Z_{0e})_{j,j+1} = \frac{1}{Y_0}\left[1 + \frac{J_{j,j+1}}{Y_0} + \left(\frac{J_{j,j+1}}{Y_0}\right)^2\right], \quad \text{for j = 0 to n} \tag{1.1}$$

$$(Z_{0O})_{j,j+1} = \frac{1}{Y_0}\left[1 - \frac{J_{j,j+1}}{Y_0} + \left(\frac{J_{j,j+1}}{Y_0}\right)^2\right], \quad \text{for j = 0 to n} \tag{1.2}$$

$$\frac{J_{01}}{Y_0} = \sqrt{\frac{\pi FBW}{2g_0g_1}}, \tag{1.3}$$

$$\frac{J_{j,j+1}}{Y_0} = \frac{\pi FBW}{2\sqrt{g_j g_{j+1}}} \quad \text{, for j = 1 to n-1} \tag{1.4}$$

and

$$\frac{J_{n,n+1}}{Y_0} = \sqrt{\frac{\pi FBW}{2g_n g_{n+1}}} \tag{1.5}$$

where Y_0 and $J_{j,j+1}$ are the characteristic admittances of terminating lines (input and output ports) and J inverters, respectively. g_0, g_1 . . . g_n are the elements of a ladder-type lowpass prototype with a normalized cutoff, $\Omega_C = 1$ and FBW is the fractional BW of bandpass filter which is equal to:

$$FBW = \frac{F_C}{BW} \tag{1.6}$$

After calculating the characteristic impedance of each resonator in (1), its width can be calculated using an online microstrip line calculator [41] based on the following familiar microstrip line Equation [42]:

$$W = \begin{cases} \frac{8he^A}{e^{2A}-2} & , \frac{W}{h} < 2 \\ \frac{2}{\pi} h \left\{ B - 1 - ln(2B-1) + \frac{\varepsilon_r - 1}{2\varepsilon_r} \left[ln(B-1) + 0.39 - \frac{0.61}{\varepsilon_r} \right] \right\} & , \frac{W}{h} > 2 \end{cases}, \tag{2}$$

$$A = \frac{Z_o}{60} \sqrt{\frac{\varepsilon_r + 1}{2}} + \frac{\varepsilon_r - 1}{\varepsilon_r + 1} \left(0.23 + \frac{0.11}{\varepsilon_r} \right), \tag{2.1}$$

$$B = \frac{377\pi}{2Z_o\sqrt{\varepsilon_r}}. \tag{2.2}$$

Two important parameters controlling the performance of the filters are the separation between the adjacent resonators, S and the tapped line at the input and at the output, L_t. Details on their effect will be addressed in the upcoming sections. S and L_t can be determined by the mutual coupling coefficient, $M_{i,i+1}$ and the external quality factor of the filter, Q_e, respectively. These design parameters are given as follows [1]:

$$Q_{e1} = \frac{g_0 g_1}{FBW}, \quad Q_{en} = \frac{g_n g_{n+1}}{FBW}, \quad M_{i,i+1} = \frac{FBW}{\sqrt{g_i g_{i+1}}} \quad \text{, for i = 1 to n} \tag{3}$$

where Q_{e1} and Q_{en} are the input and output external quality factors.

It should be mentioned here that in Full-wave Electromagnetic simulations, such as ANSYS HFSS, the relation between S and $M_{i,i+1}$ and between L_t and Q_e can be extracted using [1]:

$$Q_e = \frac{w_0}{\Delta w \pm 90^\circ}, \quad M_{i,i+1} = \pm \frac{f_{p2}^2 - f_{p1}^2}{f_{p2}^2 + f_{p1}^2}, \tag{4}$$

where w_0, $\Delta w \pm 90°$, f_{p2} and f_{p1} are the angular center frequency, absolute BW between $\pm 90°$ points, high- and low-frequency peaks, respectively.

In this study, the g's values of Chebyshev response lowpass prototype with 0.1 dB passband ripple are $g_0 = g_4 = 1$, $g_1 = g_3 = 1.0316$, $g_2 = 1.1474$. In addition, the chosen substrate material is Rogers RO4003C with $\varepsilon r = 3.55$, height h = 0.813 mm and dielectric loss tangent = 0.0027.

3. 3.95-GHZ HAIRPIN BANDPASS FILTER

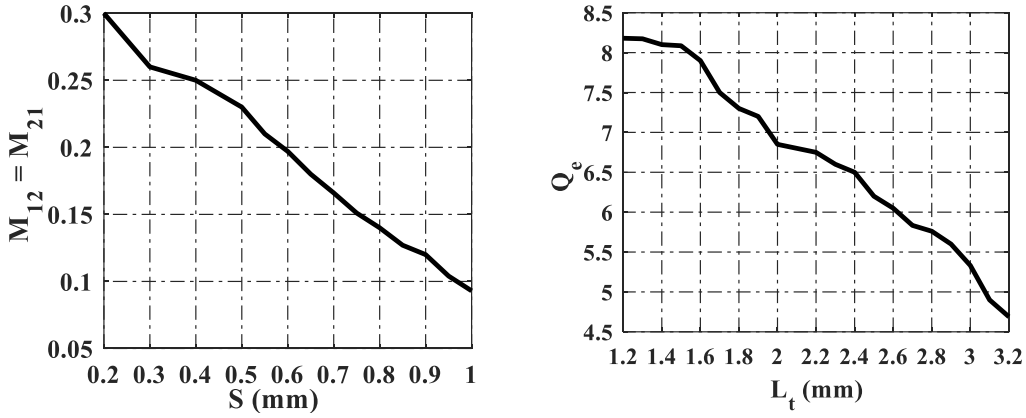
At Fc= 3.95 GHz of the 5G lower band (3.7 GHz- 4.2 GHz) and using the design Equations (1)-(4), a 3.95-GHz HPBF is designed and simulated using ANSYS HFSS. The calculated and optimized parameters are shown in Table 1, where L_{res} , W_{res}, S, L_t, L_{p1}, L_{p2} and W_p are the length of the resonator, the width of the resonator, the space between two adjacent resonators, tapping length, length of the first and second ports and width of the ports, respectively. Figures 3a and 3b show the relation between S and $M_{1,2}$ and between L_t and Q_e, respectively. The detailed parametric studies are shown in Figure 4.

Table 1. Calculated and optimized parameters of 3.95-GHz HPBF.

Parameters	Calculated	Optimized
Q_e	8.14	-
$M_{12}=M_{21}$	0.12	-
L_{res} (mm)	23.268	23
W_{res} (mm)	0.595	0.45
S (mm)	0.9	0.3
L_t (mm)	1.3	2.7
L_{p1} (mm)	-	3

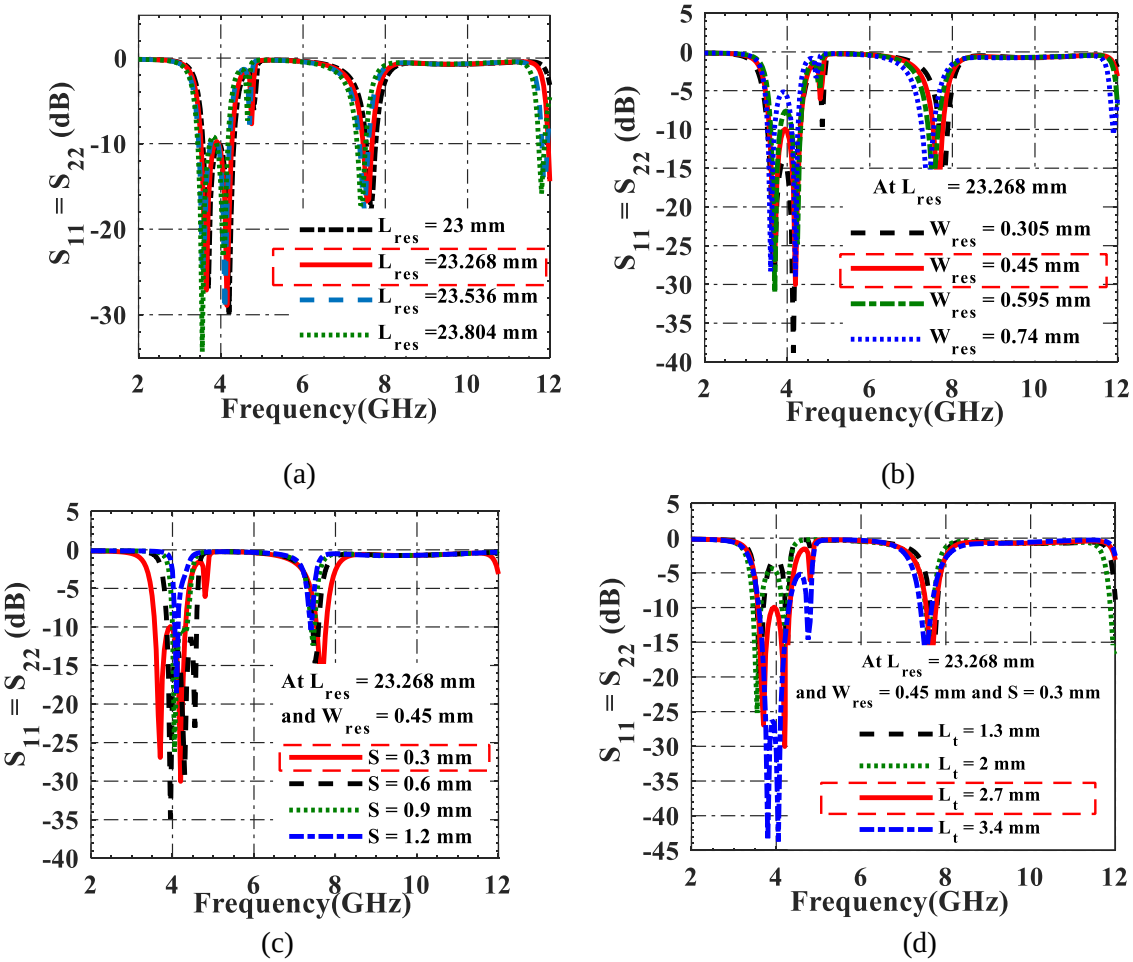
L_{p2} (mm)		3
Wp (mm)	1.819	1.5

The chosen optimized values are in red color withs dashed red box, which indicates the acceptable matching and the smallest ripples within and beyond the required band (3.7 GHz – 4.2 GHz), respectively. As noticed from Figure 4a, the band is shifted to the left or right as L_{res} increased or decreased, respectively. Good matching is obtained when W_{res} = 0.45 mm, as shown in Figure 4b. Figure 4c illustrates that to increase the BW, the coupling between resonators should be increased by decreasing S. Better matching is obtained when L_t is larger than the calculated one, as shown in Figure 4d and finally, good matching within the required band is obtained when the port width is less than 1.819 mm, as indicated in Figure 4e.

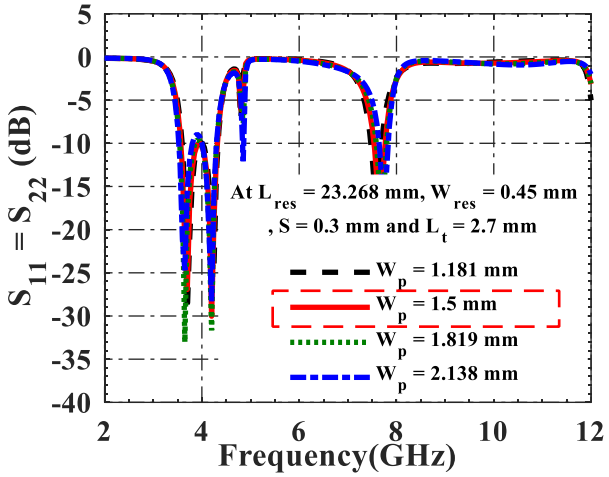


(a) (b)

Figure 3. (a) Variation of mutual coupling due to the space between adjacent resonators and (b) Variation of external factor due to the tapping length of 3.95-GHz HPBF.

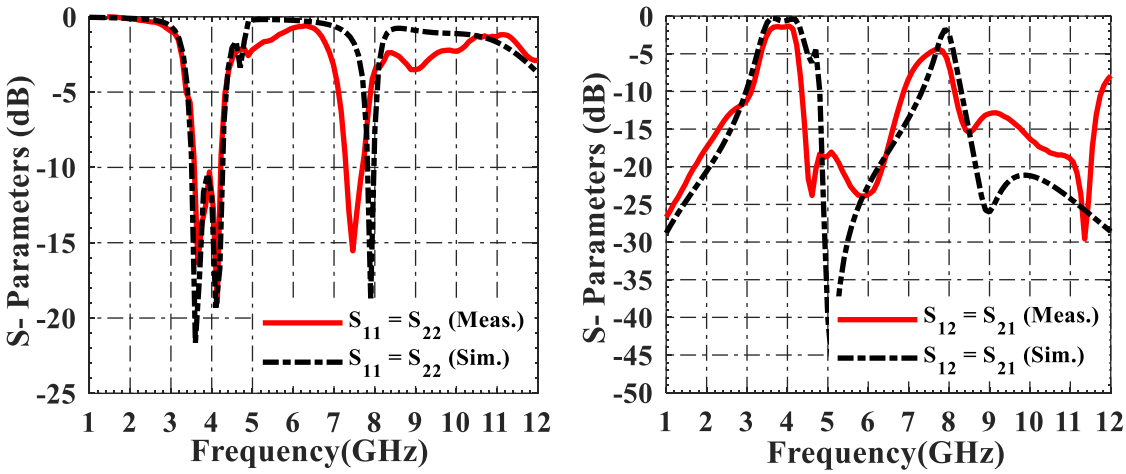


(a) (b) (c) (d)



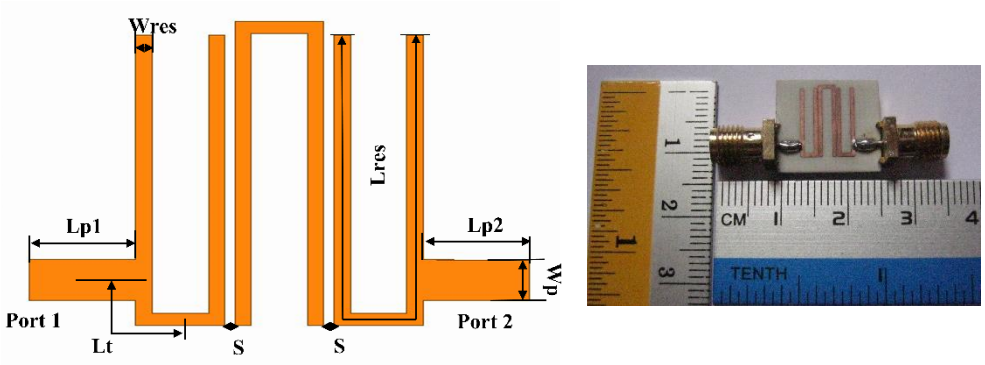
(e)

Figure 4. Parametric studies of the proposed 3.95 GHz HPBF on (a) L_{res}, (b) W_{res}, (c) S, (d) L_t and (e) W_p.



(a) (b)

Figure 5. Simulated and measured (a) Return loss and (b) Insertion loss of the proposed 3.95-GHz HPBF.



(a) (b)

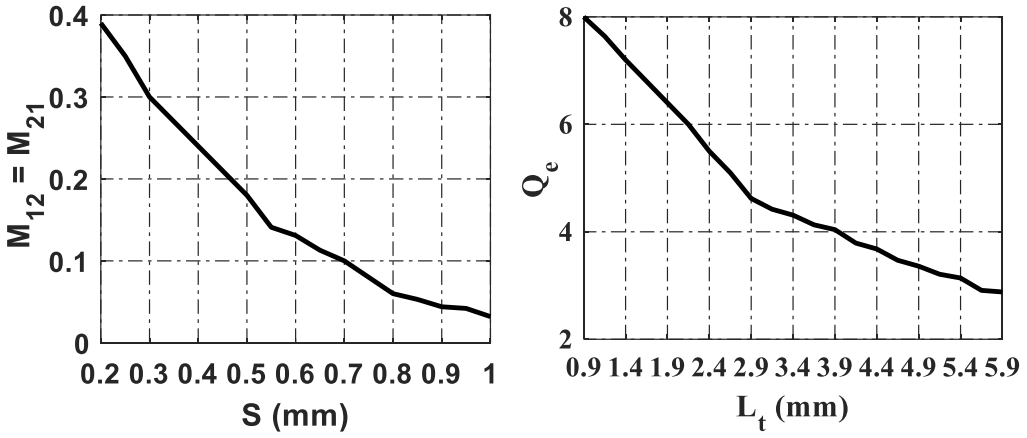
Figure 6. (a) Layout of the proposed 3.95 GHz HPBF and (b) fabricated prototype.

As shown in Figure 5a, good matching and transmitting response are obtained for the designed filter,where the measured and simulated return losses are less than -10 dB and -10.45 dB with enhanced BW of 0.1 GHz and 0.3 GHz through 3.57 GHz – 4.17 GHz and 3.45 GHz – 4.24GHz, respectively. Also, the measured and simulated insertion losses are around -1.5 dB and -0.78 dB at centre frequencies Fc = 3.75 GHz and 3.81 GHz, respectively, as shown in Figure 5b. In addition, one can observe that the

proposed 3.95 GHz HPBF can support up to only 7.8 GHz (Sim.) and 8 GHz (Meas.) harmonics suppression. Because this harmonic will occur within the UWB, this filter is not preferable in reconfigurable 5G narrow band / UWB reconfigurable antenna applications, such as Cognitive Radios (CRs). To overcome this limitation, Defected Ground and Microstrip Structures (DGS) and (DMS) can be added to the filter as in [19]-[20] and [8], respectively. Moreover, another type of compact structure filter can be used, such as interdigital filter that supports high-order harmonics suppression [1]. The layout and the fabricated prototype of the designed filter are shown in Figure 6. The circuit area of this filter is 14.1 mm x 11.2 mm (0.31 λg × 0.24 λg).

4. 6.55-GHz Uniform Transmission Line Hairpin Bandpass Filter

Based on the design equations (1) - (4), 6.55 GHz HPBF is designed. Table 2 indicates all the calculated and optimized parameters of the filter. Figure 7 shows the variation of M_{12} and Q_e due to S and L_t, respectively. The optimized parameters in Table 2 are obtained *via* parametric studies to get better filter matching responses, as indicated in Figure 8. The optimized parameters in Figure 8 are in red solid line eith a dashed red box. Figure 8a indicates that at the calculated L_{res} (14.324 mm), the band is shifted to the right and good matching within the required band is obtained when L_{res} increases a little bit to 15.524 mm. No matching is obtained when W_{res}< 0.5 mm, as shown in Figure 8b and although matching is good at W_{res} = 0.7 mm, the high frequency of the band is covered when W_{res} = 0.6 mm. Figure 8c illustrates that to increase the BW, the coupling between resonators should be increased by decreasing S and the best matching within the required frequency band is obtained at S = 0.3 mm. As shown in Figure 8d, good matching is obtained at L_t =2.9 mm. The proposed HPBF is shown in Figure 9. The dimensions of this filter are 17.3 mm x 6.91 mm (0.66 λg × 0.27 λg).



(a) (b)

Figure 7. (a) Variation of mutual coupling due to the space between adjacent resonators and (b) Variation of an external factor due to the tapping length of 6.55 UTL HPBF.

Table 2. Calculated and optimized parameters for 6.55-GHz HPBF.

Parameters	Calculated	Optimized
Q_e	7.84	-
$M_{12}=M_{21}$	0.162	-
L_{res} (mm)	14.324	15.524
W_{res} (mm)	0.5	0.6
S (mm)	0.65	0.3
L_t (mm)	1.3	2.9
$L_{p1}= L_{p2}$(mm)	-	4
Wp (mm)	1.819	1.819

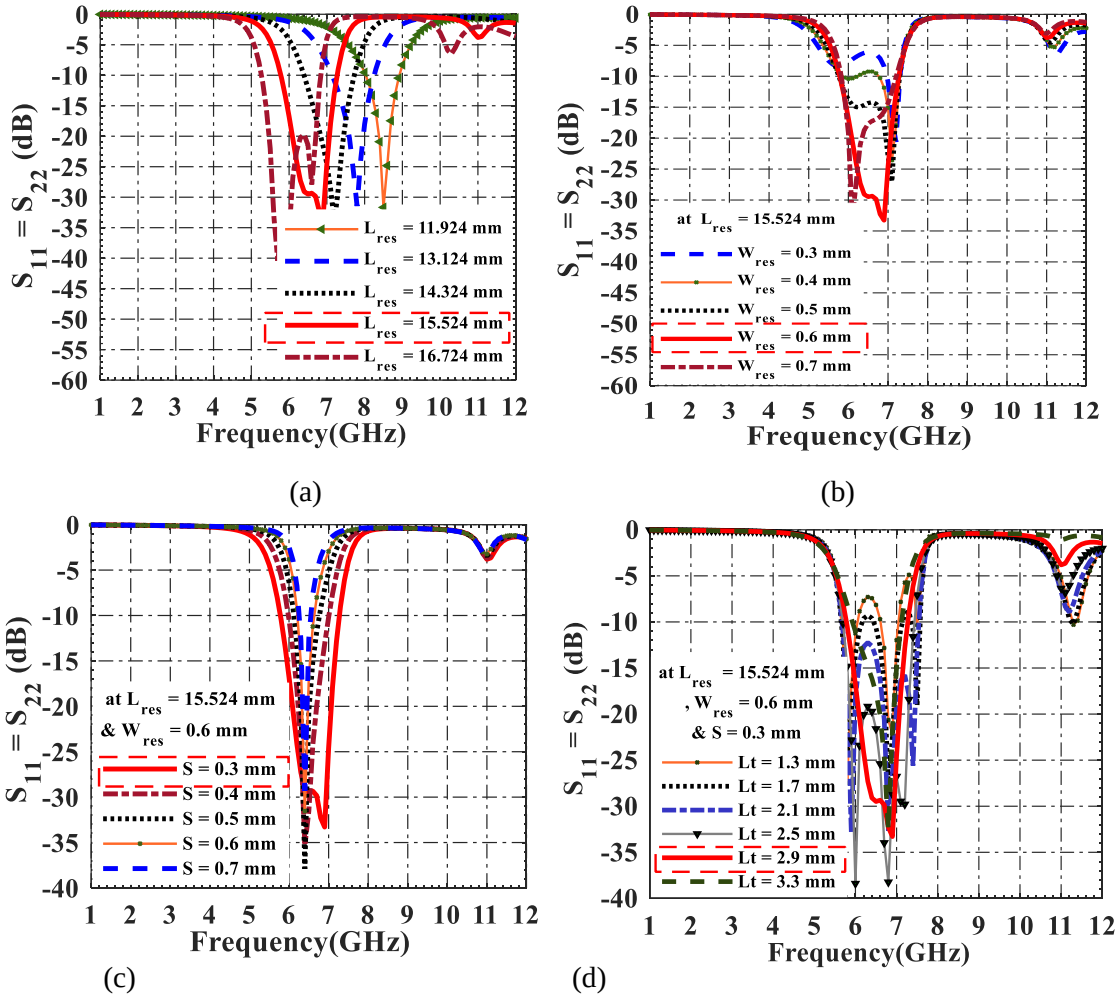


Figure 8. Parametric studies of the proposed 6.55 GHz UTL HPBF on (a) L_{res}, (b) W_{res}, (c) S and (d) L_t.

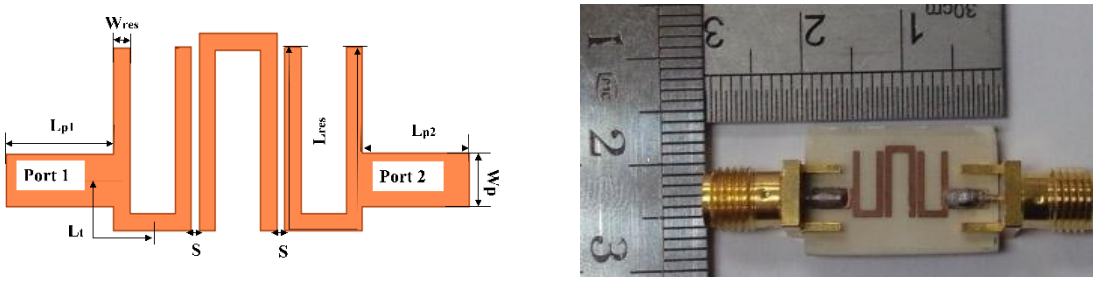
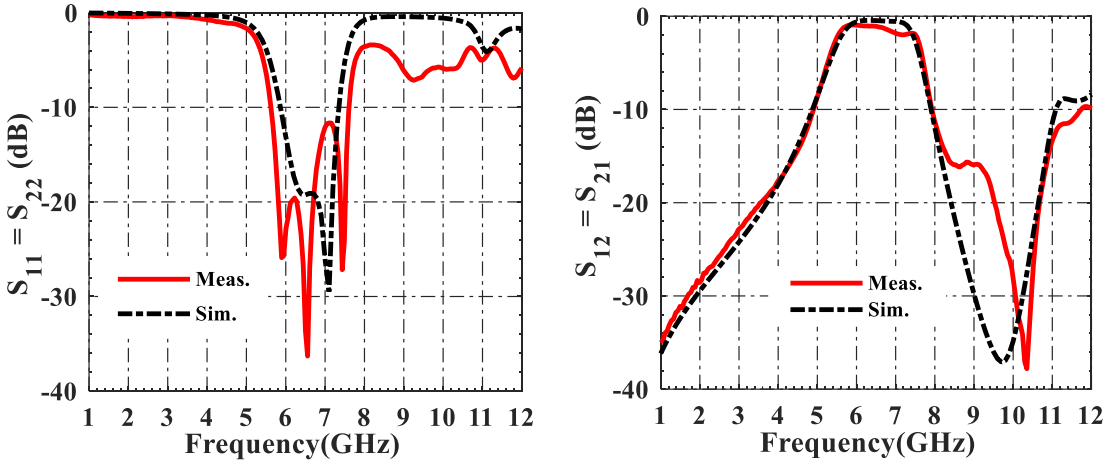


Figure 9. (a) Layout of the proposed 6.55-GHz HPBF and (b) Fabricated prototype.

Figure 10 illustrates the simulated and measured results for the designed 6.55-GHz HBPF with good impedance matching, transmission response and up to 11.1 GHz harmonics suppressions, where $S_{11} = S_{22}$ is < -11.66 dB (Meas.) and < -19 dB (Sim.) with a BW enhancement of 0.83 GHz and 0.33 GHz and up to 11 GHz at frequency ranges (5.62 GHz – 7.6 GHz) and (5.87 GHz – 7.35 GHz), respectively. However, $S_{12} = S_{21}$ is around -1.17 dB (Meas.) and -0.5 dB (Sim.) at center frequency 6.61 GHz. The difference between simulated and measured results is due to the fabrication and measurement tolerances. These good results make the filter suitable to be integrated into our 5G narrowband / UWB reconfigurable antenna project.

Finally, Table 3 shows a comparison of the proposed filters in this paper with other HPBFs in the literature with different frequency ranges. As it is clear from the table in terms of narrowband, the proposed 5G HPBF provides wider BW at the two frequency bands with good impedance matching and transmission response.



(a) (b)

Figure 10. Simulated and measured (a) Return loss and (b) Insertion loss of the proposed 6.55-GHz HPBF.

Table 3. Comparison to other related works to HPBF in the literature for the last ten years.

Ref.	Technique used	Substrate h(mm)/ ε_r	F_C GHz	3 dB FBW, Freq. Band GHz	$S_{11}= S_{22}$ (dB)<	$S_{12}=S_{21}$ (dB)	Circuit area
This Work Single Layer	UTL	0.813/3.55	3.75	21.1% 3.45– 4.24	-10	-1.5	0.31 λg × 0.24 λg
	UTL	0.813/3.55	6.61	30% 5.62– 7.6	-11.66	-1.17	0.66 λg × 0.27 λg
[4] Single Layer	UTL	0.508/2.33	5.66	4.45% 5.608– 5.86	-23.9	-4.4	0.97 λg × 0.68 λg
[14] Single Layer	UTLs *via* holes	1.52/2.2	0.92	0.004% 0.92 -0.924	-18.9	-7.7	0.32 λg x0.17 λg
[26] Single Layer	FSLs	1.6/4.4	1	25% 0.875 -1.125	-27	-1.37	0.26 λg x0.2 λg
[7] Single Layer	UTLs using IPT on LCP	0.1/3.2	30.4	15% 28.12 – 32.68	-18.9	2.41	0.50 λg × 0.48 λg
[27] Multilayers	LHRHTLs by LCPT	different/ different	2.06	24% 1.813 – 2.307	-11	1.5	11.8 × 4.8 mm2
[8] Single Layer	NCLs	0.127 /2.94	31.57	3.45% 31.02-32.11	-9	-3.5	2.16 λg × 0.25 λg
[28] Multilayers	UTLs with apertures for coupling	1.27 /6.15	2.5	4.75% 2.44 -2.56	-11	-1.65	0.42 λg × 0.41 λg
[20] Single Layer	UTLs with square DGS	1.6/4.4	3	20.72% 2.9 – 3.1	-46.64	-0.3	0.23 λg × 0.18 λg
[29] Multilayers	Higher εr substrate	0.381/9.8	8	15% at 7.4 – 8.6	-14.5	-3	0.48 λg x 0.48 λg
[22] Single Layer	UTLs with dumbbell and (SHPR) DGSs	1.524/3.48	2.4	29% 2 – 2.7	-26	-3	0.65 λg × 0.33 λg
[19] Single Layer	UTLs with square DGS	0.348/1.524	2.92	97.33% 2.82 – 3.02	-19.5	-1.6	0.87 λg × 0.29 λg
[9] Single Layer	UTLs with square DGS	1.58 /2.2	9.3	30.11% 9.11 – 9.39	−19.2	- 3.7	1.81 λg × 0.35 λg
[6] Single Layer	S.C. comb-lines with RSRL resonator	0.5 /2.55	2.45	0.4082% 2.4 -2.5	-36.71	-0.36	0.26 **λg** × 0.1 λg
[30] Multilayers	UTLs using IPDT	different/different	95.5	0.83% 91.9 -99.9	-10	-5	0.95 × 0.4 mm^2

5. CONCLUSION

Two 5G low-frequency band (3.7 GHz - 4.2 GHz and 5.975 GHz -7.125 GHz) Hairpin Bandpass Filters (HPBFs) suitable for filtering and reconfigurable antenna applications are designed and simulated in this work. At the two bands, the results are good in terms of return and insertion losses. A filter is fabricated and tested at both frequency bands. The measured S_{11} and S_{12} are < -10 dB and <- 10.66 dB and – 1.5 dB and -1.17 dB, through 3.45 GHz- 4.25 GHz and 5.62 GHz – 7.6 GHz, respectively. As

future work, many techniques can be used to reduce the size of this filter and to get further harmonics suppressions, especially at the first band (3.7 GHz – 4.2 GHz) to be compatible with 5G narrow band / UWB reconfigurable antenna.

ACKNOWLEDGEMENTS

This work was supported by the University of Science Malaysia through the RUI Grants (1001/PELECT/801405) and (304/PELECT/6315294).

REFERENCES

[1] J.-S. G. Hong and M. J. Lancaster, Microstrip Filters for RF/Microwave Applications, vol. 167, John Wiley & Sons, 2011.

[2] Q. Abdullah, N. S. M. Shah, N. Farah, W. A. Jabbar, N. Abdullah, A. Salh et al., "A Compact Size Microstrip Five Poles Hairpin Band-pass Filter Using Three-layers Structure for Ku-band Satellites Application," Telkomnika, vol. 18, no. 1, pp. 80-89, 2020.

[3] K. K. Sethi, A. Dutta, G. Palai and P. Sarkar, "Hairpin Structure Band-pass Filter for IoT Band Application," New Paradigm in Decision Science and Management, Part of the Advances in Intelligent Systems and Computing Book Series (AISC), vol. 1005, pp. 399-405, Springer, 2020.

[4] N. A. Wahab, W. N. W. Muhamad, M. M. A. M. Hamzah, S. S. Sarnin and N. F. Naim, "Design A Microstrip Hairpin Band-pass Filter for 5GHz Unlicensed WiMAX," Proc. of IEEE International Conference on Networking and Information Technology, pp. 183-186, Manila, Philippines, 2010.

[5] M. F. M. Yusoff, M. A. M. Sobri, F. Zubir and Z. Johari, "Multiband Hairpin-line Bandpass Filters by Using Metamaterial Complementary Split Ring Resonator," Indonesian Journal of Electrical Engineering and Informatics (IJEEI), vol. 7, pp. 289-294, 2019.

[6] S. K. Azam, M. I. Ibrahimy, S. Motakabber, A. Z. Hossain, and M. S. Islam, "A miniaturized hairpin resonator for the high selectivity of WLAN bandwidth," Bulletin of Electrical Engineering and Informatics, vol. 8, pp. 916-922, 2019.

[7] H.-l. Kao, C.-L. Cho, X. Dai, C.-S. Yeh, X.-Y. Zhang, L.-C. Chang et al., "Hairpin Bandpass Filter on Liquid Crystal Polymer Substrate Using Inkjet Printing Technology," Proc. of IEEE MTT-S International Microwave Symposium Digest (MTT), pp. 1-4, Seattle, WA, USA, 2013.

[8] H. Shaman, S. Almorqi, O. Haraz and S. Alshebeili, "Hairpin Microstrip Bandpass Filter for Millimeter-wave Applications," Proceedings of IEEE Mediterranean Microwave Symposium (MMS2014), pp. 1-4, Marrakech, Morocco, 2014.

[9] B. Adli, R. Mardiati and Y. Y. Maulana, "Design of Microstrip Hairpin Bandpass Filter for X-band Radar Navigation," Proc. of the IEEE 4th International Conference on Wireless and Telematics (ICWT), pp. 1-6, Nusa Dua, Indonesia, 2018.

[10] K. Kavitha and M. Jayakumar, "Design and Performance Analysis of Hairpin Bandpass Filter for Satellite Applications," Procedia-Computer Science, vol. 143, pp. 886-891, 2018.

[11] S. Ono and K. Wada, "Design and Fabrication of 3-Pole BPF Configured by Hairpin Resonators and Different Types of Coupling and Feed Types at 20 GHz," Proc. of IEEE Asia-Pacific Microwave Conference (APMC), pp. 1363-1365, Kyoto, Japan, 2018.

[12] M. Fadhil, H. Wijanto and Y. Wahyu, "Hairpin Line Bandpass Filter for 1.8 GHz FDD-LTE eNodeB Receiver," Proc. of the International IEEE Conference on Radar, Antenna, Microwave, Electronics and Telecommunications (ICRAMET), pp. 134-136, Jakarta, Indonesia, 2017.

[13] O. Sharifi-Tehrani, "Design, Simulation and Fabrication of Microstrip Hairpin and Interdigital BPF for 2.25 GHz Unlicensed Band," Majlesi Journal of Telecommunication Devices, vol. 6, pp. 115-118, 2017.

[14] F. Y. Zulkifli, R. Saputra and E. T. Rahardjo, "Microstrip Hairpin Bandpass Filter Using Via Ground Holes for 923 MHz RFID Application," Proc. of the International Symposium on Antennas and Propagation (ISAP), Jeju, pp. 1-4, [Online], Avaiable: https://www.ieice.org/cs/isap/ISAP_Archives/2011/pdf/[FrD4-6]%20A14_1003.pdf , 2011.

[15] A. Lalbakhsh, A. A. L. Neyestanak and M. Naser-Moghaddasi, "Microstrip Hairpin Bandpass Filter Using Modified Minkowski Fractal Shape for Suppression of Second Harmonic," IEICE Transactions on Electronics, vol. 95, pp. 378-381, 2012.

[16] M. Naser-Moghadasi, M. Alamolhoda and B. Rahmati, "Spurious Response Suppression in Hairpin Filter Using DMS Integrated in Filter Structure," Prog. In Electromag. Research, vol. 18, pp. 221-229, 2011.

[17] K. Vidhya and T. Jayanthy, "Design of Microstrip Hairpin Band Pass Filter Using Defected Ground Structure and Open Stubs," Proc. of the International Conference on Information and Electronics Engineering (IPCSIT), vol. 6, pp. 268-272, IACSIT Press, Singapore, 2011.

[18] M. Othman, N. M. Zaid, M. A. Aziz and H. Sulaiman, "3 GHz Hairpin Filter with Defected Ground Structure (DGS) for Microwave Imaging Application," Proc. of the IEEE International Conference on Computer, Communications and Control Technology (I4CT), pp. 411-414, Langkawi, Malaysia, 2014.

[19] N. Ismail, T. S. Gunawan, T. Praludi and E. A. Hamidi, "Design of Microstrip Hairpin Bandpass Filter for 2.9 GHz–3.1 GHz S-band Radar with Defected Ground Structure," Malaysian Journal of Fundamental and Applied Sciences, vol. 14, pp. 448-455, 2018.

[20] V. S. Kershaw, S. S. Bhadauria and G. S. Tomar, "Design of Microstrip Hairpin-line Bandpass Filter with Square Shape Defected Ground Structure," Asia-Pacific Journal of Advanced Research in Electrical and Electronics Engineering, vol. 1, pp. 21-30, 2017.

[21] T. Hariyadi and S. Mulyasari, "Design and Simulation of Microstrip Hairpin Bandpass Filter with Open Stub and Defected Ground Structure (DGS) at X-Band Frequency," Proc. of the 2nd International Conference on Innovation in Engineering and Vocational Education, IOP Conference Series: Materials Science and Engineering, vol. 306, p. 012124, Manado, Indonesia, 2018.

[22] H. Sajjad, A. Altaf, S. Khan and L. Jan, "A Compact Hairpin Filter with Stepped Hairpin Defected Ground Structure," Proc. of the 21st IEEE International Multi-topic Conference (INMIC), pp. 1-5, Karachi, Pakistan, 2018.

[23] N. Ismail, S. M. Ulfah, I. Lindra, A. S. Awalluddin, I. Nuraida and M. A. Ramdhani, "Microstrip Hairpin Bandpass Filter for Radar S-Band with Dumbbell-DGS," Proc. of the 5th IEEE International Conference on Wireless Communications and Telematics (ICWT), pp. 1-4, Yogyakarta, Indonesia, 2019.

[24] J. Ye, D. Qu, X. Zhong and Y. Zhou, "Design of X-band Bandpass Filter Using Hairpin Resonators and Tapped Feeding Line," Proc. of IEEE Symposium on Computer Applications and Communications, pp. 93-95, Weihai, China, 2014.

[25] T. Singh, J. Chacko, N. Sebastian, R. Thoppilan, A. Kotrashetti and S. Mande, "Design and Optimization of Microstrip Hairpin-line Bandpass Filter Using DOE Methodology," Proc. of the IEEE International Conference on Communication, Information & Computing Technology (ICCICT), pp. 1-6, Mumbai, India, 2012.

[26] A. Lotfi-Neyestanak and A. Lalbakhsh, "Improved Microstrip Hairpin-line Bandpass Filters for Spurious Response Suppression," Electronics Letters, vol. 48, pp. 858-859, 2012.

[27] J. Ni, "Development of Tunable and Miniature Microwave Filters for Modern Wireless Communications," Heriot-Watt University, [Online], Available: http://hdl.handle.net/10399/2843, 2014.

[28] N. Chami, D. Saigaa, A. Djaiz, R. AlThomali and M. Nedil, "A New Miniature Microstrip Two-layer Bandpass Filter Using Aperture-coupled Hairpin Resonators," International Journal of Advanced and Applied Sciences, vol. 4, pp. 10-14, 2017.

[29] M. Tan, Y. Xuan, Y. Ma, L. Li and Y. Zhuang, "Design of C-band Interdigital Filter and Compact C-band Hairpin Bandpass Film Filter on Thin Film Substrate," RF and Microwave Microelectronics Packaging II, pp. 63-73, Springer, 2017.

[30] B. Chen, Y. Tang, H. Zhu, H. Yue, Z. Wen and X. Deng, "Design of W Band Hairpin Filter with IPD Technology," Proc. of IEEE MTT-S International Wireless Symposium (IWS), pp. 1-3, Guangzhou, China, 2019.

[31] C. Schuster, A. Wiens, M. Schüßler, C. Kohler, J. Binder and R. Jakoby, "Hairpin Bandpass Filter with Tunable Center Frequency and Tunable Bandwidth Based on Screen Printed Ferroelectric Varactors," Proc. of the 46th IEEE European Microwave Conference (EuMC), pp. 1425-1428, London, UK, 2016.

[32] C. Schuster, L. Schynol, E. Polat, E. Schwab, S. Schmidt, R. Jakoby et al., "Reconfigurable Hairpin Filter with Tunable Center Frequency, Bandwidth and Transmission Zero," Proc. of the IEEE MTT-S International Microwave Workshop Series on Advanced Materials and Processes for RF and THz Applications (IMWS-AMP), pp. 79-81, Bochum, Germany, 2019.

[33] M. A. Almahadeen and A. M. Matarneh, "Performance Assessment of Throughput in a 5G System," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 6, no. 3, pp. 303-316, 2020.

[34] M. A. Taher, "Enhanced 5G Throughput Using UFMC Multiplexing," Journal of Southwest Jiaotong University, vol. 54, no. 5, pp. 1-11, 2019.

[35] I. Parvez, A. Rahmati, I. Guvenc, A. I. Sarwat and H. Dai, "A Survey on Low Latency Towards 5G: RAN, Core Network and Caching Solutions," IEEE Communications Surveys & Tutorials, vol. 20, pp. 3098-3130, 2018.

[36] J. Sachs, G. Wikstrom, T. Dudda, R. Baldemair and K. Kittichokechai, "5G Radio Network Design for Ultra-reliable Low-latency Communication," IEEE Network, vol. 32, pp. 24-31, 2018.

[37] O. N. Yilmaz, Y.-P. E. Wang, N. A. Johansson, N. Brahmi, S. A. Ashraf and J. Sachs, "Analysis of Ultra-reliable and Low-latency 5G Communication for a Factory Automation Use Case," Proc. of the IEEE International Conference on Communication Workshop (ICCW), pp. 1190-1195, London, UK, 2015.

[38] R. Pawlak, P. Krawiec and J. Żurek, "On Measuring Electromagnetic Fields in 5G Technology," IEEE Access, vol. 7, pp. 29826-29835, 2019.

[39] 5G Americas, "5G Spectrum Vision," 5G Americas White Paper, p. 50, [Online], Avaialble: https://www.5gamericas.org/5g-spectrum-vision/, Feburary 2019.

[40] S. Saleh, W. Ismail, I. S. Z. Abidin and M. H. Jamaluddin, "N-way Compact Ultra-wide Band Equal and Unequal Split Tapered Transmission Lines Wilkinson Power Divider," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 6, no. 3, pp. 291-302, September, 2020.

[41] Em:Talk, "Microstrip Line Calculator, " [Online], Available: http://www.emtalk.com/mscalc.php.

[42] D. M. Pozar, Microwave Engineering, Hoboken, NJ: John Wiley & Sons, 2012.

ملخص البحث:

في هذه الورقة، يتم تصميم ومحاكاة وتصنيع مرشِّح تمرير نطاق ترددي على شكل مُنعطَف عند نطاقين ترددّيين منخفضين: (3.7-4.2 جيجا هيرتز، و 5.975-7.125 جيجا هيرتز). وهذا المرشِّح سيكون جزءاً من مشروع الهوائيات المنتمي الى الجيل الخامس الذي نعمل عليه، والقابل لإعادة التّشكيل لكل من النطاقات الضيقة/ النطاقات فائقة العرض (UWB/NB)، والذي من شأنه أن يلعب دوراً أساسياً في الشبكات اللاسلكية الحديثة؛ مثل شبكات الراديو الإدراكي (CRs) وعلى مدى النطاقين الترددّيين، حصلنا من المرشِّح على مواءمةٍ واستجابة إرسال جيدتين، مع عرض نطاق محسَّن. وبلغ معامل الانعكاس المقاس للمرشِّح المقترح أقل من (-10) ديسيبل وأقل من (11.66) ديسيبل على مدى النطاقين الترددّيين (3.45-4.25 جيجاهيرتز، و 5.62-7.6 جيجا هيرتز) على الترتيب. أما معامل الإرسال فكان في حدود (-1.5) ديسيبل، و (-1.17) ديسيبل عند الترددين المركزيين (3.75 جيجاهيرتز، و6.61 جيجاهيرتز على التوالي. وفي هذا البحث، استخدمت برمجية المحاكاة البِنيوية للترددات العالية (HFSS) لإجراء المحاكاة. وقد تم التحقق من نتائج المحاكاة للموجة الكاملة، مع القياسات المرتبطة بالمعدات.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 07, No. 01, March 2021.

IoT Security for Smart Grid Environment: Issues and Solutions

Yuvaraaj Velayutham, Nur Azaliah Abu Bakar, Noor Hafizah Hassan and Ganthan Narayana Samy

(Received: 27-Jul.-2020, Revised: 10-Oct.-2020, Accepted: 27-Oct.-2020)

Abstract

The Internet of Things (IoT) is the Internet's latest innovation today, where every physical object is situated or where measurement, as well as communication capacities, can be seamlessly synchronized to the Internet at various rates. The most important infrastructure, the smart grid, is called the extended version of the power grid with comprehensive Internet infrastructure. The smart grid will include billions of intelligent appliances: intelligent meters, actuators, vehicles and so on, despite a few correspondence infrastructures, whether public or private. Notwithstanding, security is viewed as one of the primary considerations hampering the large scope reception and arrangement of both the IoT vision and the smart grid. To date, the issues of IoT for the smart grid are rarely discussed empirically in any academic research. This study aims to examine security problems and challenges in the IoT smart grid system. Findings show various issues that we can categorize into three parts; component issues, system issues and network issues. As a result, this study proposes a mitigation plan for the problems highlighted by developing an IoT smart grid security component model.

Keywords

Advanced metering infrastructure, Cybersecurity, Internet of things, Smart grid, Smart meter.

1. Introduction

Internet of Things (IoT) is an emerging domain that evolves from interfacing machines and people to combine smart objects. IoT can be a device that will be an embedded processor or computation device with advanced communications of the machine to machine correspondences [1]. For this intelligence and interconnection, IoT systems include integrated sensors, actuators, processors and transceivers. Sensors and actuators are instruments that support the physical environment. Sensor data must be stored and analyzed intelligently to draw useful inferences from it. An *actuator* is a device that is used to induce a change in the environment. Storage and processing of data can be carried out on the edge of the network itself or in a remote server [2].

IoT devices typically connect to the Internet through the IP (Internet Protocol) stack. IoT devices can also connect locally through non-IP networks which consume less power and connect to the Internet *via* a smart gateway. The leading communication technologies used in the IoT world are IEEE 802.15.4, low-powerWiFi, 6LoWPAN, RFID, NFC, Sigfox, LoraWAN and other proprietary protocols for wireless networks [2]. Nevertheless, some smart devices are still linked to the network through non-IP protocols, such as Bluetooth, RFID and NFC. With the advent of IoT, the smart tool and the protocols, such as IP, TCP or UDP, would be entirely seamlessly related [3]-[4].

The smart grid is an intelligent power system that provides two-way communication back to the database from power generation to electricity distribution to households. A smart grid includes technology applications that promote the incorporation and penetration of renewable energy [5]. It will be necessary to accelerate the production and widespread use of plug-in hybrid electric vehicles (PHEVs) and their potential use as grid storage. Among the benefits of the smart grid are that it is able to provide more reliable power, generate more efficient renewable power and use a mix of energy sources, in addition to being able to work with smart devices and smart homes and most importantly, it will reduce our carbon footprint [6]. Essentially, the smart grid, together with wireless communication-connected smart metres, will monitor how much energy a net-positive enterprise produces and reimburses.

The smart grids form a part of an IoT system that allows all forms of lighting, traffic signals, transport

Y. Velayutham, N. A. A. Bakar, N. H. Hassan and G. N. Samy are with Razak Faculty of Technology and Informatics, Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia. Emails: yuvaraajvelayutham@icloud.com, azaliah@utm.my, noorhafizah.kl@utm.my and ganthan.kl@utm.my

congestion, parking areas, road warnings and early detection of such items as power inflows due to earthquakes and extreme weather [7]. Wireless devices, such as sensors, radio modules, gateways and routers, are part of the technology that makes the IoT-enabled energy grid "smart". These devices ensure sophisticated connectivity and communication to encourage customers to take more energy use decisions in order to save electricity and expenses in cities and to allow electricity authorities to restore power more quickly after an emergency [5]. The smart grid enables a power provider to analyze system health considerably more fully than before. For example, a power utility can detect real-time demands for power with smart metres with granularity and exactness that is simply not possible with older technology.

Overall, smart grid connectivity networks should comply with time synchronization, reliability, latency and data criticality including support for multicast [8]. Furthermore, interoperability is a big problem in smart grid networking [9]. Most importantly, any device connected to communication systems may be subject to unscrupulous and malicious individuals, whose primary purpose is to access sensitive information [7]. This makes the critical infrastructure to be monitored and operated in a much more efficient manner. This introduces security challenges to the smart grid infrastructure; for example, man-in-the-middle, session hijacking, spoofing and Denial of Service (DOS) attacks [10].

Based on the IoT security issues and challenges arising in a smart grid environment, this paper aims to investigate the existing possible security vulnerabilities. In order to achieve that, this paper will start with a discussion on the use of IoT in the smart grid environment (Section 2), followed by the analysis of security attacks on IoT for smart grid (Section 3). Then, Section 4 will explain the mitigation actions and in Section 5, proposed components for IoT secured smart grid are presented. Finally, in Section 6, the paper concludes with all the related findings.

2. USING IOT IN THE SMART GRID ENVIRONMENT

The future of grid networks is IoT. The National Institute of Standards and Technology (NIST) defined IoT for the smart grid as integrating the old power grid with the current ICT emerging grid [11]. Unlike traditional power grids, the smart grid can sustain or manage power distribution's demand, achieve power delivery efficiency and minimize energy losses [12]. According to the US Department of Energy's Office of Electricity Delivery and Energy Reliability, a smart grid is a digital-infrastructure grid allowing two-way contact between the utility and its customers. The "smart" aspect is its ability to adjust to variable supply and demand. Technologies that yield today's IoT-enabled "smart" energy grid include wireless devices, like sensors, radio modules, gateways and routers [13]. These devices provide advanced networking and communications that encourage customers to make smart choices in terms of energy use, allow cities to conserve resources and costs and enable energy agencies to recover power more quickly after an outage.

2.1 Overview of Smart Grid Design

Smart grid networks are designed to connect various remote controls, smart meters, smart grids, cameras and other network-based products [11]. IoT allows current grid connectivity across the two-way data source and network across the energy system using IoT devices that interconnect customers, distributors and service providers. It can also limit human interference to track meters, home portals and other relevant devices to ensure that grid power is safely managed [14]. Most smart grids follow the NIST defined smart grid model illustrated in Figure 1.

In general, NIST outlines four (4) types of smart grid entity which are: the service provider, the operations, the distribution and the markets. There are three (3) main groups of users; namely, the building or commercial users, the industrial users and home users. Using IoT in the smart grid allows two-way data exchange between the entities and all components of the smart grid [16]. Beside sensors and drives, other intelligent devices and transmitting and storage fields, the touch is conceivable, given the use of smart meters and other smart devices on the end-customer side. This allows the monitoring of energy usage and demand while enabling consumers to watch and change their activities [17].

IoT structures can be guided to another step of efficiency and execution due to a greater degree of controllability and perceptibility, which gives power systems enormous benefits. There are many advantages associated with the smart grid application of IoT. For example, the Advanced Metering

Infrastructure (AMI) can be accessed without any trouble using the IoT smart grid [18]. AMI is responsible for processing, disassembling, storing and distributing the intelligent metering information submitted to the utility organization's billing, outage control and electricity demand forecasting systems. Accessibility of real-time calculation provides consumers and manufacturers with critical signs to better satisfy their energy needs and supplies [11].

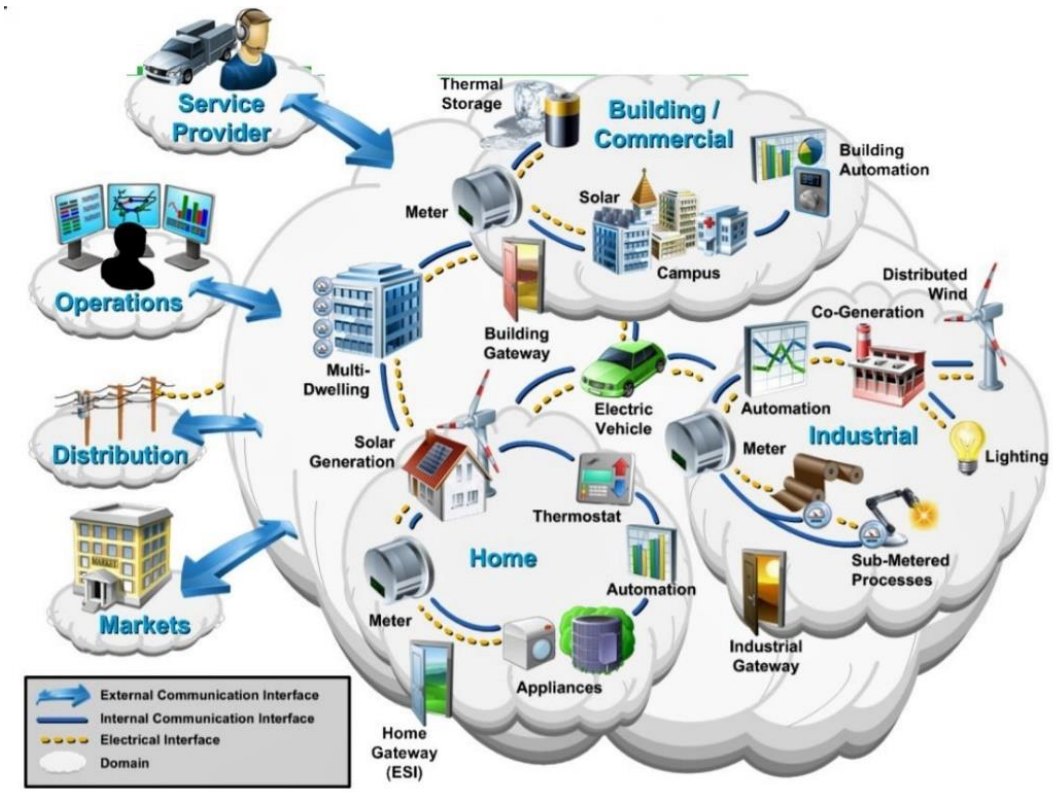


Figure 1. NIST customer domain smart grid model [15].

The IoT smart grid can quickly repair itself (self-recovery) in the event of any external or internal aggravation or risk [18]. It also allows the self-rebuilding of the system, after attacks, cataclysmic events, power outages or breakdown of the network components by complex reconfigurations in order to recover electricity. Besides, it also can create a micro-scale grid and self-sufficiently protected islands in the event of power loss, even as it distinguishes the source of energy leakage [12]. This increases the power grid's performance, modelling and analysis, thereby increasing the power grid's consistency.

Due to the usage of a wide range of sensors, actuators and intelligent meters used to monitor the entire power grid, they may send intermittent data to the utility on-demand or on other occasions while often responding to demands from customers, due to their willingness to communicate in both directions [19]. Added to delivering details on the state of the last mile grid, these instruments can be remotely controlled, managed and operated to include updated Supervisory Control and Data Acquisition (SCADA) features. This helps promote bi-directional electricity, because end-customers can also sell abundant energy from homes, particularly from sources during peak hours, such as sun-based or bio-gas sources around the building [10]. Using the IoT smart grid, this can easily track demand and request reactions on the smart grid, thereby allowing complex electricity pricing components to be influenced. Competitive energy pricing increases peak load management capabilities by charging more significant prices on top occasions to discourage consumption and lower prices at off-peak periods, encouraging reduced utilization and idle use of electricity [14].

The provision of real-time, fast and bi-directional information sharing allows more excellent connectivity with end-users of energy. It provides utilities with a critical insight into consumer consumption habits that further enhance the interconnected grid [11]. The extensive installation of sensors, as well as signal processing and real-time connectivity equipment, helps assess individual grid pieces' position. This helps handle energy, plan for future and current opportunities properly and make

the transmission lines and transformers function in this manner, contributing to a reliable transfer of electricity. These reviews can result in clearly identifying signs of line defects, a decrease in the risk of catastrophic failure and a decline in operation and repair costs in these lines, thereby improving the unwavering efficiency of the transmission system [8].

2.2 The Benefits of Smart Grid

Today, the current grid faces challenges daily; for example, blackouts, current overloading and service disruption in a particular area. However, this is only determined if the end-user files a complaint on the occurrence. In contrast, with the implementation of smart monitoring on the grid, we can react proactively to the downtimes. Smart grid technology is the key to easy integration and customer reliability [8]. This system can solve problems very quickly in a current system since it can reduce the workforce and strive for sustainable, effective, secure and quality energy for all consumers. According to [13], [20]-[22], the following are the benefits of smart grid technology.

Smarter Energy Use: The technology from smart grid helps reduce energy usage and costs by using and retaining data. Smart lighting, for example, can be tracked around various areas by using smart urban technology, immediately accommodating settings like rain, changing the production to suit traffic conditions or time of the day and instantly identifying light outages and fixing them. Users can change the temperature of their home thermostats for use in consumer applications when working or on holiday [21].

Cleaner Energy Use: Smart grid technologies have lower battery requirements, are carbon-efficient and are designed to reduce the maximum load of distribution feeders. The US Department of Energy is incorporating renewable technologies into its intelligent IoT management framework for sustainable solutions. There is an ability to support all levels of the distribution network through integrated wind turbines, solar panels, micro-grid technologies and feeder automation systems [13].

Lower Costs: Today, power outages and interruptions of the electricity system cost Americans at least $150 billion per year and set the price tag of about $500 per person. If the world's population continues to expand, the older networks cannot meet growing demands. Smart grids are built to reduce costs by monitoring intelligent electricity and redirecting the source from the moment a power failure is detected [13].

Improved Transportation and Parking: IoT smart sensors can collect data in real-time for drivers and authorities to obtain information. Ultimately, this would reduce traffic delays, provide better parking options, alert drivers to traffic events and townscape structural damage and allow electronic payment at road tolls and car park metres. Future IoT technology is also expected to charge electric vehicles wirelessly [20].

Assistance in Waste and Water Management: The smart network will benefit smart cities by rising their productivity and reducing their waste management solution costs. In order to track inventory and minimize fraud, IoT apps can provide real-time data. Cloud-based monitoring and traffic management can increase time and scheduling on lorry routes. Smart energy analytics can collect data about water flow, pressure, temperature and much more, which allows customers to keep track of their usage practises [21].

Energy Enablement in Developing Countries: Smart grids could be used to convert power to sparsely populated regions from simple on-off electrification methods; for example, from battery-based household electrification to neighbourhood grids, which would then connect to national and regional grids. These grids would be crucial for the implementation of new energy infrastructures in developing countries that are suffering from the consequences of the overflowing population. This will potentially pave the way for economic development [22].

IoT allows the remote detection and control of smart grid resources through an adaptable communication network that enables better synchronization between the physical environment and PC-based control systems [12]. This is proven to improve productivity and precision and allow the grid to meet present-day and future energy needs. However, the advancement of technologies always comes with challenges, especially in terms of security. The following section will discuss the IoT security issues in the smart grid area.

3. Related Works on IoT for Smart Grid Security Issues

In the literature, the need for detailed investigation of smart grid security issues has been suggested [9]-[10], [17]. The energy industry is essential, and without doubt, cybersecurity challenges will be present and faced by smart grid IoT devices. IoT devices on a grid environment may reach hundreds and thousands of tools spanning across the regions; they are most vulnerable to cyberattacks. Malware threat is a significant hindrance to efficient information exchange on the IoT [23]. A cyber-attack on these IoT devices may cause loss of valuable data or even halt in company production. For example, in 2015, an attack on the Ukrainian power grid made it possible for hackers to stop and monitor the network with BlackEnergy malware in order to hack the grid and SCADA Network. This resulted in a vast blackout, where over 700,000 users did not receive any electricity [24]. This clearly showed that security is a significant impediment to the introduction or service of the IoT-based smart grid.

What makes IoT protection more complicated is the big number of tools deployed which would not add security to the devices. Unlike computers, we would be able to reinstall or wipe the entire network, but most IoT devices do not support that just yet. A study by Salameh, Dhainat and Benkhelifa [25] showed that the efficiency of these wireless network sensor systems varies as many IoT devices are mostly tied to the manufacturers. The end consumers do not have access to fiddle with the devices as the manufacturers lock them [26]. This paper presents IoT security issues in the smart grid distributed over three classifications; firstly the component security issues, secondly the system security issues and thirdly the network security issues.

3.1 Issues of Component Security

Many experts and engineers have evaluated the safety deficiency of the smart meter. A malicious code attack may disrupt the expected behaviour of the intelligent meter. DOS attacks can be powered to avoid the contact of legitimate smart meters with different nodes. For example, an unapproved node can perform eavesdropping (passive man-in-the-middle) stealthily to identify sensitive data about the client's energy use, current charges and appliances used in the household [8]. Furthermore, an attacker can send out false information imitating to be the legitimate smart meter. There is a function in smart meters called Remote-Connect-Disconnect. This function enables operators or engineers to collect the maintenance information from the meter for troubleshooting purposes. This allows a back door in the smart meter itself for the attacker, and the attacker may use this to falsify the data of the meters [17].

The IoT smart grid network has been named a home portal, which receives information from the smart meter on power usage and shows it on the household's mobile device or even the computer. The home app or smart meter provides a service provider with power usage information for budgetary benefit. However, eavesdropping will destroy this gateway communication [17]. Another threat is posed by the Phasor Measurement Unit (PMU) device, which can gather field estimations sending voltages and electrical quantities to the Phasor Data Connector (PDC). The PDC perceives the information from various PMUs, mixes it as a single post.

Moreover, it interacts with other operating domains. A malicious node can spoof PMU attacks, alter PMU messages, provide estimated vitality data and even replay PDC and PMU messages. Such attacks impact critical decision-making processes, such as fault detection and location of incidents. For instance, when an attacker replays an old PMU message that contains vitality estimation misfortunes or line blackouts, the operating system may choose to kill the power for a zone [14].

3.2 Issues of System Security

Smart grid operations have a few control frames which with similar goals and specifications. The Energy Management System (EMS) and the Delivery Management System (DMS) will assume transmission control and energy dispersal. At the same time, SCADA will assist in electronic power systems [5]. Hence, such control and management systems that conduct critical tasks, such as regulating voltage, identifying blackouts, transferring power intensity for distribution and transmission of electricity, should be protected from attacks. They are vulnerable to malicious node attacks that target the DoS control systems, which will later affect their functionality.

Similarly, a false information attack against a control system will influence the smart grid automated decisions. For instance, sending invalid measure energy will affect the distribution and transmission

activities. At the same time, systems depend on false PMU data control choices. A malicious node can replay PMU transmission estimation information; thus, the control centre concludes the decision based on the PMU data [27]-[28]. Messages are transmitted in multi-hop within the AMI system and interchanges between smart meters and the control centre. Therefore, it is possible to accelerate man-in-the-middle attacks and change the energy consumption data before transmitting messages. Moreover, there is a possibility for an attacker to stealth-listening on data trading between the smart meter and the control centre through a remote communication channel.

The Wide Area Monitoring Protection and Control System (WAMPAC) will share information on transmission with other control systems, delivering real-time monitoring and warning capability and maintaining effective transmission and aggregation of electric grids. The grouping of attacks indicates that the WAMPAC system is additionally vulnerable to Denial-of-Service (DOS) attack. At the same time, applications give real-time activity and performance [27]. DOS attacks can occur in different layers of communication. For example, a malicious node can transmit jamming that fills the small-medium with noise flags and can severely impact the data in real-time [28]. The jamming attack will harm the system's functionality, and an authentic node cannot get messages back. Besides, different kinds of assaults; for example, spoofing and man-in-the-middle, can be pushed only as the full or partial channels of communication can be jammed. One more aspect to be considered is the malware attack. Malware threat is a significant hindrance to efficient information exchange on the IoT, including for the smart grid [23]. Modelling malware propagation is one of the most imperative applications aimed at understanding the mechanisms for protecting the smart grid environment.

3.3 Issues of Network Security

The Neighborhood Area Network (NAN) protects and tracks smart meter connections in a single geographical area. Conventions that are available on NAN systems include the Routing Protocol for Low Power and Lossy Networks (RPL), the Minimum Transmission Energy Protocol (MTE) and the *Ad Hoc* on Demand Multipath Distance Vector (AOMDV) [15]. The smart grid network's basic features can be targeted under various attacks by the RPL routing protocol for NAN networks. For example, the Wireless Sensor Network (WSN) attacks will affect the IoT RPL directing convention. WSN is a group of spatially deployed sensor nodes that acknowledge or remotely observe diverse environmental variables or natural events. The sensor nodes are used to collect the surrounding natural events, process data, respond to base station requests and commands or transmit the data to other neighbour sensors. These features indirectly expose WSN to more types of attacks, including DoS attack due to the open wireless communication and physical risks [29]. Currently, the Home Area Network (HAN) can handle the link between the smart meter and the HAN devices. HAN can use distinctive networking technologies, such as Zigbee, Bluetooth and WiFi [3]. Present protection protocols, such as IDS, IPsec, VPN and PKI, can be extended to the smart grid; however, it is still inadequate to guarantee that these protocols are secured for the smart grid environment.

In summary, we conclude the possible attack types and their classification associated with the Smart Grid IoT implementation in Table 1.

Table 1. IoT smart grid possible security issue and attack types.

Security Issue Classification	Attacked Type	Related Study
Component Security	• Malicious code attack • DOS attacks • Passive man-in-the-middle • False information from smart meter • Phasor Measurement Unit (PMU) attack • Phasor Data Connector (PDC) attack	[8], [14], [17]
System Security	• Energy Management System (EMS) attack • Delivery Management System (DMS) attack • False information attack • Man-in-the-middle attacks • Stealth-listening • WAMPAC system attack	[5], [23], [27], [28]

	• DOS attack • Malware threat	
Network Security	• Routing Protocol attack • Wireless Sensor Network (WSN) attack • DoS attack	[3],[15], [29]

Until now, the concept of IoT has concentrated on the demand side, with little attention to the supply side. Smart grid technologies all contribute to efficient IoT energy management solutions that are currently lacking in the existing security framework. The next section explains the mitigation control for the security concerns highlighted.

4. MITIGATION OF SECURITY CONCERNS

With the current emerging technology and fast-evolving internet technologies, IoT devices are an essential element in the network. However, with the implementation of IoT with the smart grid, it should be further secured and guarded at all times. From the previous discussion on smart grid security issues, this paper highlighted a few components that could be implemented to prepare the environment to face the attacks over the smart grid network.

Firstly, a guideline must be established to ensure the rules and regulations of IoT in a smart grid environment. This includes access to grants and privileges. Getting these predefined access privileges to grid devices and network functionality eliminates the possibility of malicious user access [30]. The IoT-based smart grid is a centrally managed and optimized cyber-physical system; access controls are necessary to ensure network connectivity to customers and devices. For example, in access control aspects, Discretionary Access Control (DAC), Compulsory Access Control (MAC) and Roll-based Access Control (RBAC) will extend unwavering consistency and dispense with possible safety hazards.

While deploying IoT systems, traffic between IoT devices and control centres, including utility-supported servers, needs to be encrypted. As mentioned by [1], encoding messages using rigorous encryption methods is crucial, because it diminishes an intruder's ability to decrypt data or produce useful information for trickery. It ensures both material security and secrecy, as it involves identifying computers in the system and authorizing what each network machine completes. System authentication is typically the central stage of software exchange sessions, often culminating in a shared session key for encryption and verification of data packets and maintaining performance validity [31]. Since IoT smart grid contact is time-sensitive and traffic-intensive, an authentication scheme will need little messaging interaction between grid devices. The authentication process will ensure that the meter will not accept commands from an unapproved system. In contrast, the validation process will provide identity verification and acceptance.

IoT systems must be scalable to be upgraded to implement bugs and software upgrades fairly and effectively [32]. Unfortunately, most developers now develop software without contemplating applying any leap of imagination to update potential firmware [33]. Nonetheless, they will accept that creativity, operating systems and computer code look at potential threats and flaws in the future and improvements will be made to address these problems. Deploying firmware upgrades can be difficult if not designed to provide upgrades. Given a smart IoT system's sheer size, frequent firmware overhaul upgrades are the sensible and rational approach relative to the significant replacement of outdated systems in reach. Cybersecurity concerns are significantly heightened when businesses mix modern and old technologies, irrespective of overall network security. Consequently, maintaining a steady protocol that considers agile software delivery would cause the network security vulnerability to be closed, thus mitigating possible hazards.

The environmental stability of all grid systems is paramount. The tamper-resistant device can be used and integrated into grid segments to avoid unwanted physical entry. Remote exposure by unauthorized workers will lead to data stored, such as authorization, identity, use and account details in compromised computers [8]. Remote wiping technologies should be set up to uninstall or lock network resources to protect confidential private information from leaking because intruders can use them maliciously. It is crucial to strengthen the physical security of the facilities, where servers and control rooms are located [34]. This provides a focus area from which those intending to damage the network, such as hackers and disgruntled former staff, can easily control the entire IoT smart grid.

While designing IoT for smart grid applications, integrity and end security will remain the critical factor, since the IoT devices may be used for surveillance or law enforcement purposes. Nevertheless, it also can be exposed as a double-edged weapon, whereby it can be exploited by a terrorist [10]. Therefore, from the beginning, both manufacturers and users should ensure that no bypass or malicious code is installed on the smart grid application. They should also ensure that they do not mass-produce devices with a single arrangement of default logins and should not render specific logins for each model, because this provides an easy opportunity for DoS attacks on the devices.

DoS attacks are the IoT's most crucial challenge. Thus, a viable network layer programming approach is required to prevent DoS assaults [35]. This can be resolved by using the fast hopping Internet Protocol (IP). It allows consumers an easy way to cover their contact sessions' content and destination site. This is achieved by concealing a server's real IP address behind a large pool of IP, which ultimately impedes the detection of network traffic destination through various switches. The real-time shift in the server's IP address happens concurrently with all registered customers and applications.

5. PROPOSED COMPONENTS TO SECURE SMART GRID WITH IOT

From the discussion on the IoT smart grid security issues and the suggested mitigation control, this study proposes an IoT Smart Grid Security Strengthening model. This model focuses explicitly on strengthening because we believe that the initial security aspect has been implemented in both IoT and smart grid design. However, due to unexpected cybersecurity threat, specifically on IoT, the additional model will be beneficial to ensure security adherence. Figure 2 shows the proposed model discussed.

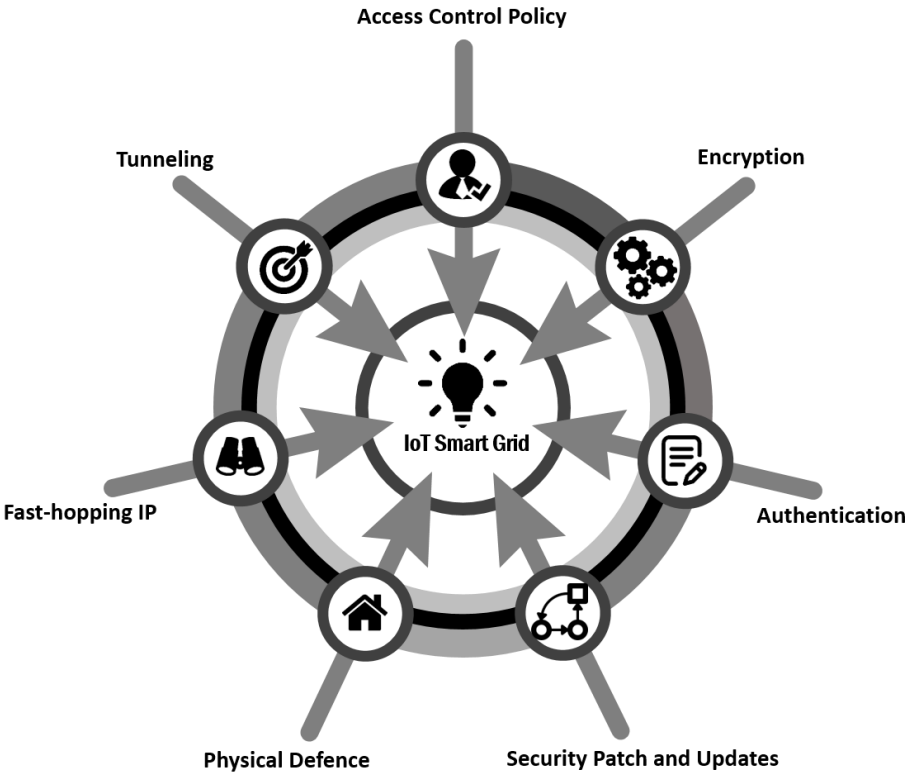


Figure 2. IoT smart grid security strengthening model.

There should be an ***access control policy*** in place in the environment. In that manner, we can avoid intruders from gaining physical or logical access to the critical infrastructure, besides encryption. As dangerous as the IoT network could get, state-of-the-art ***encryption*** on the communication is essential. For example, we could use MD5 or SHA 256 or 3DES to encrypt the data that is being stored. Password and credential policy plays a vital role. There must be an ***authentication*** mechanism in place for the systems to recognize and identify legitimate users. The authentication mechanism will then block intruders. This will not make it an easy process for intruders to get into the system. There must not be a natural or common password being used.

As soon as the ***security patch and updates*** are installed, they should be considered to be updated on the smart meter or IoT devices. Improper test on the security updates and pieces may break the existing equipment. It is highly suggested to test the development environment before implementing the production environment, which indirectly allows the closure of backdoors and loopholes in the system. Furthermore, there must not be a single loophole or backdoor in the critical infrastructure. The fundamental issue to consider is ***physical defence***. As we learn, smart grid SCADA and IoT are the most vital technologies. Environmental protection is essential and can be used as first-level security for an organization against intruders and natural disasters. ***Fast-hopping IP*** implements an innovative Internet security solution which should not be neglected. Thus, the network does not use static TCP, which helps protect the system and delay the attacker attempt. ***Tunnelling*** will be built to protect this vital infrastructure further; for example, the IPSEC tunnel or site-to-site tube. It will cover the smart grid network from man-in-the-middle-attacks and any other similar attacks.

Smart electronic devices need end-to-network systems worldwide to protect the smart grid from the control centre to the broadcast substations. This technology includes network-level monitoring systems, such as Home Access Network (HAN), Neighbourhood Access Network (NAN) and Family Access Network (FAN), with endpoint devices, such as smart meters or other Intelligent Electronic Devices (IEDs), substations and control centres. A smart grid access system allows for numerous networking advancements, such as Zigbee, Wimax and WI-Fi.

For example, HAN addresses specific mobile gadgets using the Zigbee protocol. Within a Zigbee configuration, Particular Zigbee provides different machine security arrangements. Contemplating introducing Zigbee security modules to handle Zigbee technology remains a flexible research point within the HAN. However, Zigbee specifications are meant for simplified activities, such as remote controls. Zigbee's partnership functions to make the NAN mesh network a standard. The network field can establish Wimax-dependent connectivity between remote devices and substations. Thus, configuring the similar networking systems used in the smart grid will ensure that the smart grid network works to the end of the security infrastructure.

Furthermore, the use of IPsec convention needs a commitment to incorporate an end-to-end smart grid security network. This case involves an IPsec investigation into Zigbee and Wimax. Zigbee has been designed for local networks, so web-based apps do not talk explicitly. Not just that; specifically, HAN and smart meters require Internet data transmission. The 6LowPAN allows sharing IPv6 packets to and from IEEE802.15.4-based systems. If 6LowPAN is used in the home area network, expanded security prerequisites must be addressed. Additionally, Wimax's IP-based protection for the FAN network must be investigated as an IPsec's configuration in the smart grid network, which may pose several problems because it has different specifications.

6. CONCLUSION AND FUTURE WORK

In this paper, we highlighted that securing the smart grid network is essential, as the data exchanged is sensitive, and the management operations are crucial. The smart grid is spread out in various spaces, as heterogeneous devices and systems are used in a distributed manner. We categorized types issues of security; firstly component security, secondly system security and thirdly network security. For component security, the smart meter is a vital element which is vulnerable to various forms of assaults, such as spoofing, eavesdropping, infusing false information and targeting replays. We have also featured that an intruder can spoof the smart meter's character to gain access to all home devices.

For system security and network security, DOS attacks are the frequent attack type that can influence control systems rendering devices inaccessible to network demands. Besides, there is also the man-in-the-middle attack on the AMI network and unique sending assault on conventions that separate an excellent hub that does not have the option to reach its neighbours and the control location. Therefore, this paper highlighted the essentiality in implementing an end-to-end security engineering, access control, physical security, frequent patches and updates, fast hopping IP and tunnelling to protect the smart grid.

The proposed model has the potential to identify threats accurately and can provide an extensive security measure when supplied with adequate IoT smart grid equipment. At this moment, this proposed model is still at the experimental scope, and it still requires much experimentation to distinguish an optimal

parameter with the real IoT smart grid evaluation. The IoT is capable of changing how we think of cities around the world. In order to improve and replace old architectures, IoT links people and governments to innovative urban solutions by inventing smart grid technologies. Corporations, utilities and private citizens that use grid power and thus benefit from the implementation of smart grid technologies by municipalities include all residents, urban services and critical infrastructures. The intelligent grid is stable, effective, green and good for customers, utility companies and the environment.

ACKNOWLEDGEMENTS

The authors thank the distinguished reviewers for reviewing this article. The research is financially supported by Universiti Teknologi Malaysia (UTM) Transdisciplinary Research (TDR) Grant Q.K130000.3556.06G26.

REFERENCES

[1] K. Kandasamy, S. Srinivas, K. Achuthan and V. P. Rangan, "IoT Cyber Risk: A Holistic Analysis of Cyber Risk Assessment Frameworks, Risk Vectors and Risk Ranking Process," EURASIP Journal on Information Security, vol. 2020, pp. 1-18, 2020.

[2] P. Sethi and S. R. Sarangi, "Internet of Things: Architectures, Protocols and Applications," Journal of Electrical and Computer Engineering, vol. 2017, pp. 1-25, 2017.

[3] B. Champaty, S. K. Nayak, G. Thakur, B. Mohapatra, D. Tibarewala and K. Pal, "Development of Bluetooth, Xbee and Wi-Fi-based Wireless Control Systems for Controlling Electric-Powered Robotic Vehicle Wheelchair Prototype," Robotic Systems: Concepts, Methodologies, Tools and Applications, IGI Global, pp. 1048-1079, 2020.

[4] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal and B. Sikdar, "A Survey on IoT Security: Application Areas, Security Threats and Solution Architectures," IEEE Access, vol. 7, pp. 82721-82743, 2019.

[5] S. K. Rathor and D. Saxena, "Energy Management System for Smart Grid: An Overview and Key Issues," International Journal of Energy Research, vol. 44, no. 6, pp. 4067-4109, 2020.

[6] X. Fang, S. Misra, G. Xue and D. Yang, "Smart Grid—The New and Improved Power Grid: A Survey," IEEE Communications Surveys & Tutorials, vol. 14, no. 4, pp. 944-980, 2011.

[7] S. K. Goudos, P. Sarigiannidis, P. I. Dallas and S. Kyriazakos, "Communication Protocols for the IoT-based Smart Grid," IoT for Smart Grids, pp. 55-83, Springer, 2019.

[8] F. Dalipi and S. Y. Yayilgan, "Security and Privacy Considerations for IoT Application on Smart Grids: Survey and Research Challenges," Proc. of the 4th IEEE International Conference on Future Internet of Things and Cloud Workshops (FiCloudW), pp. 63-68, Vienna, Austria, 2016.

[9] B. Jelacic, I. Lendak, S. Stoja, M. Stanojevic and D. Rosic, "Security Risk Assessment-based Cloud Migration Methodology for Smart Grid OT Services," Acta Polytechnica Hungarica, vol. 17, no. 5, pp. 113-134, 2020.

[10] M. Z. Gunduz and R. Das, "Cyber-security on the Smart Grid: Threats and Potential Solutions," Computer Networks, vol. 169, p. 107094, 2020.

[11] L. Tightiz and H. Yang, "A Comprehensive Review on IoT Protocols' Features in Smart Grid Communication," Energies, vol. 13, no. 11, p. 2762, 2020.

[12] A. Ghasempour, "Internet of Things in Smart Grid: Architecture, Applications, Services, Key Technologies and Challenges," Inventions, vol. 4, no. 1, p. 22, 2019.

[13] M. Faheem *et al.*, "Smart Grid Communication and Information Technologies in the Perspective of Industry 4.0: Opportunities and challenges," Computer Science Review, vol. 30, pp. 1-30, 2018.

[14] S. Eom and J.-H. Huh, "The Opening Capability for Security against Privacy Infringements in the Smart Grid Environment," Mathematics, vol. 6, no. 10, p. 202, 2018.

[15] S. Lee, H. Lim, W. Go, H. Park and T. Shon, "Logical Architecture of HAN-centric Smartgrid Model," Proc. of the IEEE Int. Conf. on Platform Technology and Service, 2015, pp. 41-42, Jeju, S. Korea, 2015.

[16] T. Alladi, V. Chamola and S. Zeadally, "Industrial Control Systems: Cyberattack Trends and Countermeasures," Computer Communications, vol. 155, pp.1-8, 2020.

[17] K. Kimani, V. Oduol and K. Langat, "Cybersecurity Challenges for IoT-based Smart Grid Networks," International Journal of Critical Infrastructure Protection, vol. 25, pp. 36-49, 2019.

[18] A. Ghosal and M. Conti, "Key Management Systems for Smart Grid Advanced Metering Infrastructure: A Survey," IEEE Communications, Surveys & Tutorials, vol. 21, no. 3, pp. 2831-2848, 2019.

[19] T. Alladi, V. Chamola, J. J. Rodrigues and S. A. Kozlov, "Blockchain in Smart Grids: A Review on Different Use Cases," Sensors, vol. 19, no. 22, p. 4862, 2019.

[20] G. Dileep, "A Survey on Smart Grid Technologies and Applications," Renewable Energy, vol. 146, pp. 2589-2625, 2020.

[21] N. S. Nafi, K. Ahmed, M. A. Gregory and M. Datta, "A Survey of Smart Grid Architectures, Applications, Benefits and Standardization," Journal of Network and Computer Applications, vol. 76, pp. 23-36, 2016.

[22] N. Nidhi, D. Prasad and V. Nath, "Different Aspects of Smart Grid: An Overview," Nanoelectronics,

Circuits and Communication Systems, Part of the Lecture Notes in Electrical Engineering, vol. 511, pp. 451-456, Springer, 2019.

[23] K. E. Mwangi, S. Masupe and J. Mandu, "Modelling Malware Propagation on the Internet of Things Using an Agent-based Approach on Complex Networks," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 6, no. 01, pp. 26-40, 2020.

[24] J. E. Sullivan and D. Kamensky, "How Cyber-attacks in Ukraine Show the Vulnerability of the US Power Grid," The Electricity Journal, vol. 30, no. 3, pp. 30-35, 2017.

[25] H. B. Salameh, M. Dhainat and E. Benkhelifa, "A Survey on Wireless Sensor Network-based IoT Designs for Gas Leakage Detection and Fire-fighting Applications," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 5, no. 02, pp. 60-72, 2019.

[26] E. Manavalan and K. Jayakrishna, "A Review of Internet of Things (IoT) Embedded Sustainable Supply Chain for Industry 4.0 Requirements," Computers & Industrial Engineering, vol. 127, pp. 925-953, 2019.

[27] P. A. Pegoraro, A. Meloni, L. Atzori, P. Castello and S. Sulis, "PMU-based Distribution System State Estimation with Adaptive Accuracy Exploiting Local Decision Metrics and IoT Paradigm," IEEE Transactions on Instrumentation and Measurement, vol. 66, no. 4, pp. 704-714, 2017.

[28] A. Meloni, P. A. Pegoraro, L. Atzori and S. Sulis, "An IoT Architecture for Wide-area Measurement Systems: A Virtualised PMU-based Approach," Proc. of the IEEE International Energy Conference (ENERGYCON), pp. 1-6, Leuven, Belgium, 2016.

[29] I. Almomani and K. Sundus, "The Impact of Mobility Models on the Performance of Authentication Services in Wireless Sensor Networks," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 06, no. 1, pp. 75-93, 2020.

[30] N. A. Bakar, W. M. W. Ramli and N. H. Hassan, "The Internet of Things in Healthcare: An Overview, Challenges and Model Plan for Security Risks Management Process," Indonesian Journal of Electrical Engineering and Computer Science (IJEECS), vol. 15, no. 1, pp. 414-420, 2019.

[31] N. Židková, M. Maryška, P. Doucek and L. Nedomova, "Security of Wi-Fi As a Key Factor for IoT," Hradec Economic Days, DOI: 10.36689/uhk/hed/2020-01-101, 2020.

[32] F. I. Salih, N. A. A. Bakar, N. H. Hassan, F. Yahya, N. Kama and J. Shah, "IoT Security Risk Management Model for Healthcare Industry," Malaysian J. of Comp. Science, vol. sp2019, no. 3.9, pp. 131-144, 2019.

[33] R. Mahmoud, T. Yousuf, F. Aloul and I. Zualkernan, "Internet of Things (IoT) Security: Current Status, Challenges and Prospective Measures," Proc. of the 10th IEEE International Conference for Internet Technology and Secured Transactions (ICITST), pp. 336-341, London, UK, 2015.

[34] J. Pacheco and S. Hariri, "IoT Security Framework for Smart Cyber Infrastructures," Proc. of the 1st IEEE International Workshops on Foundations and Applications of Self* Systems (FAS* W), pp. 242-247, Augsburg, Germany, 2016.

[35] T. Alladi, V. Chamola, B. Sikdar and K.-K. R. Choo, "Consumer IoT: Security Vulnerability Case Studies and Solutions," IEEE Consumer Electronics Magazine, vol. 9, no. 2, pp. 17-25, 2020.

"IoT Security for Smart Grid Environment: Issues and Solutions," Y. Velayutham, N. A. A. Bakar, N. H. Hassan and G. N. Samy.

ملخص البحث:

تعدّ إنترنت الأشياء آخر إبداعات الإنترنت هذه الأيام، وتعد البنية التحتية الأهم - الشبكة الذكية- النسخة الموسعة من شبكة القدرة، وهي ذات بنية إنترنت شاملة. وتحتوي الشبكة الذكية على بلايين الأجهزة الذكية؛ كالمقاييس الذكية، والمشغلات، والنواقل وغيرها، مع وجود القليل من البنى التحتية الخاصة بالتراسل.

وتجدر الإشارة الى أن الأمان ينظر اليه على أنه الشاغل الأبرز الذي يعيق الاستقبال واسع النطاق والترتيب المتعلق برؤية كلٍ من إنترنت الأشياء والشبكة الذكية.

ترمي هذه الورقة الى فحص المشكلات والتحديات المرتبطة بالأمان في أنظمة إنترنت الأشياء - الشبكة الذكية. وقد أسفرت الدراسة عن مسائل متعددة يمكننا تصنيفها في ثلاث فئات: مسائل متعلقة بالمكونات، وأخرى متعلقة بالأنظمة، وثالثة متعلقة بالشبكة. وبناءً على ذلك، يقترح الباحثون خطةً للتخفيف من المشكلات الخاضعة للدراسة؛ عبر تصميم أنموذجٍ للحفاظ على أمان انظمة إنترنت الأشياء - الشبكة الذكية.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 07, No. 01, March 2021.

EFFICIENT DEEP FEATURES LEARNING FOR VULNERABILITY DETECTION USING CHARACTER N-GRAM EMBEDDING

Mamdouh Alenezi[1], Mohammed Zagane[2] and Yasir Javed[1]

(Received: 19-Aug.-2020, Revised: 2-Oct.-2020 and 28-Oct.-2020, Accepted: 5-Nov.-2020)

ABSTRACT

Deep Learning (DL) techniques were successfully applied to solve challenging problems in the field of Natural Language Processing (NLP). Since source code and natural text share several similarities, it was possible to adopt text classification techniques, such as word embedding, to propose DL-based Automatic Vulnerabilities Prediction (AVP) approaches. Although the obtained results were interesting, they were not good enough compared to those obtained in NLP. In this paper, we propose an improved DL-based AVP approach based on the technique of character n-gram embedding. We evaluate the proposed approach for 4 types of vulnerabilities using a large c/c++ open-source codebase. The results show that our approach can yield a very excellent performance which outperforms the performances obtained by previous approaches.

KEYWORDS

Software security, Vulnerability detection, Deep features learning, Character N-gram embedding.

1. INTRODUCTION

Disastrous consequences related to exploiting software vulnerabilities can be avoided if these vulnerabilities are early detected and fixed before software deliverance. Many solutions to automatic vulnerabilities prediction (AVP) have been proposed. Manual vulnerable code detection is very hard and very costly, especially when dealing with software with a large codebase. These solutions aim to assist developers and minimize costs related to detection and fixing of vulnerabilities by letting them focus their effort and time on the components (files, classes or functions) that are most probable to be vulnerable. Researchers have proposed several approaches to develop vulnerability prediction models (VPMs) that are cable of discriminating vulnerable components from clean components. The most important works were to propose data-driven approaches based on using software attributes, such as software metrics with machine learning (ML) techniques to build VPMs. The major limitation of these approaches lies in the fact that important semantic and syntactic characteristics of the code that may give insight about vulnerabilities cannot be captured by using only static code attributes.

Motivated by the success of using deep learning (DL) techniques in other fields, such as natural language processing (NLP) and image processing, researchers in recent research works (see related work section) in the field of AVP begin to apply DL techniques to predict and locate vulnerabilities. Since source code shares several characteristics of the natural text and the same thing is valid for programming language and natural language (both have: vocabulary, syntactic and semantic characteristics, …etc), researchers have proposed to deal with source code written in a programming language like dealing with the natural text of a natural language. Therefore, techniques used in some applications of NLP, such as text classification, are adopted in the field of AVP to predict and locate vulnerabilities: classifying source code entities (file, function or slices) as vulnerable or clean (Figure 1). More specifically, the techniques, such as word embedding and bag-of-word used in NLP to automatically extract features from the natural text, are applied to automatically extract features from the source code. The automatically-extracted features are then used as input for a classifier based on machine learning (ML), which classifies source code as vulnerable or clean (Figure 1: solid lines). In DL-based approaches, the output of the first step of feature extraction (the input vectors) is passed to a deep neural network (DNN) to learn more hidden features (deep features). Since the important hidden

1. M. Alenezi and Y. Javed are with Department of Computer Science, Prince Sultan University, Riyadh, KSA. Emails: malenezi@psu.edu.sa, yjaved@psu.edu.sa
2. M. Zagane is with Department of Computer Science, University Mustapha Stambouli of Mascara, Mascara, Algeria. Email: mohamed.zaagane@univ-mascara.dz

features which become the actual classifier inputs are learned *via* the DNN, the first step of feature extraction (word embedding, bag-of-word, …etc) is considered in the DL-based approach as input vectorization.

Two main DL-based approaches are proposed (Figure 1: dashed lines). In the first approach, a DNN is used to deeply learn hidden features from the vectorized inputs and predict vulnerabilities (i.e., as a classifier), while in the second approach, a DNN is only used to learn hidden features which are then used as inputs (features) for an ML-based classifier that predicts vulnerabilities.

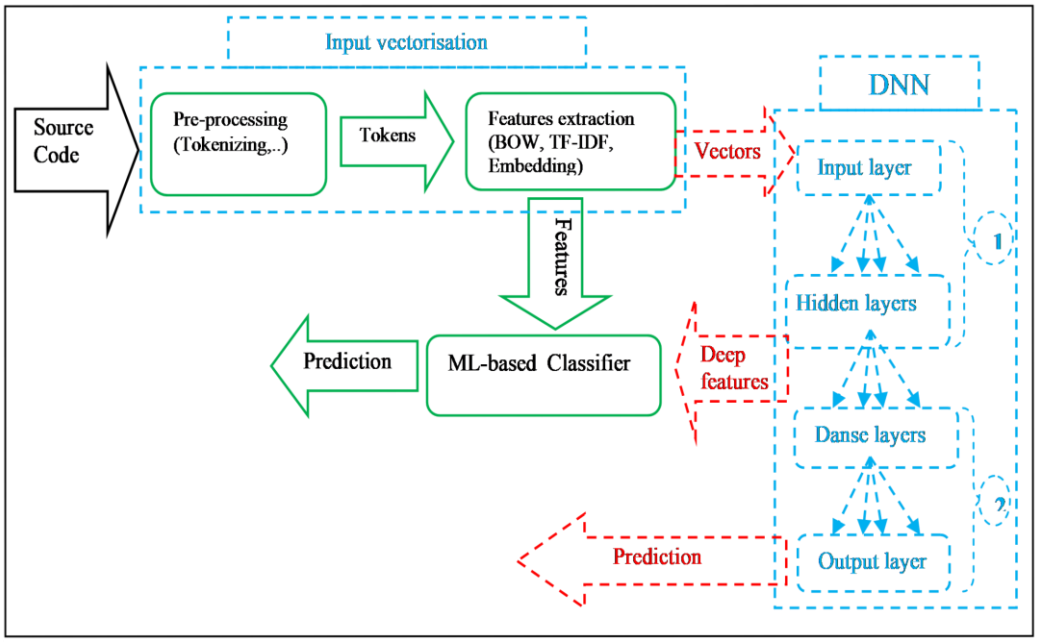


Figure 1. Vulnerabilities prediction approaches inspired by NLP techniques (dashed lines: DL-based approach, solid lines: ML-based approach) (1: learning deep hidden features, 2: classification).

Source code and natural text have main similarities that make it possible to adopt techniques used in NLP also in AVP. The most important adopted technique is the word embedding technique. The word or token embedding allows representing the words of a text in the form of vectors suitable to be processed by DNNs. The efficiency of this method compared to other methods, such as Bag-of-Word, is that it allows preserving semantic and syntactic information of words. On the other hand, there are characteristics which are specific to source code, making obtaining good performance very challenging. The most important characteristic is the large vocabulary and the rare words that can have source code. To address this problem, researchers proposed to apply vocabulary reduction methods. These methods allowed them to initiate using word embedding on the code, but at the expense of reduced performance. This represents a limitation, because these methods can cause a significant loss of valuable information related to vulnerabilities.

This research aims to address this limitation by proposing a code embedding solution that can be applied without reducing vocabulary, thus improving the vulnerability detection performance. The proposed approach lies in using N-gram-based embeddings at the character level. Compared with previous methods, the proposed embedding method can be applied without reducing vocabulary and enables the semantics of sub-tokens to be learned, which can avoid out-of-vocabulary tokens, thus reducing the possibility of information loss. Besides, we embed code at the slice granularity level, which allows the vulnerable code to be precisely identified.

The contribution of this work is two-fold:

- Proposing and evaluating an efficient and effective input vectorization approach based on the character n-gram embedding technique proposed by the Facebook research team [1]. The proposed approach allowed us to improve the performance of vulnerability detection.
- Proposing and making publically available a dataset generated following the proposed approach. This dataset can be used by other researchers in other research works or to train concrete vulnerability detection systems.

The remainder of this paper is organized as follows: in Section 2, we present the most relevant related works, while in Section 3, we describe the proposed approach and in Section 4, we present the experimental evaluation. In Section 5, we discuss the obtained results, while in Section 6, we highlight the limitations of the study and in Section 7, we summarize the work done in this study and indicate some perspectives for future works.

2. RELATED WORK

In this section, we present the most related works in the field of vulnerability prediction. To show the difference and the contribution of the recent DL-based approach, we begin by briefly presenting the previous ML/static-code-attributes-based approaches in the first sub-section, then in the second sub-section, we present DL/automatically-learned-features approaches. For the sake of brevity, we will focus only on the works that used the technique of word embedding to represent source code.

2.1 Traditional ML-based Approaches

Applying traditional ML techniques to predict software vulnerabilities has attracted the attention of several researchers. Indeed, considerable research works have been done to propose automatic vulnerability prediction (AVP) approaches based on machine learning (ML) and manually-defined static code features, such as software metrics ([2]–[9]) and text-based features [7], [10]. These works were motivated by the success of similar works [11]–[15] that have been done to predict software defects and by the fact that several code attributes, such as complexity, size and coupling (which can be quantified by corresponding software metrics), are proven in practice to be correlated to vulnerabilities. As reported in [16], the task of defining features is tedious, subjective and sometimes error-prone because of the complexity of the problem. This means that the quality of the resulting features and therefore the effectiveness of the resulting detection system varies with the individuals who define them. Another major drawback of these approaches lies in the fact that important semantic and syntactic characteristic of the code, which may give insight about vulnerabilities, cannot be captured by using only static code attributes. Another limitation of these approaches inherited by the coarse granularity level (file, class and method), in which software metrics are calculated, is that vulnerabilities cannot be located in much fine granularity. Recent works have tried to improve these approaches. Researchers in [9] have tried to combat the limitation of coarse granularity by proposing to calculate metrics at the slice granularity which allowed to improve the performance of the proposed VPMs (Precision: 95.1%, Recall: 95.0% FN Rate: 4.91%) and to locate with much precision the vulnerable lines. Other studies, such as [17]-[18], investigated using the automatically-learned features to build prediction models. However, the ML-based approaches still suffer from the missing semantic and syntactic features of the code and cannot learn deeply hidden features of the code which may exhibit a better way of characteristics of vulnerabilities. This is why in recent studies, researchers begin to use DL in AVP to benefit from the power of DL in learning hidden features. The most important of these studies are presented in the next sub-section.

2.2 DL-Based Approaches

DL techniques have been successfully applied to solve challenging problems in fields, such as NLP and image processing. Motivated by this success, researchers of the field of AVP in recent years begin investigating the application of DL techniques to predict and locate vulnerabilities in source code. The researchers' aim was essentially to benefit from the power of DNNs to learn deep hidden features that can perfectly characterize the vulnerable code, which was impossible using classic ML techniques, as well as to use them as a classifier (Figure 2).

Unlike the ML-based approach where several works have been carried out, few researchers have addressed AVP using DL techniques. The first use of DL in AVP was done by Catal et al. in [19]-[20]. In the first study, they conducted a literature review to investigate DL algorithms that can be applied in AVP. They concluded that, depending on the availability of the data, different kinds of DL algorithms can be applied in AVP: supervised learning models, unsupervised deep learning models or semi-supervised learning. In the second study, they proposed a web service-based VPM to predict vulnerable files of web applications. They used a dataset [21] proposed by [7] to train several machine learning techniques that exist in the Azure Machine Learning Studio environment and a Multi-Layer

Perceptron (MLP). They reported that the best performance (AUC: 76,5%) is achieved by the MLP. The type of VPMs' inputs was a set of code metrics. Researchers in [22] also used software metrics with DL to predict vulnerabilities. The authors investigated the usefulness of using software metrics as input for DNNs to locate vulnerabilities. Researchers reported that they used a large dataset [23] suitable for DL. The code metrics used as inputs for the DNNs (MLP and LSTM) were calculated at the slice ([24]–[26]) granularity level which allows them to locate the vulnerable lines of code. Based on comparing the obtained results (Recall: 73.9%, Precision: 74.4% and FN Rate: 26.14%) with the results reported by similar works that adopted techniques used in the field of NLP, the authors concluded that software metrics represent good -but not the best- data to use with DL-based approaches in AVP and that software metrics are more suitable for ML-based approaches which gave them very good results (Recall: 93.7%, Precision: 93.2% and FN Rate: 6.25%).

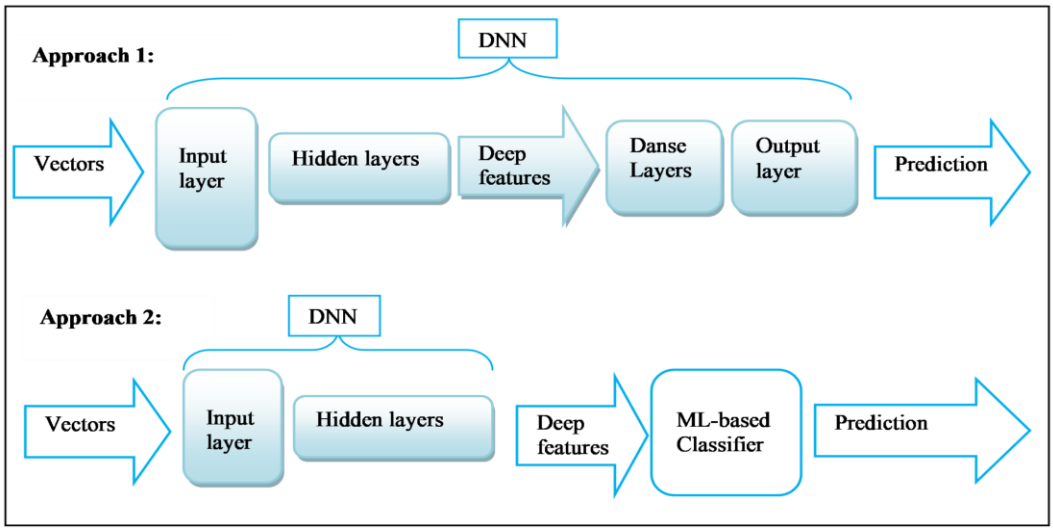


Figure 2. Using DNNs to predict vulnerabilities.

As we said before, the similarities between source code and the natural text have motivated researchers in the field of AVP to adopt techniques used in NLP to predict vulnerabilities. Essentially, techniques, such as word or token embedding used in NLP to "vectorize" inputs (representing text as vectors suitable to be used as inputs for DNNs), were adopted by recent works [16], [18], [27]–[30] to represent source code as vectors. The DNNs are used to learn from the vectorized inputs deep hidden features of the code that are related to vulnerabilities (Figure 1). Z. Li et al. in their works [16], [28]-[29] used the word2vec tool [31]-[32] which is based on using NNs to learn a vector representation of the word that preserves its semantic meaning based on its context, starting from the idea that words with similar meanings will tend to appear in contexts with similar words. Researchers in [18], [27] used custom embedding techniques inspired by previous works done on sentence classification, such as [33].

In all of these studies, token embedding was at the token level. This means that a distinct vector is assigned to each token in the training set. For an embedding model to be efficient and effective, it must provide representation for all or at least most of the words that compose the vocabularies. In AVP, this represents a challenging problem to solve, because source code may have very large vocabularies and many rare words induced by the fact that ways of writing code, especially the task of naming variables and functions, vary from developer to developer. To combat this problem, researchers used techniques to reduce vocabulary size by mapping user-defined variables and functions and all literal values (number and string) to special tokens. For example, in [18], all integers, real numbers, exponential notation and hexadecimal numbers are replaced with a generic <num> token and constant strings are replaced with a generic <str> token. Also, all rare tokens (e.g. occurring only once in the corpus) and tokens which exist in test sets but do not exist in the training set are replaced with a special token <unk>. In [16], [28]-[29], all user-defined variables and functions were mapped to representations, such as VAR1, VAR2, FUN1, FUN2, …etc.

Using these techniques, researchers were able to reduce the vocabularies size and partially benefit from the power of token embedding. However, these techniques of reducing vocabularies may lead to

a very important loss of information by abstracting away certain syntactic and semantic characteristics of the code that are useful for vulnerability detection, which represents a limitation. To the best of our knowledge, no recent work has addressed this limitation. Instead, in recent works, researchers tried to address other aspects of DL-based AVP. In [30], researchers studied the cross-domain AVP. They proposed and evaluated a method to learn cross-domain representations in a range of cross-domain settings, including cross-project, cross-vulnerability and prediction of recent software vulnerabilities. Researchers in [34] addressed the problem of class imbalance between vulnerable code and non-vulnerable code. A new fuzzy oversampling method is proposed to rebalance the training data. In [35], both cross-project and class imbalance problems were studied.

To fill the research gap highlighted in the previous paragraph, we propose in this study an improved DL-based approach to detect vulnerabilities. Instead of reducing vocabularies and using token-level embedding, we propose an approach based on the works [1], [36] done by Facebook AI Research. The proposed approach is presented in detail in the next section.

3. PROPOSED APPROACH

The proposed approach is adopted from the communally used approach described in Figure1 (dashed lines) which was inspired by the previous works in the field of NLP. In our approach, DNNs are used to learn deep hidden features and as a classifier Figure 2 (approach 1). The different aspects of the proposed approaches: granularity level and input vectorization, are described in the following sub-sections.

3.1 Granularity Level

In previous studies of AVP, whether ML-based or DL-based, researchers investigated vulnerabilities prediction at different levels of granularity: file [7]-[8], [18], function/method [27], [37] and slice [9], [16], [22], [28]-[29]. Prediction at a coarse level (file and function) does not locate the vulnerable lines of code; instead, it can identify the components (files or functions) that require more focus from developers, which is less useful especially when the components are very large. Because the objective of the AVP is to assist developers and minimize the costs of vulnerabilities detection by minimizing the human intervention as much as possible, these coarse granularity levels are to be avoided.

A slice is a reduced version (few lines of code) of a source component automatically-extracted from the original component by analyzing its data flow and control flow in respecting a slicing criterion [25]. Slicing is useful in several software engineering applications, such as debugging, program comprehension and change impact prediction because it can give insight into multiple behavioral aspects of the source entity, such as all lines that change the value of a variable or that participate in computing the return value of a function [38]. In AVP, this can be useful, for example, to get all statements that are related to critical function calls (memory management, string manipulation, …etc.).

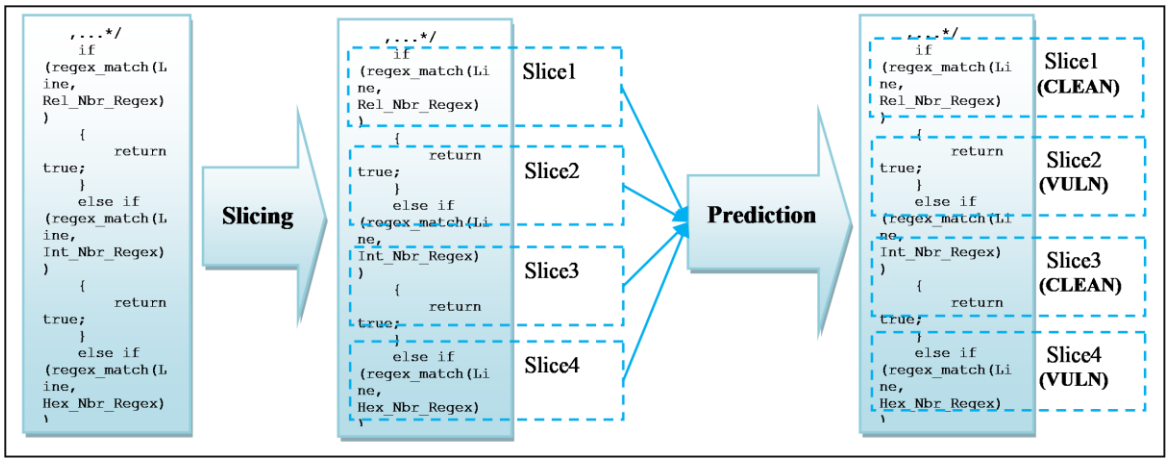


Figure 3. The adopted granularity level.

This way, only lines of code that are related to vulnerabilities can be extracted and analysed, which leads to indicate the exact location of vulnerabilities [22].

Currently, the slice level is the finest level of granularity to locate vulnerabilities. Predicting the status of a slice (clean or vulnerable) is just like locating the vulnerable lines (that the vulnerable slices contain) (Figure 3). Therefore, the slice level of granularity is adopted in this study.

3.2 Input Vectorization

As shown in Figure 3, a detection system built based on our proposed approach will take as input the source code of a software component (s) (file(s) or function(s)) and give in the output the vulnerable lines (those lines that compose the slices predicted as vulnerable). Since the prediction (deep features learning + classification) is made *via* DNNs, the source code of each extracted slice must be converted into vectors suitable to be used as input for DNNs (Figure 4, solid lines).

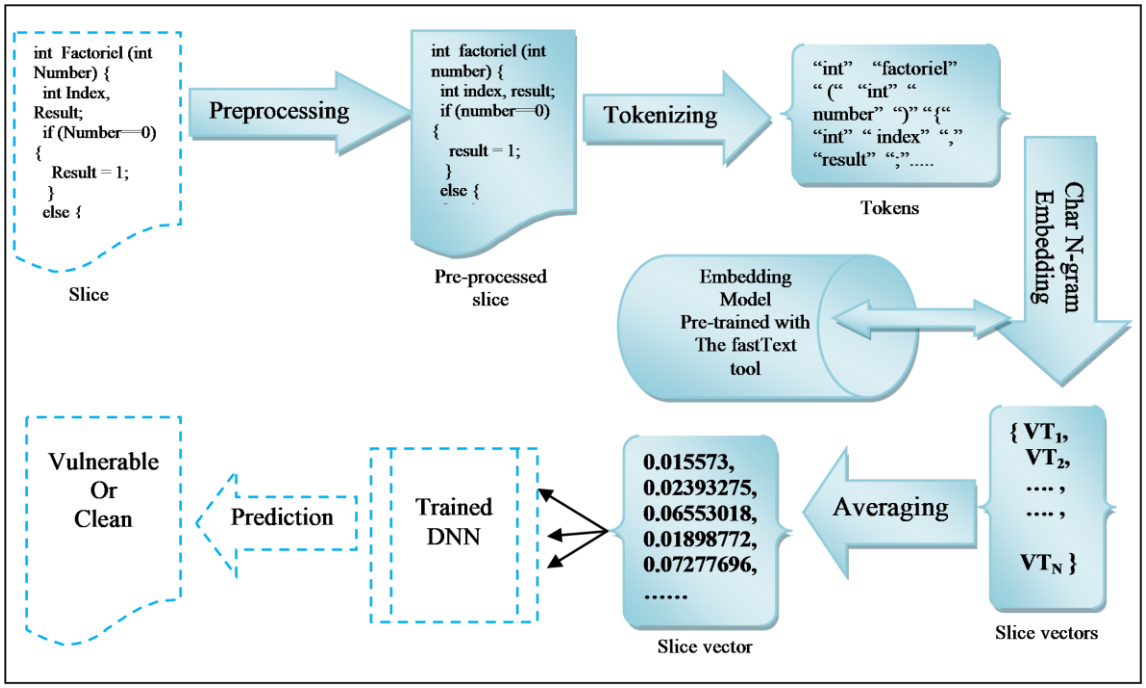


Figure 4. The proposed input vectorization approach (VTi means the vector representation of the i^{th} token).

We aim to benefit from the power of embedding techniques to vectorize source code without losing useful semantic and syntactic information related to vulnerabilities induced by reducing vocabularies techniques. To achieve this aim, instead of reducing the vocabularies and applying token-level embedding as it was done in previous works (see related works section), we adopted solution [1], [36] proposed by the NLP community to deal with the embedding task in languages with large vocabularies that contain many rare words which is the case with source code. The strength of the solution proposed by [1] lies in using character n-gram-based embedding, which means that every token is embedded using all its character n-gram (sub-tokens). This has two advantages:

- Avoiding vocabulary reduction, because it is possible to embed almost any token using vectors of its sub-tokens.
- Enriching semantic information of tokens by the information of their sub-tokens.

In our approach, a very light pre-processing (removing comments and lowercasing the code) is made on the source code before transforming it into a set of tokens. Then, all the tokens are embedded using a pre-trained embedding model which gives a set of vectors. Because on one hand, the DNNs take as input vectors with fixed size and on the other hand slices may contain a variable number of tokens, we relied on the principle of sentence embedding to get a fixed-length that represents the slice (which is considered as a sentence). This can be done using a very simple approach, such as summing or averaging all token vectors of the sentence [39] or with sophisticated approaches, such as the approach proposed in [40]. We applied the simple approach by averaging all the token vectors of a slice to get its vector representation.

4. EXPERIMENTAL EVALUATION

In this section, we present the experiments conducted to evaluate our proposed approach. We begin by presenting the data preparation in 4.1 and then in 4.2, we present the implementation and architecture of the DNNs and finish by presenting the performance evaluation.

4.1 Data Preparation

To carry out the experiments, we prepared a dataset generated following the steps of the proposed input vectorization approach (Figure 4). The prepared dataset is then used to train and validate the DNNs to detect vulnerabilities. Before presenting our dataset, we begin by briefly describing the original dataset from which we retained the labeled source codes used to generate our dataset.

4.1.1 Slices Dataset

The labeled source codes from which we generated the dataset are retained from a dataset proposed by [28] which is publically available in [41]. The original dataset contains 420627 labeled slices, including 56395 vulnerable slices and 364232 clean slices. The slices were generated based on 1591 open-source C/C++ programs from the National Vulnerability Database (NVD) and 14,000 programs from the Software Assurance Reference Dataset (SARD). For the sake of brevity, the process of extracting and labeling the slices will not be described here. For more information about that process, see the slices dataset in [28]. We have chosen this dataset because it meets our needs:

- The source codes are organized inside the dataset in the form of labeled slices.
- Several types of vulnerabilities are considered (Library/API function call, array usage, pointer usage and arithmetic expression-related vulnerabilities).
- The dataset is very large, which makes it very suitable for DL techniques.
- On the contrary to other proposed datasets, the labeling process is based on real vulnerability reports mined from famous vulnerability databases, such as NVD, which is more efficient than using static analyzer tools to label source codes which can lead to mislabeling the source entities, because these tools can make high false-positive/negative reports.

4.1.2 Embedding Model Preparation

The proposed dataset contains two parts. The first part (vector dataset), which is used to train and validate the proposed DL-based VPMs, contains the vector representations and the class labels (Vulnerable/Clean) of each slice from the slices that exist in the original dataset. As shown in Figure 4, the vector representations are calculated based on the pre-trained embedding model. This model is prepared and trained using the second part of the dataset.

The data used to train the embedding model contains the tokenized slices (the tokens) without class labels. To prepare this data, we developed a C++ parser to parse the original dataset and do the following steps :

1. Extracting the slice.
2. Applying a minimalist pre-processing (removing comments if any).
3. Tokenizing the slice.
4. Composing a dataset line by gathering the tokens (separated by spaces) and adding it to the dataset.
5. Repeating the above steps for all the slices of the original dataset.

Using this data and the fastText tool [1], [36], we built the embedding model. The most important parameters of the model are its dimension (dim) and the range of size for the subwords (the minimum character n-grams size (minn) and the maximal character n-grams size (maxn)). The dimension controls the size of the vectors, where the larger they are, the more information they can capture, but the model requires more data to be learned. The range of size for the subwords controls the character n-grams that can be extracted from the tokens [42]. Building the embedding model with the defaults recommends that the values of these two parameters (dim:100, minn:3 and maxn:6) were very sufficient (see results section). However, since the size of the vectors also controls the architecture of

DNNs (the number of neurons in the input layer = the size of the vectors), we built two other models with the size of the vectors of 50 and 200. The other parameters related to training the models were also set to their default recommended values: 5 for the number of epochs and 0.05 for the learning rate.

4.1.3 Vectors Dataset (The Proposed Dataset)

To predict the status (vulnerable or clean) of a slice, its vector representation is extracted and passed to the trained DNN which learns more deep hidden features from the input vector and classifies them (Figure 4). To train the DNN, we prepared a labeled dataset that contains all the vector representations of the slices which exist in the original dataset with their class label (vulnerable or clean). As we mentioned before, the vectors are calculated based on the built embedding model. We developed a Java application to generate the dataset. The application is based on the Java implementation of the fastText API provided by the famous deep learning library Deeplearning4J [43]. The dataset contains a total number of 420627 instances, including 56395 instances with the 'vulnerable' class label and 364232 instances with the 'clean' class label. Detailed descriptive statistics by each type of vulnerabilities can be observed in Table 1.

Table 1. Descriptive statistics about the proposed dataset.

Type of vulnerabilities	Number of instances with class label (vulnerable)	Number of instances with class label (clean)	Total
Part1: Function Call (FC)	13603	50800	64403
Part2: Array Usage (AU)	10926	31303	42229
Part3: Pointer Usage (PU)	28391	263450	291841
Part4: Arithmetic Expression (AE)	3475	18679	22154
Total	56395	364232	420627

The embedding models and all the tools developed and used to generate them are made publically available in the public Github repository [44] for researchers who may want to replicate the study or to use it in other works.

4.2 DL-based VPM Construction and Evaluation

4.2.1 DL-based VPM Construction

We used the implementation of the Multi-Layer Perceptron (MLP) provided by the Java API of the WekaDeeplearning4J package [45] to construct our DL-based VPMs. This package provides a rich API inherited from the Weka API [46] and the Deeplearning4J library that facilitate not only the construction and the validation of a prediction model, but also the deployment of the built models in a concrete Java application to be used in production.

The effectiveness of the deep hidden features learning and consequently the overall prediction performance of the DL-based VPMs can be affected by several parameters related to the used DNN architecture: number of hidden layers, number of neurons in each layer, …etc. and the parameters related to the training process: learning rate, number of epochs, …etc. We considered all these parameters when conducting experiments. Some of these parameters, such as the learning rate, were set to their recommended values based on previous similar studies and what experts in the field of DL recommend. Other parameters, such as those related to the DNN architecture, were tuned experimentally. For example, when tuning the number and the size (in terms of the number of neurons) of the hidden layers, we started with simple architecture and each time we increased the complexity of the architecture, we observed the obtained results until we got the best performance.

4.2.2 VPM Evaluation

To accurately evaluate our DL-based VPMs and avoid the possibility of obtaining biased results, we used the technique of K-fold cross-validation to train and validate them. Using this technique, the dataset is randomly divided into K folds of equal sizes. K-1 folds are retained and used as the training

set and the remaining fold is used as the testing set. This process is repeated such that all folds are used as the testing set and also as part of the training set. The final performance results are then calculated by averaging the results of all the iterations. Because our dataset is large enough, we set the value of K to 3. This allowed us to get very good results and reduce computation time.

A perfect VPM must have the following features:

- The VPM must predict as vulnerable only the actually vulnerable source entities. If this feature is not sufficiently achieved (i.e., the model leverages high false-positive predictions), the costs of vulnerability detection will not be minimized, since developers will still waste time and effort in looking for vulnerabilities in non-vulnerable source entities. This characteristic can be measured by the metric of the False Positives Rate (FPR).
- The VPM must not miss any vulnerability. This characteristic is very important, because if it is altered, vulnerabilities will be delivered with the software and can be exploited, which can lead to disastrous security issues. This characteristic can be measured by the metric of the False Negatives Rate (FNR).
- The VPM must make a precise and effective prediction. This characteristic can be measured by metrics, such as Precision or Recall.

Since obtaining a perfect VPM (FPR=0%, FNR=0% and Precision=100%) is impossible in practice, the objective is to minimize as much as possible the FPR and the FNR and to maximize as much as possible Precision. These metrics can be calculated from the outputs of the VPM: True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). The descriptions and the formulae of these metrics are shown in Table 2. For the sake of completeness, we also considered reporting in the results section, the obtained performance in terms of additional performance metrics that are communally reported in related works.

Table 2. The performance metrics.

Metric	Formula	Description
Precision	$\frac{TP}{TP+FP}*100$	The percentage of instances classified as positives and are actually positives.
FP Rate	$\frac{FP}{FP+TN}*100$	The percentage of positives that are falsely classified as real negatives.
FN Rate	$\frac{FN}{FN+TP}*100$	The percentage of negatives that are falsely classified as real positives.

5. RESULTS AND DISCUSSION

In Table 3, we report the obtained results in terms of the main performance metrics which we based our work on, in order to draw the conclusions: Precision (the higher the better), FPR and FNR (the lower the better). For the sake of completeness and for comparing the obtained results with the reported results in similar works, we report in Table 4 the obtained results in terms of additional performance metrics: Recall and F1 (the higher the better).

These results were obtained using a Multi-Layer Perceptron with the following architecture: 1 input layer with 100 neurons, 3 hidden layers with 128, 64 and 32 neurons and an output layer with 2 neurons. The MLP was trained and validated using the proposed dataset and 3-fold cross-validation. The parameters related to the training algorithm were set as follows (learning rate: 0,01, batch size:128) and the other parameters were set to their default recommended values (more information about these defaults recommended values can be found in the documentation of the Java API of Wekadeeplearning4j [47]). When carrying out the experiments, we begin by using the first part of the dataset which is related to the FC vulnerabilities to tune experimentally the DNN's hyperparameter related to the number of epochs to train through and the size of the vectors for the dataset. We used only the FC part due to computational constraints. For the sake of showing the impact of these parameters on the VPM performance, we report results for different values of these parameters in the first rows of Table 3 and Table 4 for the number of epochs and in Table 6 for the size of the vectors.

Table 3. Results.

Vulnerabilities	No. of epochs	Precision (%)	FP Rate (%)	FN Rate (%)
FC	100	95,1	13,25	4,84
	500	96,8	8,30	3,12
	1000	**97,1**	**7,66**	**2,87**
AE	1000	98,0	7,48	1,93
AU	1000	97,1	6,36	2,86
PU	1000	**98,6**	**5,51**	**1,44**

Table 4. Results in terms of additional performance indicators.

Vulnerabilities	No. of epochs	F1 (%)	Recall (%)
FC	100	95,06	95,2
	500	96,84	96,9
	1000	**97,1**	**97,1**
AE	1000	**98.04**	**98,1**
AU	1000	97,1	97,1
PU	1000	98,55	98,6

As can be seen in Table 3 and Table 4, the obtained performances in terms of all the performance metrics (whether the main ones or the additional ones) and for all the studied types of vulnerabilities (FC, AE, AU and PU) are very excellent. Indeed, the obtained precision was between 97,1% and 98,6%, the FNR was between 2,87% and 1,44% and the FPR was between 7,66% and 5,51%, which is very promising and outperformed the obtained performances in the previous works that used the same original dataset and the same granularity level (slice) ([16], [29]) and others ([18], [27]) that used different datasets and different granularity levels (Table 5). We believe that this performance improvement is due to the two strengths of the proposed input vectorization approach, which were missing in the previous approaches. The first is embedding the source code without reducing the vocabulary, which led to preserving all semantic and syntactic information of the source code related to vulnerabilities. The second is that this information is enriched by embedding source code in character n-gram level, because each token is embedded using its sub-tokens. For example, considering a token named “BufferSize”, using the technique of character n-gram embedding, this token will be embedded using all its character n-gram, including “Buffer” and “Size”, which means that two important things are granted. The first is that even if the token “BufferSize” does not exist in the training set, it will be possible to get its vector representation from its sub-tokens vector representations. The second is that the overall semantic meaning of the original token “BufferSize” will be enriched by the semantic meaning of its sub-tokens, including “Buffer” and “Size”.

We observed that the obtained values in terms of FPR were slightly higher when compared to the obtained values in terms of FNR (5,51% vs 1,44%). We confirm what other researchers [29] concluded about this situation. Indeed, improving the performance of a VPM in terms of one of these two metrics can affect its performance in terms of the other. As we mentioned before and as it is clear in Table 6, our obtained results outperformed the reported results of all the previous studies in terms of all performance metrics, except in terms of FPR. While we obtained in the best case 5,51%, researchers in [29] reported a value of 1,4%. The reported value is better than what we got, but it comes at the cost of much higher FNR (5,6%) than what we got (1,44%). In vulnerability prediction, the negative impact of FNR is much important than the negative impact of FPR. This is because FPR impacts the effectiveness of the model in terms of the cost of detection, while FNR impacts its effectiveness in the side of letting vulnerabilities undiscovered, which is very dangerous. Therefore, we believe that the advantages in terms of a lower FNR far outweigh the disadvantages concerning the slightly higher FPR.

We observed also that all the best values for all the experimental cases were obtained using the part of the dataset that is related to PU vulnerabilities. Since this part is the larger one in the dataset (see Table1), we believe that this was due to the sufficient amount of data that this part contains, which let the DNN learn more efficiently than with the other parts. This lets us conclude that sufficient labeled data is very important when using the DL technique to predict vulnerabilities.

Table 5. Comparison with the reported results of previous studies.

Study	Granularity level	Vocabulary reducing	Vectorization / Technique	Performances (%)
[16]	Slice	yes	word2vec-based techniques / (Token level)	-FPR: 5,7 -FNR: 7,0 -P: 88,1 -R: / -F1: 90,5
[29]	Slice	yes	word2vec-based techniques / (Token level)	**-FPR: 1,4** -FNR: 5,6 -P: 90,8 -R: / -F1: 92,6
[27]	Function	yes	Custom embedding technique / (Token level)	-FPR: / -FNR: / -P: / -R: / -F1: 84,0
[18]	File	yes	Custom embedding technique / (Token level)	-FPR: / -FNR:/ -P: 92,0 -R:93,0 -F1 :91,0
Our Study	**Slice**	**no**	**fastText-based embedding / (Character n-gram level)**	-FPR : 5,51 **-FNR :1,44** **-P :98,6** **-R :98,6** **-F1 : 98,55**

Table 6. Results for different vector sizes.

Vector size	Precision (%)	FP Rate (%)	FN Rate (%)	Recall (%)	F1 (%)
50	96,3	09,37	03,68	96,3	96,27
100	97,1	7,66	2,87	97,1	97,1
200	**97,3**	**07,49**	**02,67**	**97,3**	**97,29**

To investigate the impact of the size of the vectors on the VPM performances, we pre-trained 3 embedding models with 3 different values of the vector size parameter: 50,100 and 200. Then, we used the pre-trained models to prepare 3 datasets. The prepared datasets were then used to train and validate 3 VPMs. The obtained results are shown in Table 6. As can be seen, there were no significant differences between the 3 VPMs performances. This can be interpreted by the fact that models with higher vectors size (>200) need very big training data to capture much useful vector representation. Finally, we can conclude that the difference in performances (Precision: +0.2%, FPR: -0,17%, FNR: -0,2%) obtained by increasing the vector size to 200 is not worth the constrains in terms of increasing the model size induced by increasing the size of the vectors. That way in this study, we opted for 100 as the size of the vectors.

5.1 Limitations

We are aware that our work may have the following limitations:

- Since the original dataset from which we generated the data used to evaluate the proposed approach is limited to C/C++ code, we cannot conclude that the approach is suitable for other types of applications that are written in other languages. Evaluating the proposed approach for these types of applications represents an intersecting open problem for future research works.
- As the focus of the study was on proposing an input vectorization approach, we used only one DNN model, the MLP. Even though the obtained results using this model were very sufficient, other DNN models, such as RNN and CNN, must be considered in future works.
- We evaluated our approach for 4 types of vulnerabilities using binary classification. A multiclass classification will be a better choice for future works.
- More types of vulnerabilities must be considered in future works.

6. CONCLUSIONS

This paper investigated predicting vulnerabilities using the technique of deep learning. We aimed at improving the communally adopted approach in recent similar works which was inspired by the previous application of DL in NLP. Our contribution was to propose an efficient input vectorization approach based on embedding source code in the character n-gram level. Using the proposed approach with DNNs, we were able to efficiently learn deep hidden features and detect vulnerabilities with much better accuracy. The strengths of our method lie in preserving and enriching semantic and syntactic information related to vulnerabilities that can be extracted from the code. Indeed, the achieved performance outperformed those obtained in previous similar works, which means that our method represents a valuable alternative to the input vectorization methods used in previous works.

As part of this research, we proposed a dataset extracted from a labeled and large C/C++ codebase. The dataset was prepared based on the proposed input vectorization approach. We make it with other important data publically available for the community. As part of the future works and since the results obtained have been very promising, we plan to implement the proposed approach in a concrete solution that can be used in production. The solution can be in the form of a standalone AVP tool or an IDE plug-in. Improving further the proposed approach by addressing the limitations and the open research problems indicated in the previous section and considering more code structure for learning more comprehensive program semantics that is suitable for AVP, represent subjects for interesting future works a well.

ACKNOWLEDGEMENTS

The authors would like to thank the anonymous reviewers for their insightful comments and suggestions that helped in improving the paper.

REFERENCES

[1] P. Bojanowski, E. Grave, A. Joulin and T. Mikolov, "Enriching Word Vectors with Subword Information," Trans. Assoc. Comput. Linguist., vol. 5, pp. 135–146, DOI: 10.1162/tacl_a_00051, 2017.

[2] Y. Shin, A. Meneely, L. Williams and J. A. Osborne, "Evaluating Complexity, Code Churn and Developer Activity Metrics As Indicators of Software Vulnerabilities," IEEE Trans. Softw. Eng., vol. 37, no. 6, pp. 772–787, DOI: 10.1109/TSE.2010.81, 2011.

[3] T. Zimmermann, N. Nagappan and L. Williams, "Searching for a Needle in a Haystack: Predicting Security Vulnerabilities for Windows Vista," Proc. of the 3rd Int. Conf. on Software Testing, Verification and Validation (ICST 2010), pp. 421–428, DOI: 10.1109/ICST.2010.32, Paris, France, 2010.

[4] P. Morrison, K. Herzig, B. Murphy and L. Williams, "Challenges with Applying Vulnerability Prediction Models," Proceedings of the 2015 Symposium and Bootcamp on the Science of Security (HotSoS '15), pp. 1–9, DOI: 10.1145/2746194.2746198, 2015.

[5] S. Moshtari and A. Sami, "Evaluating and Comparing Complexity, Coupling and a New Proposed Set of Coupling Metrics in Cross-project Vulnerability Prediction," Proceedings of the 31st Annual ACM Symposium on Applied Computing (SAC '16), pp. 1415–1421, DOI: 10.1145/2851613.2851777, 2016.

[6] I. Abunadi and M. Alenezi, "Towards Cross Project Vulnerability Prediction in Open Source Web Applications," Proceedings of the the International Conference on Engineering & MIS 2015 (ICEMIS '15), pp. 1–5, DOI: 10.1145/2832987.2833051, 2015.

[7] J. Walden, J. Stuckman and R. Scandariato, "Predicting Vulnerable Components: Software Metrics *vs.* Text Mining," Proc. of the 25th IEEE International Symposium on Software Reliability Engineering (ISSRE), pp. 23–33, DOI: 10.1109/ISSRE.2014.32, Naples, Italy, 2014.

[8] M. Zagane and M. K. Abdi, "Evaluating and Comparing Size, Complexity and Coupling Metrics As Web Applications' Vulnerabilities Predictors," Int. J. Inf. Technol. Comput. Sci., vol. 11, no. 7, pp. 35–42, DOI: 10.5815/ijitcs.2019.07.05, 2019.

[9] M. Zagane, M. K. Abdi and M. Alenezi, "A New Approach to Locate Software Vulnerabilities Using Code Metrics," Int. J. Softw. Innov., vol. 8, no. 3, pp. 82–95, DOI: 10.4018/IJSI.2020070106, Jul. 2020.

[10] A. Hovsepyan, R. Scandariato, W. Joosen and J. Walden, "Software Vulnerability Prediction Using Text Analysis Techniques," Proceedings of the 4th International Workshop on Security Measurements and Metrics (MetriSec '12), p. 7, DOI: 10.1145/2372225.2372230, 2012.

[11] B. Turhan and A. Bener, "A Multivariate Analysis of Static Code Attributes for Defect Prediction," Proceedings of the 7th IEEE International Conference on Quality Software (QSIC 2007), pp. 231–237, DOI: 10.1109/QSIC.2007.4385500, 2007.

[12] H. Abandah and I. Alsmadi, "Call Graph Based Metrics to Evaluate Software Design Quality," Int. J. Softw. Eng. and Its Appl., vol. 7, no. 1, pp. 1–12, 2013.

[13] T. Hall, S. Beecham, D. Bowes, D. Gray and S. Counsell, "A Systematic Literature Review on Fault Prediction Performance in Software Engineering," IEEE Transactions on Software Engineering, vol. 38, no. 6. pp. 1276–1304, DOI: 10.1109/TSE.2011.103, 2012.

[14] B. Turhan, A. Bener and T. Menzies, "Nearest Neighbor Sampling for Cross Company Defect Predictors," Proceedings of the 1st International Workshop on Defects in Large Software Systems (DEFECTS'08), p. 26, DOI: 10.1145/1390817.1390824, 2008.

[15] T. Menzies, J. Greenwald and A. Frank, "Data Mining Static Code Attributes to Learn Defect Predictors," IEEE Trans. Softw. Eng., vol. 33, no. 1, pp. 2–14, DOI: 10.1109/TSE.2007.10, 2007.

[16] Z. Li et al., "VulDeePecker: A Deep Learning-based System for Vulnerability Detection," Proceedings of Network and Distributed System Security Symposium, DOI: 10.14722/ndss.2018.23158, 2018.

[17] T. Shippey, D. Bowes and T. Hall, "Automatically Identifying Code Features for Software Defect Prediction: Using AST N-grams," Inf. Softw. Technol., vol. 106, pp. 142–160, DOI: 10.1016/j.infsof.2018.10.001, Feb. 2019.

[18] H. K. Dam, T. Tran, T. T. M. Pham, S. W. Ng, J. Grundy and A. Ghose, "Automatic Feature Learning for Predicting Vulnerable Software Components," IEEE Trans. Softw. Eng., pp. 1–1, DOI: 10.1109/TSE.2018.2881961, 2019.

[19] C. Catal, "Can We Predict Software Vulnerability with Deep Neural Network ?" Proc. of the 19th Int. Multiconference INFORMATION SOCIETY- IS, no. October, pp. 19–22, Ljubljana, Slovenia, 2016.

[20] C. Catal, A. Akbulut, E. Ekenoglu and M. Alemdaroglu, "Development of a Software Vulnerability Prediction Web Service Based on Artificial Neural Networks," Proc. of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 59–67, DOI: 10.1007/978-3-319-67274-8_6, 2017.

[21] J. Walden, J. Stuckman and R. Scandariato, "Web Apps Vulnerability Dataset," [Online], Available: http://seam.cs.umd.edu/webvuldata, 2014.

[22] M. Zagane, M. K. Abdi and M. Alenezi, "Deep Learning for Software Vulnerabilities Detection Using Code Metrics," IEEE Access, vol. 8, pp. 74562–74570, DOI: 10.1109/ACCESS.2020.2988557, 2020.

[23] M. Zagane and M. K. Abdi, "Code Mmetrics Dataset (PU)," [Online]. Available: https://github.com/codemetricsdaset/slice_codemetricsdataset/.

[24] F. Tip, "A Survey of Program Slicing Techniques," J. Program. Lang., vol. 5399, no. 3, pp. 1–65, 1995.

[25] M. Weiser, "Program Slicing," IEEE Trans. Softw. Eng., vol. SE-10, no. 4, pp. 352–357, DOI: 10.1109/TSE.1984.5010248, Jul. 1984.

[26] J. Silva, "A Vocabulary of Program Slicing-based Techniques," ACM Comput. Surv., vol. 44, no. 3, pp. 1–41, DOI: 10.1145/2187671.2187674, Jun. 2012.

[27] R. Russell et al., "Automated Vulnerability Detection in Source Code Using Deep Representation Learning," Proceedings of the 17th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 757–762, DOI: 10.1109/ICMLA.2018.00120, Orlando, USA, 2019.

[28] Z. Li, D. Zou, S. Xu, H. Jin, Y. Zhu and Z. Chen, "SySeVR: A Framework for Using Deep Learning to Detect Software Vulnerabilities," arXiv:1807.06756v2, pp. 1–13, DOI: 10.21227/fhg0-1b35, Jul. 2018.

[29] Z. Li, D. Zou, J. Tang, Z. Zhang, M. Sun and H. Jin, "A Comparative Study of Deep Learning-based Vulnerability Detection System," IEEE Access, vol. 7, pp. 103184–103197, DOI: 10.1109/ACCESS.2019.2930578, 2019.

[30] S. Liu et al., "CD-VulD: Cross-Domain Vulnerability Discovery Based on Deep Domain Adaptation," IEEE Trans. Dependable Secur. Comput., pp. 1–1, DOI: 10.1109/TDSC.2020.2984505, 2020.

[31] T. Mikolov, K. Chen, G. Corrado and J. Dean, "Efficient Estimation of Word Representations in Vector Space," Proc. of the 1st International Conference on Learning Representations (ICLR 2013), arXiv:1301.3781v3, [Online], Available: https://storage.googleapis.com/pub-tools-public-publication-data/pdf/41224.pdf, 2013.

[32] C. Tomas Mikolov, "Word2Vec.," Google Inc., Mountain View, [Online], Available: https://code.google.com/archive/p/word2vec/.

[33] Y. Kim, "Convolutional Neural Networks for Sentence Classification," Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1746–1751, DOI: 10.3115/v1/D14-1181, Doha, Qatar, 2014.

[34] S. Liu, G. Lin, Q.-L. Han, S. Wen, J. Zhang and Y. Xiang, "DeepBalance: Deep-Learning and Fuzzy Oversampling for Vulnerability Detection," IEEE Trans. Fuzzy Syst., pp. 1–1, DOI: 10.1109/TFUZZ.2019.2958558, 2019.

[35] X. Ban, S. Liu, C. Chen and C. Chua, "A Performance Evaluation of Deep-learnt Features for Software Vulnerability Detection," Concurrency Computation, vol. 31, no. 19, DOI: 10.1002/cpe.5103, 2019.

[36] T. M. P. Bojanowski, E. Grave and A. Joulin, "fastText," Library for Efficient Text Classification and Representation Learning, [Online], Available: https://fasttext.cc/.

[37] X. Du et al., "LEOPARD: Identifying Vulnerable Code for Vulnerability Assessment through Program Metrics," Proceedings of the 41st International Conference on Software Engineering (ICSE '19), vol. 2019-May, pp. 60–71, DOI: 10.1109/ICSE.2019.00024, Jan. 2019.

[38] K. Pan, S. Kim and E. Whitehead, Jr., "Bug Classification Using Program Slicing Metrics," Proc. of the 6th IEEE International Workshop on Source Code Analysis and Manipulation, pp. 31–42, DOI: 10.1109/SCAM.2006.6, 2006.

[39] J. Wieting, M. Bansal, K. Gimpel and K. Livescu, "Towards Universal Paraphrastic Sentence Embeddings," Proc. of the 4th International Conference on Learning Representations (ICLR 2016), pp. 1-19, [Online], Available: https://arxiv.org/pdf/1511.08198.pdf, 2016.

[40] S. Arora, Y. Liang and T. Ma, "A Simple But Tough-to-beat Baseline for Sentence Embeddings," Proc. of the 5th International Conference on Learning Representations (ICLR 2017), pp. 1-16, [Online], Available: https://openreview.net/pdf?id=SyK00v5xx, 2019.

[41] Z. Li, D. Zou, S. Xu, H. Jin, Y. Zhu and Z. Chen, "SeVC and SyVC Dataset," [Online], Available: https://github.com/SySeVR/SySeVR/.

[42] T. M. P. Bojanowski, E. Grave, A. Joulin, "fastText Documentation," [Online], Available: https://fasttext.cc/docs/.

[43] DL4J, "Deep Learning for Java," [Online], Available: https://deeplearning4j.org/, 2020.

[44] GitHub, "Char N-gram Embedding Dataset for DL-based AVP," [Online], Available: https://github.com/dzresearcher/char_n-gram_embedding_dataset_for_DL_AVP.

[45] S. Lang, F. Bravo-Marquez, C. Beckham, M. Hall and E. Frank, "WekaDeeplearning4j: A Deep Learning Package for Weka Based on Deeplearning4j," Knowledge-Based Syst., vol. 178, pp. 48–50, DOI: 10.1016/j.knosys.2019.04.013, Aug. 2019.

[46] Machine Learning Group at the University of Waikato, "Weka API Online Doc," [Online], Available: http://weka.sourceforge.net/doc.dev/.

[47] GitHub, "Online Documentation of the Wekadeeplearning4j Java API," [Online], Available: https://waikato.github.io/wekaDeeplearning4j/.

ملخص البحث:

لقد تم بنجاحٍ تطبيق تقنيات التّعلُّم العميق لحل المشكلات التي تشكل تحدياتٍ في مجال معالجة اللغات الطبيعية. ونظراً لأن رمز المصدر والنص الطبيعي يشتركان في عدد من الأمور المتشابهة، فقد أمكن اعتماد تقنيات تصنيف النصوص - مثل تضمين الكلمات - لاقتراح طرقٍ تستند على التعلُّم العميق للقيام بالتّنبؤ الآلي بالثغرات البرمجية. وعلى الرغم من أن النتائج كانت مثيرة للاهتمام، فلم تكن بالجودة الكافية مقارنةً بتلك التي تم الحصول عليها في معالجة اللغات الطبيعية. وفي هذه الورقة، نقترح طريقة محسنة قائمة على التنبؤ الآلي بالثغرات البرمجية بناءً على تقنية تضمين الرموز (ن-غرام). وقد عملنا على تقييم الطريقة المقترحة لأربعة أنواع من الثغرات البرمجية مستخدمين قاعدة رموز ضخمة ذات مصدر مفتوح بلغة C/C++. وقد أظهرت النتائج أن الطريقة المقترحة تتمتع بأداءٍ جيد للغاية يتفوق على كثيرٍ من الطرق الواردة في دراساتٍ سابقة.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 07, No. 01, March 2021.

QUANTUM-DOT CELLULAR AUTOMATA-BASED SUPERIOR DESIGN OF CONSERVATIVE REVERSIBLE PARITY LOGIC CIRCUITS

Ali H. Majeed

(Received: 1-Sep.-2020, Revised: 16-Oct.-2020, Accepted: 5-Nov.-2020)

ABSTRACT

Quantum-dot Cellular Automata (QCA) is an innovative technology in the nano-scale for changing the CMOS revolution with an alternative one. It provides some benefits in reversible logic, like competitive power consumption and feature size. Therefore, much attention is paid to producing different reversible circuits using that technique. This paper presents a superior model for a reversible Feynman gate-based odd parity generator and checker. The proposed model can be utilized for loss bit detection /checking in telecommunication systems. The circuit verification is carried out using the QCADesigner tool. The proposed Feynman gate provides an improvement of 50% and 48% in terms of latency and cost, respectively. The parity generator, parity checker and nano-communication circuit have complexity reduction by 25%, 37% and 24%, respectively, in terms of requiring cells.

KEYWORDS

QCA, Reversible gate, Parity generator, Parity checker.

1. INTRODUCTION

In 1965, a table for integrated CMOS devices inside a chip was predicted by Moore [1]. The devices in this table grow exponentially as Moore described. Over time, the number of devices within the chip will reach the maximum value, so that it cannot be increased due to physical restrictions. This motivated scientists to think about new solutions to replace CMOS in order to keep Moore's table continuing. The QCA paradigm is one of the new nanotechnologies explored in 1993 by Lent et al. [2] as a CMOS alternative in digital systems [3]. This technology has a different computational paradigm compared to CMOS [4]. The basic building block in this nano-technique is a quantum cell. QCA cell has a square shape injected with two electrons [5]. The Coulomb repulsion of electrons enforces it to localize at corners. Because there are only two probabilities for localizing the electrons, QCA cells can represent binary numbers. Reversible circuits for parity generator and parity checker are necessary in telecommunication systems for the self-detection of errors [6]. This paper presents a new QCA structure of the reversible Feynman gate. The presented gate is used for designing a new form of reversible parity generator/checker. The QCADesigner tool [7] will be used in default parameters to display the input/output waveforms.

This paper will be organized as follows:

Section 2 will give a QCA background, while Section 3 will explain the reversible Feynman gate. Sections 4 will show the reversible Odd-Parity Generator/Checker and Section 5 will explain the proposed nano-communication system module. Section 6 will detail the simulation results and comparison and finally, the conclusion will be presented in Section 7.

2. BACKGROUND

This section provides an overview of QCA technology in terms of the main unit and working principle.

2.1 QCA Basics

The QCA-based design consists of a group of cells. Each cell contains four holes (dots) and two particles (electrons) [8]-[9]. These particles have the ability to tunnel between dots inside the cell, but cannot

Ali H. Majeed is with Department of Electrical Engineering, Faculty of Engineering, University of Kufa, Iraq. Email: alih.alasady@uokufa.edu.iq

escape out [10]. The Coulomb interaction forces the electrons to settle in a diagonal position depending on the driver cell [11]. Cell polarization is illustrated in Figure 1, where P=1 represents (logic1) and P=-1 represents (logic 0) [12]. QCA wire and many logic functions can be constructed by arranging a set of cells.

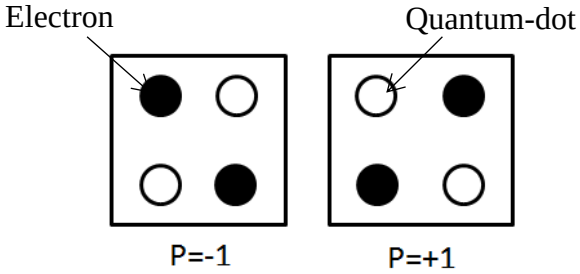


Figure 1. QCA cell polarization [13].

2.2 QCA Wire

The QCA wire consists of a group of primary cells, where it transfers the logical value from input to output and the Coulomb interaction transfers the polarization from one cell to another [14]-[15]. The two main configurations of QCA wire are illustrated in Figure 2.

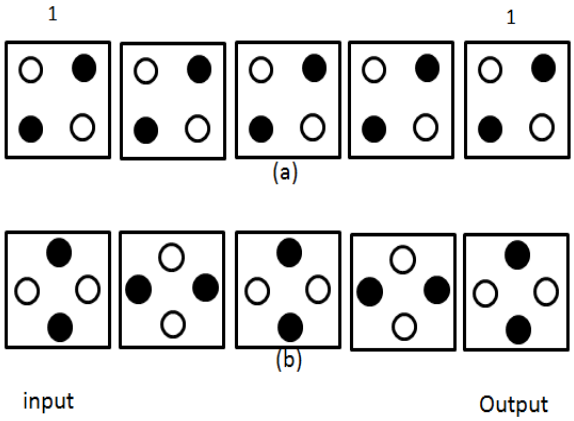


Figure 2. QCA binary wire (a) Normal arrangement (b) Rotated arrangement [15].

2.3 QCA Building Units

The dominant block in QCA circuits is the majority gate, where when applying any input to 1 or 0, the AND or OR gate can be obtained [16]-[17]. Many researchers paid attention to this gate as in [18]-[20]. The majority gate with three inputs is presented in two forms, as shown in Figure 3 [21]. The general formula of this gate is given by:

$$\mathrm{Maj}_{(A,B,C)} = AB + BC + CA \tag{1}$$

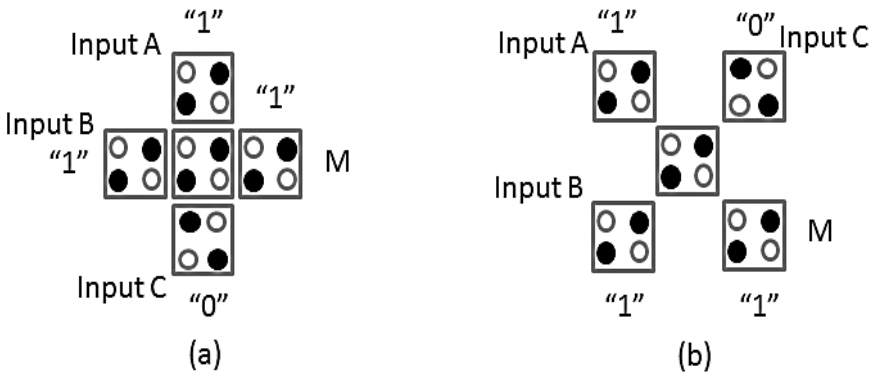


Figure 3. Majority gate layouts.

The primary building blocks in QCA are inverter and majority gate, where it is possible to design any logic circuit with these blocks only [22]. The three configurations of the QCA inverter are shown in Figure 4.

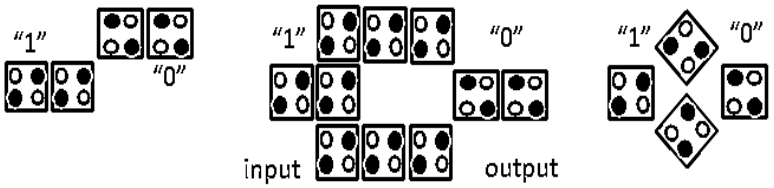


Figure 4. QCA inverter layouts.

2.4 QCA Clock Signal

The computation and synchronization in QCA are done using the clock signal [23]. Clocking is important also for letting the information flow from input to output [11], [24]. The barrier level is essential for tunnelling the electrons between dots, where it can be controlled by the clock signal. The polarization of cell remains unclear as long as the clocking is low. Whenever the clocking reaches the highest level, the cell gets its fixed polarization. Adiabatic switching is essential in QCA by splitting the clock signal into four phases (switch, hold, release and relax), where the QCA circuit can be divided further into four clock zones. Figure 5 illustrated this process [25]-[26].

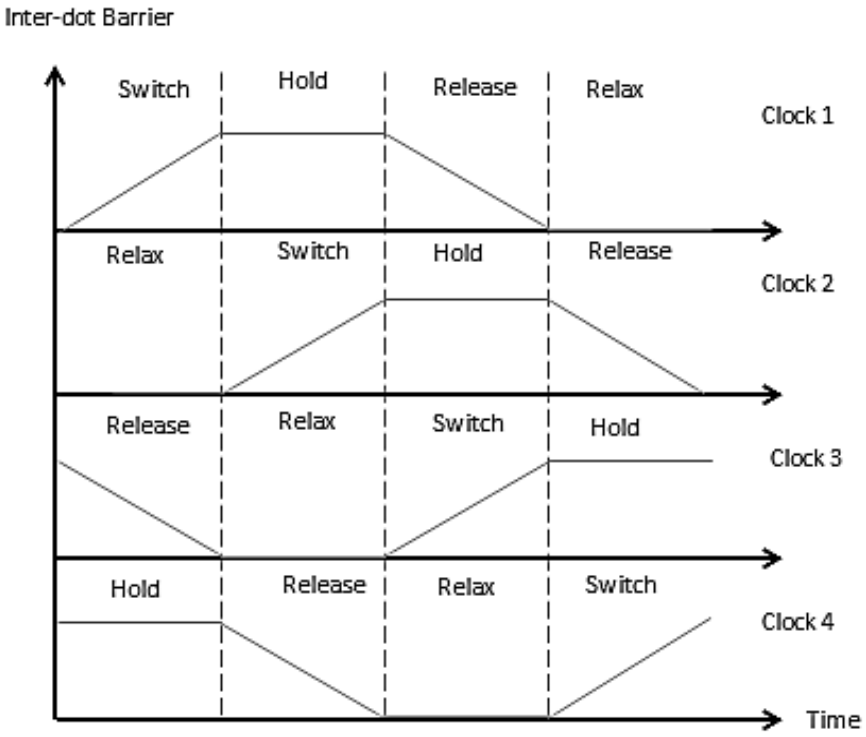


Figure 5. Clock signal phases in four zones.

3. REVERSIBLE FEYNMAN GATE

The Feynman gate is a logic reversible block that receives 2 inputs (X1 and X2) and generates 2 outputs (M and N). The input value can be identified by the output value because, there is a one-to-one matching between the input-output values. Output equations can be expressed as follows:

M=x1 and N=X1⊕X2. The Feynman logic diagram [6] and the proposed QCA layout with input/output waveforms are shown in Figure 6. The proposed Feynman gate, inspired from the XOR gate given by [27], has many advantages, such as its complexity=11 cells and area=0.00096 μm^2 with latency=0.25

clock cycle. These features make its superior compared with previous designs. A QCA cell generally has two electrons. These particles change their position depending on the principle of electron repulsion and are affected by the driver cell as well as the surrounding cells. Therefore, several QCA gates have been suggested in the literature taking advantage of this potential, such as [28]-[29]. The QCA-XOR gate used in this work was derived from QCA's inherent capability and did not follow any Boolean function. The performance of the proposed gate in terms of power consumption is achieved using the estimator tool called QCAPro [30]. The average energy dissipated (leakage and switching) can be calculated in three different levels of King Energy (0.5 E_k, 1 E_k and 1.5 E_k) using this tool, as detailed in Figure 7. The output as observed from the simulation waveforms can be expressed as in Table 1.

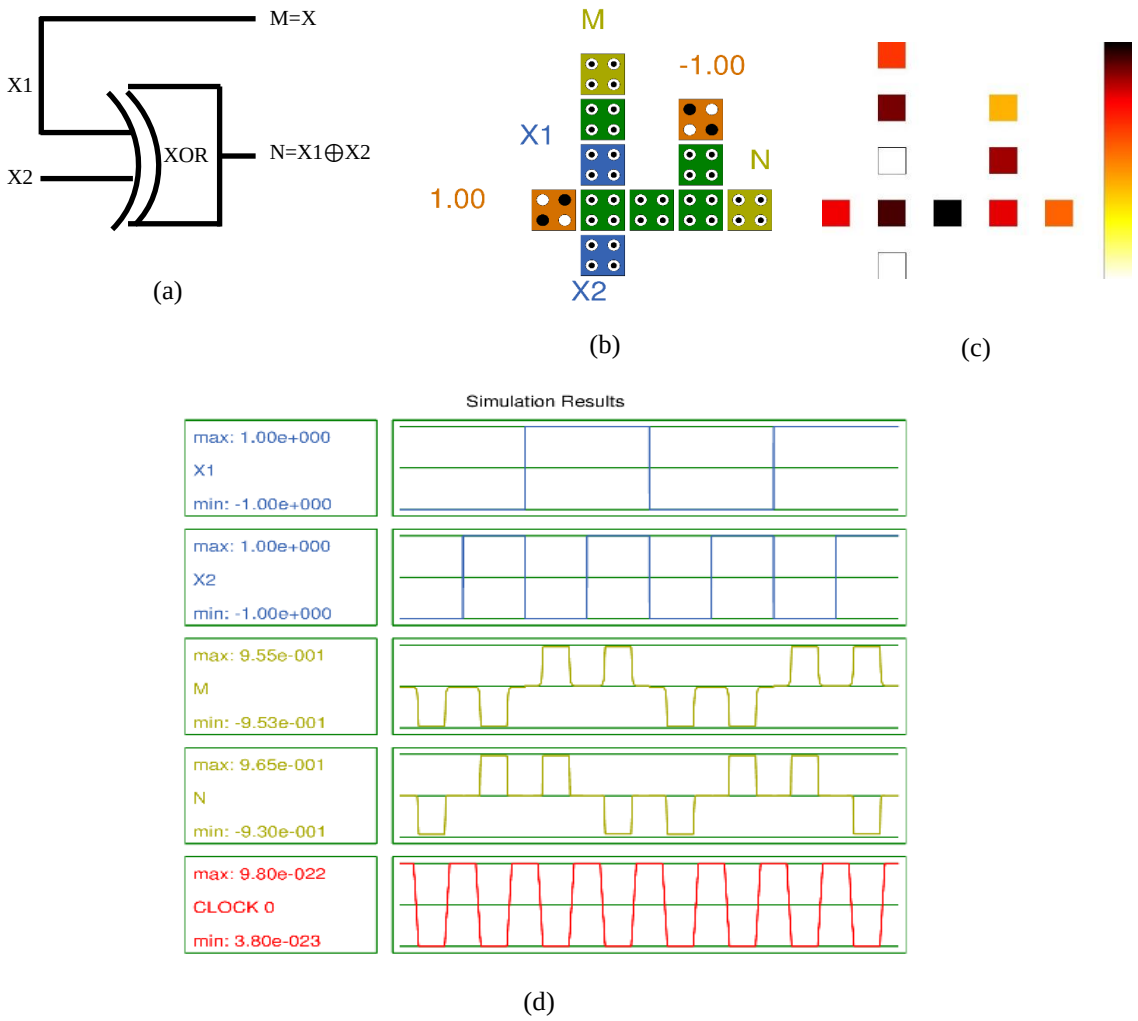


Figure 6. Feynman gate (a) Logic diagram [6], (b) QCA form (c) Power map at 0.5 E_k and (d) Simulation waveforms.

Table 1. Input-output Feynman gate.

X1	X2	M	N
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

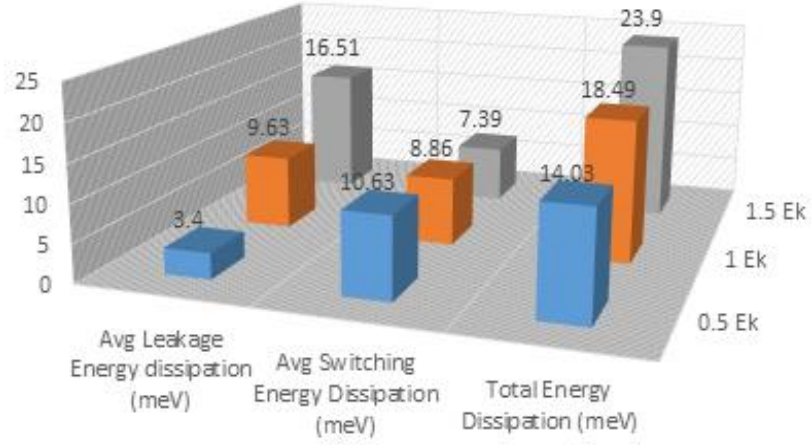


Figure 7. Power consumption of the proposed gate in three different levels.

4. REVERSIBLE ODD-PARITY GENERATOR / CHECKER

The odd-parity generator has been designed utilizing a cascade of proposed Feynman gates. The parity-bit generator gives an output when applying three values (X1, X2 and X3) at the inputs. The logic diagrams with QCA form are illustrated in Figure 8. The wonderful feature of the proposed generator reduces the complexity by 0.25%, where it has complexity=24 cells and area=0.035 μm^2 with latency=0.75 clock cycle.

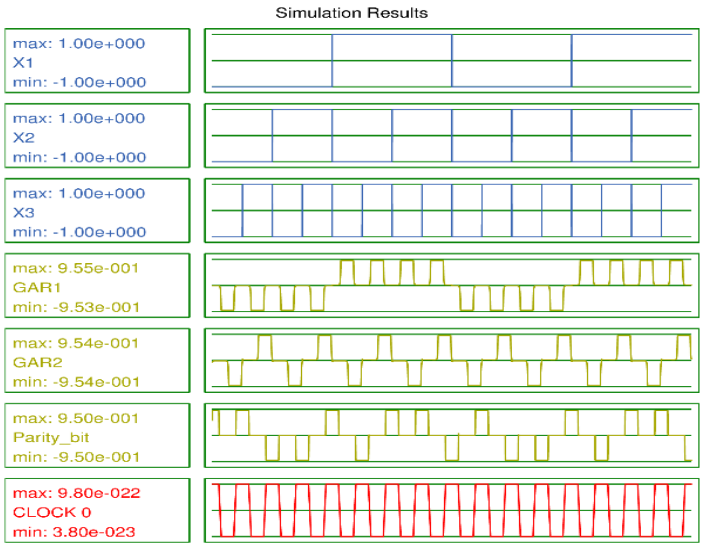


Figure 8. Reversible odd parity generator (a) Logic diagram [6] (b) Proposed QCA form and (c) Simulation results.

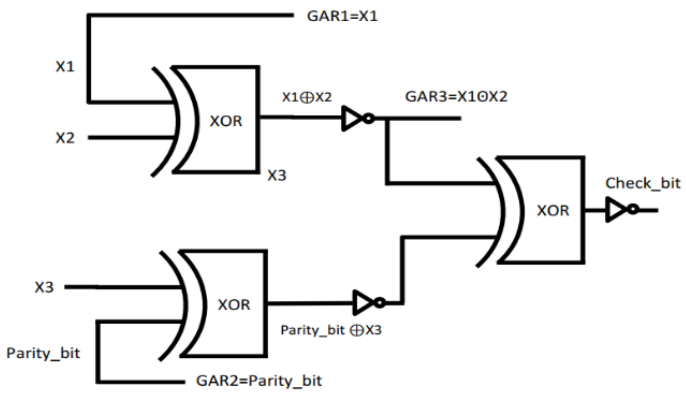
The extra parity-bit forms a message word with an odd number of 1's, where if the four-bit message received contains even 1's, that indicates an error to have occurred in the transmission. The input-output odd parity as observed from the simulation results shown in Figure 8c can be expressed as illustrated in Table 2.

The parity checker is utilized for checking whether an error that occurs at the parity-bit has been previously added in the message at the transmitter. Check-bit=1 when the number of 1's becomes even at the message word (input message includes parity-bit) indicating an occurring error; otherwise, check-bit=0 when error-free. The reversible odd-parity checker produces an output by applying XNOR operation for the input bits and parity-bit. Figure 9 illustrates the logic diagram and proposed QCA layout with simulation results.

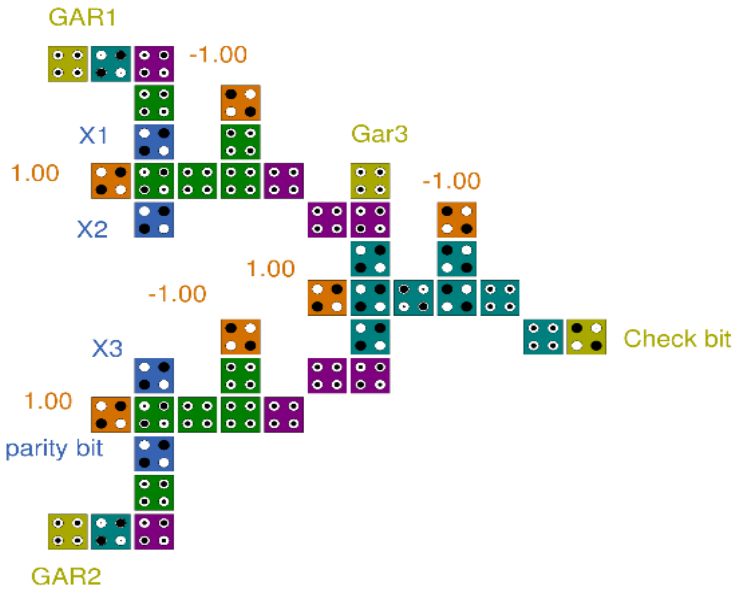
Table 2. Input-output odd parity generator.

X1	X2	X3	GAR1	GAR2	Parity-bit
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	1	0

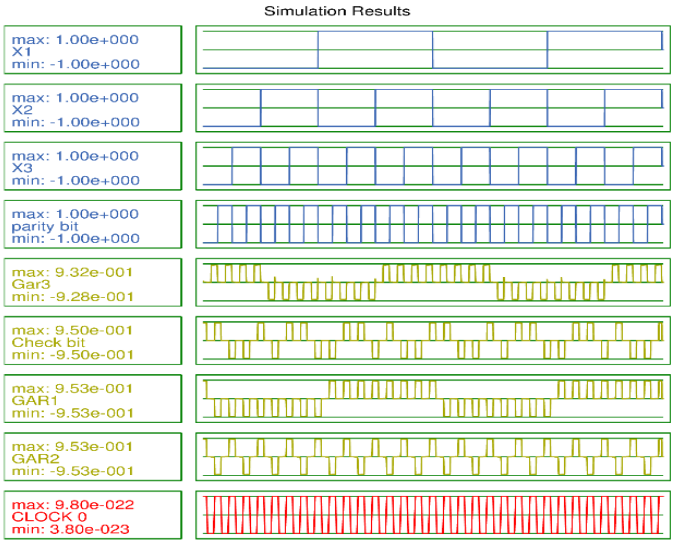
The wonderful feature of the proposed odd parity checker is that using a coherence vector simulation engine reduces the complexity by 0.37%, where it has complexity=42 cells and area=0.06 μm^2 with latency=0.75 clock cycle. Table 3 explains the input-output binary data as observed from the simulation results shown in Figure 9c.



(a)



(b)



(c)

Figure 9. Reversible odd parity checker (a) Logic diagram [6] (b) Proposed QCA layout (c) Simulation results.

Table 3. Input-output odd parity checker.

X1	X2	X3	Parity bit	GAR1	GAR2	GAR3	Check bit
0	0	0	0	0	0	1	1
0	0	0	1	0	1	1	0
0	0	1	0	0	0	1	0
0	0	1	1	0	1	1	1
0	1	0	0	0	0	0	0
0	1	0	1	0	1	0	1
0	1	1	0	0	0	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	0	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	0	0	1
1	0	1	1	1	1	0	0
1	1	0	0	1	0	1	1
1	1	0	1	1	1	1	0
1	1	1	0	1	0	1	0
1	1	1	1	1	1	1	1

5. THE PROPOSED NANO-COMMUNICATION SYSTEM MODULE

For the nano-communication system, the self-checking is important for error detection in telecommunication networks. An optimal nano-communication system has been performed utilizing the proposed reversible generator/checker (odd-parity). The proposed module has been constructed by three separated blocks (transmitter, receiver and transmission medium). The transmitter generates an extra bit (parity bit) as a tail of three input message bits to form a pattern with an odd number of 1's.

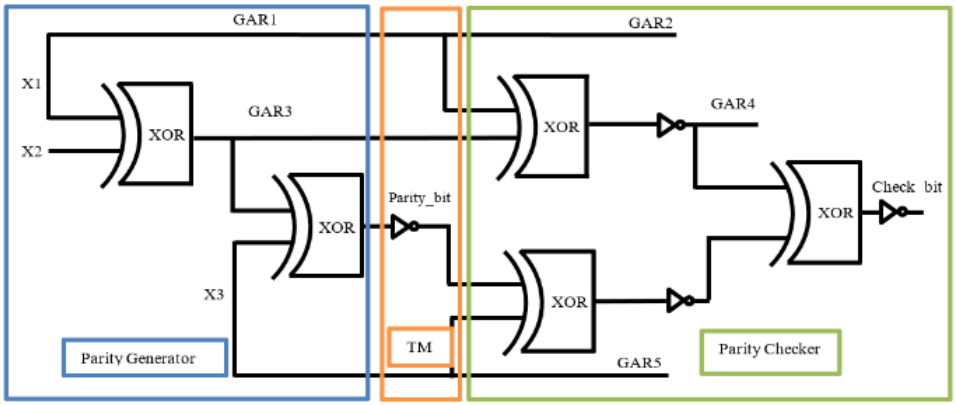


Figure 10. Nano-communication system block diagram proposed by [6].

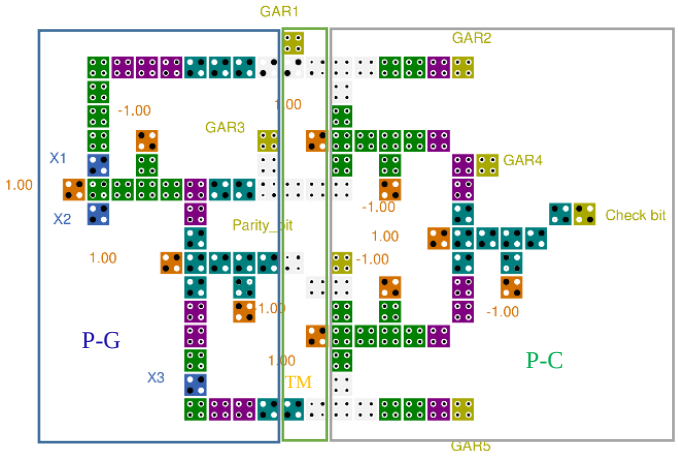


Figure 11. Proposed QCA layout of the nano-communication system.

The receiver receives the message word that contains the tail (parity bit) which has been generated by the transmitter. The receiver checks the parity bit for detecting whether an error has occurred. The message pattern is error-free if it contains an odd number of 1's; otherwise, the receiver indicates an occurring error through the transmission. Figure 10 illustrates the module logic diagram, while the proposed QCA form of the nano-communication system is shown in Figure 11. The proposed module requires 106 cells only and the occupation area=0.14 μm^2 with delay latency=1.75 clock cycle. The transmission medium is a link between transmitter and receiver. The message word includes the parity bit sent to the receiver *via* a transmission medium. The truth table of the nano-communication system is illustrated in Table 4.

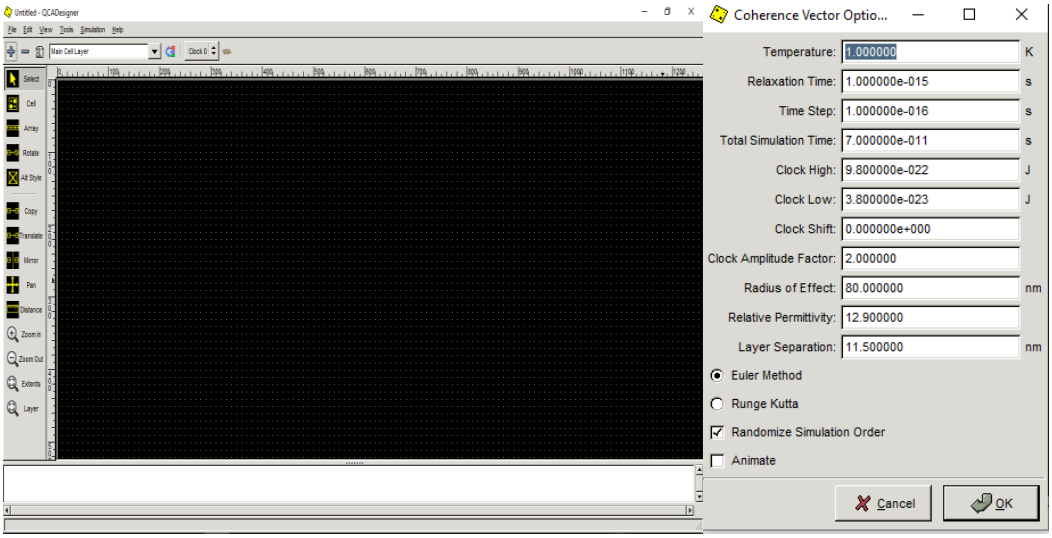
Table 4. Nano-communication system truth table.

Parity Generator by transmitter				Parity Checker				
message word			Parity bit	Received Message by receiver				Check bit
0	0	0	1	0	0	0	1	0
0	0	1	0	0	0	1	0	0
0	1	0	0	0	1	0	0	0
0	1	1	1	0	1	1	1	0
1	0	0	0	1	0	0	0	0
1	0	1	1	1	0	1	1	0
1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	0

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 07, No. 01, March 2021.

6. SIMULATION RESULTS AND PERFORMANCE COMPARISON

The QCADesigner simulation tool V2.0.3 with default parameters has been used for verifying the performance of the proposed circuits. This tool is more flexible in handling cells and circuit design. Further, circuit simulation is carried out with the same tool to show the output waveforms. In this work, this tool is used in default parameters and the screen layout with parameters is shown in Figure 12. The simulation results are illustrated in Figures 6d, 8c, 9c and 13. The proposed QCA layouts are superior in many aspects, such as area, number of cells needed and minimum delay latency. A comparison between existing and proposed designs is illustrated in Table 5. In QCA technology, many factors have been presented to compare circuits, such as delay, complexity, area and cost. The cost function in the QCA has been presented in several approaches. In this research, the cost function is calculated as the approach presented in [31]. The proposed Feynman gate provides an improvement of 50% and 48% in terms of latency and cost, respectively. The parity generator gives an improvement of 25 % and 20% in terms of cell count and cost, while the improvements for the proposed parity checker are as follows: 26%, 37%, 25% and 65% in terms of area, cell count, latency and cost, respectively. In addition, the proposed nano-communication system gives improvements of 10%, 24%, 13% and 40% in terms of area, cell count, latency and cost, respectively.



(a) (b)

Figure 12. QCADesigner tool (a) Screen layout, (b) Default parameters.

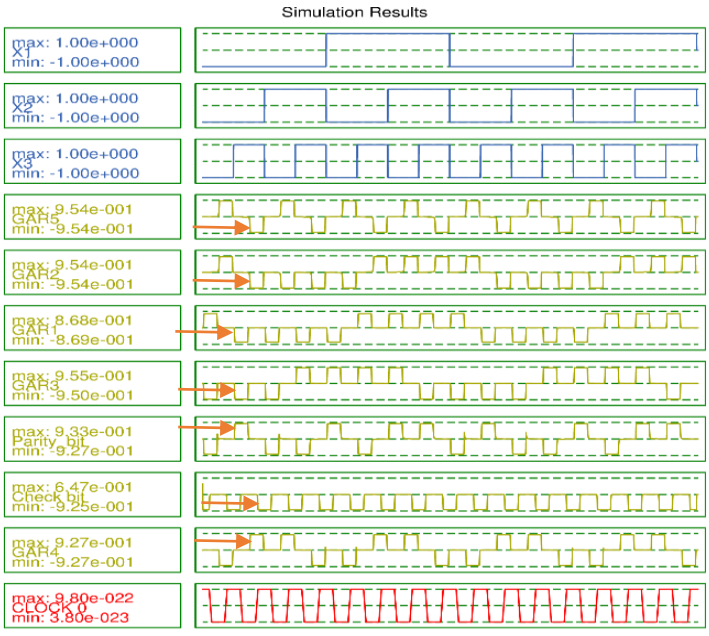


Figure 13. The input/output waveforms of the proposed nano-communication system.

Table 5. Comparison of the proposed circuits with existing designs.

QCA Circuit	Area (μm^2)	Cell Count	Latency	Cost (Area*Delay*Complexity)
Feynman Gate [32]	0.112	90	1.75	17.64
Feynman Gate [33]	0.038	43	0.75	1.2255
Feynman Gate [34]	0.08	75	1.25	7.5
Feynman Gate [35]	0.038	54	0.5	1.026
Feynman Gate [36]	0.07	53	0.75	2.7825
Feynman Gate [37]	0.038	34	0.75	0.969
Feynman Gate [6]	0.017	16	0.5	0.136
Feynman Gate [38]	0.0092	11	0.5	0.0506
proposed Feynman Gate	0.0096	11	0.25	0.0264
Parity Generator [33]	0.078	72	1.75	9.828
Parity Generator [6]	0.033	32	0.75	0.792
Proposed Parity Generator	0.035	24	0.75	0.63
Parity Checker [33]	0.143	130	2	37.18
Parity Checker [6]	0.081	67	1	5.427
Proposed Parity Checker	0.06	42	0.75	1.89
Nano-Communication Circuit [33]	0.479	293	2	280.694
Nano-Communication Circuit [6]	0.155	140	2	43.4
Proposed Nano-Communication Circuit	0.14	106	1.75	25.97

7. CONCLUSION

Designing a logic circuit in the nano-level is the goal of scientists in the digital world. QCA technology is one of many proposed solutions. This paper introduces a new QCA layout of odd parity bit circuit (generator and checker). The presented design is in optimal form and implemented based on the proposed layout of the QCA Feynman gate. An efficient nano-communication system is accomplished using the proposed generator and checker circuit. This system has the ability to self-check the error that might occur in message words during transmission. The proposed QCA layout has a superior performance in many metrics used in QCA circuit, such as area, delay (latency) and complexity (cell count) in comparison with conventional presented circuits. The proposed circuits have been verified using QCADesigner tool V 2.0.3 with default parameters.

REFERENCES

[1] G. E. Moore, "Cramming More Components onto Integrated Circuits, Reprinted from Electronics, volume 38, number 8, April 19, 1965, pp.114 ff," IEEE Solid-state Circuits Society Newsletter, vol. 11, no. 3, pp. 33-35, 2006.

[2] C. S. Lent et al., "Quantum Cellular Automata," Nanotechnology, vol. 4, pp. 49-57, 1993.

[3] H. Rashidi and A. Rezai, "High-performance Full Adder Architecture in Quantum-dot Cellular Automata," The Journal of Engineering, vol. 2017, pp. 394-402, 2017.

[4] Ali H. Majeed, E. AlKaldy, M. S. B. Zainal and Danial B. M. D. Nor, "A New 5-input Majority Gate without Adjacent Inputs Crosstalk Effect in QCA Technology," Indonesian Journal of Electrical Engineering and Computer Science, vol. 14, pp. 1159-1164, 2019.

[5] M. Ali Hussien, Z. Moh'd Shamian and A. Esam, "Quantum-dot Cellular Automata: Review Paper,"

International Journal of Integrated Engineering, vol. 11, no. 8, pp. 143-158, 2019.

[6] A. N. Bahar, F. Ahmad, N. M. Nahid, M. Kamrul Hassan, M. Abdullah Al-Shafi and K. Ahmed, "An Optimal Design of Conservative Efficient Reversible Parity Logic Circuits Using QCA," International Journal of Information Technology, vol. 11, pp. 785-794, DOI: 10.1007/s41870-018-0226-9, 2018.

[7] K. Walus, T. J. Dysart, G. A. Jullien and R. A. Budiman, "QCADesigner: A Rapid Design and Simulation Tool for Quantum-dot Cellular Automata," IEEE Trans. on Nanotechnology, vol. 3, pp. 26-31, 2004.

[8] A. H. Majeed, B. Salih, M. S. bin Zainal and D. Bin Md Nor, "Power Efficient Optimal Structure CAM-cell in QCA Technology," Indian Journal of Science and Technology, vol. 12, no. 37, pp. 1-6, 2019.

[9] I. Edrisi Arani and A. Rezai, "Novel Circuit Design of Serial–Parallel Multiplier in Quantum-dot Cellular Automata Technology," Journal of Computational Electronics, vol. 17, pp. 1771-1779, 2018.

[10] A. Majeed, E. AlKaldy and S. Albermany, "An Energy-efficient RAM Cell Based on Novel Majority Gate in QCA Technology," SN Applied Sciences, vol. 1, A., no. 1354, pp. 1-8, 2019.

[11] K.-M. Qiu and Y.-S. Xia, "Quantum-dots Cellular Automata Comparator," Proc. of the 7th International Conference on ASIC, 2007, pp. 1297-1300, Guilin, China, 2007.

[12] A. H. Majeed, E. Alkaldy, M. S. bin Zainal and D. Bin Md Nor, "Synchronous Counter Design Using Novel Level Sensitive T-FF in QCA Technology," Journal of Low Power Electronics and Applications, vol. 9, no. 3, pp. 1-13, 2019.

[13] M. Divshali, A. Rezai and S. Hamidpour, "Design of Novel Coplanar Counter Circuit in Quantum Dot Cellular Automata Technology," Int. Journal of Theoretical Physics, vol. 58, pp. 2677-2691, 2019.

[14] M. B. Khosroshahy, M. H. Moaiyeri, K. Navi and N. Bagherzadeh, "An Energy and Cost Efficient Majority-based RAM Cell in Quantum-dot Cellular Automata," Results in Physics, vol. 7, pp. 3543-3551, 2017.

[15] A. H. Majeed, M. S. B. Zainal, E. Alkaldy and D. M. Nor, "Full Adder Circuit Design with Novel Lower Complexity XOR Gate in QCA Technology," Transactions on Electrical and Electronic Materials, vol. 21, pp. 198-207, 2020.

[16] S. Azimi, S. Angizi and M. Moaiyeri, "Efficient and Robust SRAM Cell Design Based on Quantum-dot Cellular Automata," ECS Journal of Solid State Science and Technology, vol. 7, pp. Q38-Q45, 2018.

[17] M. N. Divshali, A. Rezai and A. Karimi, "Towards Multilayer QCA SISO Shift Register Based on Efficient D-FF Circuits," International Journal of Theoretical Physics, vol. 57, pp. 3326-3339, 2018.

[18] M. A. Tehrani, K. Navi and A. Kia-Kojoori, "Multi-output Majority Gate-based Design Optimization by Using Evolutionary Algorithm," Swarm and Evolutionary Computation, vol. 10, pp. 25-30, 2013.

[19] R. Zhang, P. Gupta and N. K. Jha, "Majority and Minority Network Synthesis with Application to QCA-SET- and TPL-based Nanotechnologies," IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems, vol. 26, pp. 1233-1245, 2007.

[20] M. R. Bonyadi, S. M. R. Azghadi, N. M. Rad, K. Navi and E. Afjei, "Logic Optimization for Majority Gate-based Nanoelectronic Circuits Based on Genetic Algorithm," Proc. of the IEEE 2007 International Conference on Electrical Engineering, pp. 1-5, Lahore, Pakistan, 2007.

[21] M. Bagherian Khosroshahy, M. Hossein Moaiyeri and K. Navi, Design and Evaluation of a 5-input Majority Gate-based Content-addressable Memory Cell in Quantum-dot Cellular Automata, Proc. of the 19th IEEE International Symposium on Computer Architecture and Digital Systems (CADS), DOI: 10.1109/CADS.2017.8310671, Kish Island, Iran, 2017.

[22] H. Majeed Ali, E. Alkaldy, S. Zainal Mohd, K. Navi and D. Nor, "Optimal Design of RAM Cell Using Novel 2:1 Multiplexer in QCA Technology," Circuit World, vol. 46, pp. 147-158, 2019.

[23] M. Ali Hussien, Z. Mohd Shamian, A. Esam and N. Danial Md, "A Content-addressable Memory Structure Using Novel Majority Gate with 5-input in Quantum-dot Cellular Automata," International Journal of Integrated Engineering, vol. 12, no. 4, pp. 28-38, 2020.

[24] H. Rashidi, A. Rezai and S. Soltany, "High-performance Multiplexer Architecture for Quantum-dot Cellular Automata," Journal of Computational Electronics, vol. 15, pp. 968-981, 2016.

[25] F. Ahmad, G. M. Bhat and P. Z. Ahmad, "Novel Adder Circuits Based on Quantum-dot Cellular Automata (QCA)," Circuits and Systems, vol. 05, pp. 142-152, 2014.

[26] E. Alkaldy, A. H. Majeed, M. S. bin Zainal and D. Bin Md Nor, "Optimum Multiplexer Design in

Quantum-dot Cellular Automata," Indonesian Journal of Electrical Engineering and Computer Science, vol. 17, pp. 148-155, 2020.

[27] H. Chen, H. Lv, Z. Zhang, X. Cheng and G. Xie, "Design and Analysis of a Novel Low-power Exclusive-OR Gate Based on Quantum-dot Cellular Automata," Journal of Circuits, Systems and Computers, vol. 28, no. 8, p. 1950141, 2019.

[28] M. M. Abutaleb, "A Unique Cell-based Configuration of XOR Gates in Quantum-dot Cellular Automata Nanotechnology," Proc. of the 2019 IEEE International Conference on Sensors and Nanotechnology, pp. 1-4, Penang, Malaysia, 2019.

[29] E. Taherkhani, M. H. Moaiyeri and S. Angizi, "Design of an Ultra-efficient Reversible Full Adder-subtractor in Quantum-dot Cellular Automata," Optik - International Journal for Light and Electron Optics, vol. 142, pp. 557-563, 2017.

[30] S. Srivastava, A. Asthana, S. Bhanja and S. Sarkar, "QCAPro: An Error-power Estimation Tool for QCA Circuit Design," Proc. of the 2011 IEEE International Symposium of Circuits and Systems (ISCAS), 2011, pp. 2377-2380, Rio de Janeiro, Brazil, 2011.

[31] D. Bahrepour and N. Maroufi, "A 2-bit Full Comparator Design with Minimum Quantum Cost Function in Quantum-dot Cellular Automata," Journal of Information Systems and Telecommunication (JIST), vol. 6, pp. 197-203, 2018.

[32] B. Debnath, J. C. Das, D. De, F. Ghaemi, A. Ahmadian and N. Senu, "Reversible Palm Vein Authenticator Design with Quantum Dot Cellular Automata for Information Security in Nanocommunication Network," IEEE Access, vol. 8, pp. 174821-174832, 2020.

[33] J. C. Das and D. De, "Quantum-dot Cellular Automata Based Reversible Low Power Parity Generator and Parity Checker Design for Nanocommunication," Frontiers of Information Technology & Electronic Engineering, vol. 17, pp. 224-236, 2016.

[34] P. Biswas, N. Gupta and N. Patidar, "Basic Reversible Logic Gates and It's QCA Implementation," Int. Journal of Engineering Research and Applications, vol. 4, no. 6, pp. 12-16, 2014.

[35] J. C. Das and D. De, "Reversible Binary to Grey and Grey to Binary Code Converter Using QCA," IETE Journal of Research, vol. 61, pp. 223-229, 2015.

[36] M. Abdullah Al-Shafi, M. S. Islam and A. N. Bahar, "A Review on Reversible Logic Gates and It's QCA Implementation," Int. Journal of Computer Applications, vol. 128, no. 2, pp. 27-34, 2015.

[37] A. N. Bahar, S. Waheed and M. A. Habib, "A Novel Presentation of Reversible Logic Gate in Quantum-dot Cellular Automata (QCA)," Proc. of the 2014 IEEE International Conference on Electrical Engineering and Information & Communication Technology, 2014, pp. 1-6, Dhaka, Bangladesh, 2014.

[38] M. M. Abutaleb, "Robust and Efficient QCA Cell-based Nanostructures of Elementary Reversible Logic Gates," The Journal of Supercomputing, vol. 74, pp. 6258-6274, 2018.

ملخص البحث:

تُعد آليات النُّقط الكمية الخلوية ذاتية التشغيل تقنيةً مبتكرة في نطاق تكنولوجيا النانو من أجل الاستعاضة عن ثورة تقنية (CMOS) بأخرى بديلة. وهي توفر بعض الفوائد في المنطق القابل للقلب؛ مثل الاستهلاك المنافس للقدرة، وحجم السِّمات. من هنا، اتجه الكثير من الاهتمام الى إنتاج داراتٍ قابلةٍ للقلْب باستخدام تلك التقنية المبتكرة.

تقدم هذه الورقة البحثية أنموذجاً متفوقاً لمولدٍ وفاحصٍ للتكافؤ الفردي استناداً على بوابة "فاينمان" قابلةٍ للقلْب. ومن الممكن الاستفادة من الأنموذج المقترح في الكشف عن أو فحص الفقْد في أنظمة الاتصال. وقد تم التحقق من دارات الأنموذج باستخدام أداة (QCADesigner). وتوفر بوابة "فاينمان" المقترحة تحسيناً بنسبة (50%) و (48%) من حيث التأخير، والكلفة على الترتيب. وقد قللت: مولد التكافؤ، وفاحص التكافؤ، ودارة اتصالات النانو درجة التعقيد من حيث عدد الخلايا المطلوبة بنسبة (25%) و (37%) و (24%) على الترتيب.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 07, No. 01, March 2021.

C-ELEMENT DESIGN IN QUANTUM DOT CELLULAR AUTOMATA

Mutaz Al-Tarawneh and Ziyad A. Altarawneh

(Received: 30-Aug.-2020, Revised: 14-Oct.-2020, Accepted: 5-Nov.-2020)

ABSTRACT

The continuous market demands for high-performance and energy-efficient computing systems have steered the computational paradigm and technologies towards nano-scale quantum dot cellular automata (QCA). This paper presents novel simple and complex QCA-based C-element structures. The proposed structures were thoroughly analyzed based on key design parameters, such as area, energy dissipation and robustness against structural defects. Simulation results demonstrate that the proposed simple structures have achieved up to 56% and 66% improvement in area and energy dissipation, respectively. On the other hand, the complex structures have shown a profound immunity against structural defects and achieved up to 143% improvement as compared to the simple structures. The proposed C-element structures can be considered as viable blocks for asynchronous designs.

KEYWORDS

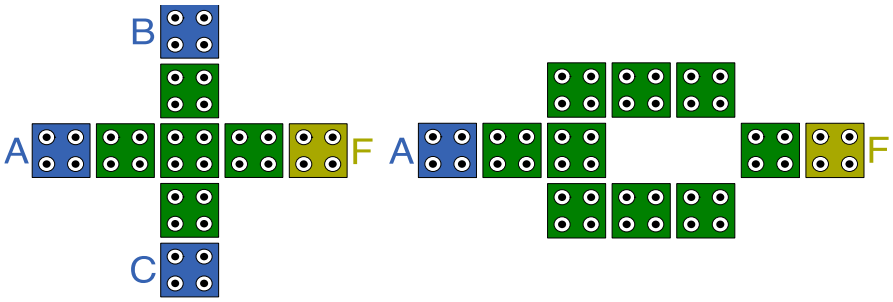
QCA, Asynchronous circuits, C-Element, Robustness, Structural defects.

1. INTRODUCTION

Over the past few decades, the microelectronics industry has been driven by increasing market demands for enhanced integration, energy efficiency and speed of integrated circuits (ICs). This was done primarily by scaling the transistor feature size and by implementing specific device architectures, such as FinFET and Gate-All-Around (GAA) nanowires [1]-[3]. Nonetheless, as the transistor feature size is reduced, some issues, like power consumption and increased leakage current, are beginning to dominate device performance due to various quantum effects and increased process variation levels at nano-scale, halting the advantages of device scaling being adopted. According to the International Technology Roadmap for Semiconductors (ITRS), the development of new computational paradigms and device structures is inevitable in future technology nodes with regard to device technology and clocking strategies [2], [4]. In this context, the nano-scale Quantum-dot Cellular Automata (QCA) technology is one potential alternative that is anticipated to overrule the VLSI technology and deliver rapid advancements in the internet of things (IoT) era [5] and information security [6]. The concept of QCA was first introduced in [7]. Unlike conventional CMOS-based structures, in which information is transferred by the flow of electrical current, QCA depends on the coulombic interaction between adjacent cells. In addition, the polarization level of the confined electrons within a QCA cell represents the binary levels in QCA-based structures. This ultimately allows the QCA-based structure to surpass CMOS-based counterparts in terms of switching speed, device density and power consumption [8]-[9]. Hence, QCA is considered as a transistor-less technology which can serve as a replacement technology to design nano-scale digital circuits [10]-[12].

Typically, QCA devices are described on the basis of symmetric square cells, whereby all computational logic gates and memory structures can be correctly imitated. These structures can be implemented by assembling QCA cells in a specific geometric pattern to achieve the desired logic function. In QCA technology, the primitive building blocks are the majority voter and inverter gates, as shown in Figure 1. The conventional AND and OR logic gates can be simply implemented based on the 3-input majority gate by setting one of the inputs to either ”0” or ”1”, respectively. An important issue in the design of QCA circuits is the switching of QCA cells from one state to another that is controlled by external clock signals. A clock signal has four sequential phases; namely, switch, hold, release and relax [13], as depicted in Figure 2.

M. Al-Tarawneh and Z. A. Altarawneh are with the Faculty of Engineering, Mutah University, Al-Karak, Jordan. Emails: mutaz.altarawneh@mutah.edu.jo and zdtarawneh@mutah.edu.jo



(a) (b)

Figure 1. Basic QCA building blocks: (a) Majority gate, (b) Inverter.

The purpose of these phases is to allow or deny the tunneling of the confined electrons in a QCA cell and in turn, achieving stable logic states and information flow by controlling the inter-dot barrier of a cell [14]. In addition, the QCA cells in a particular structure are typically grouped into sequential sub-arrays known as clock zones. The clock signals applied to consecutive clock zones are phase-shifted by 90 degrees to synchronize the polarization change within the QCA structure and prevent back-propagation of information between adjacent cells assigned to different clocking zones, as shown in Figure 3.

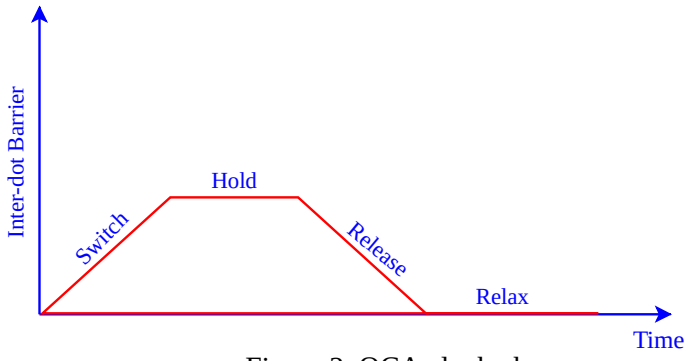


Figure 2. QCA clock phases.

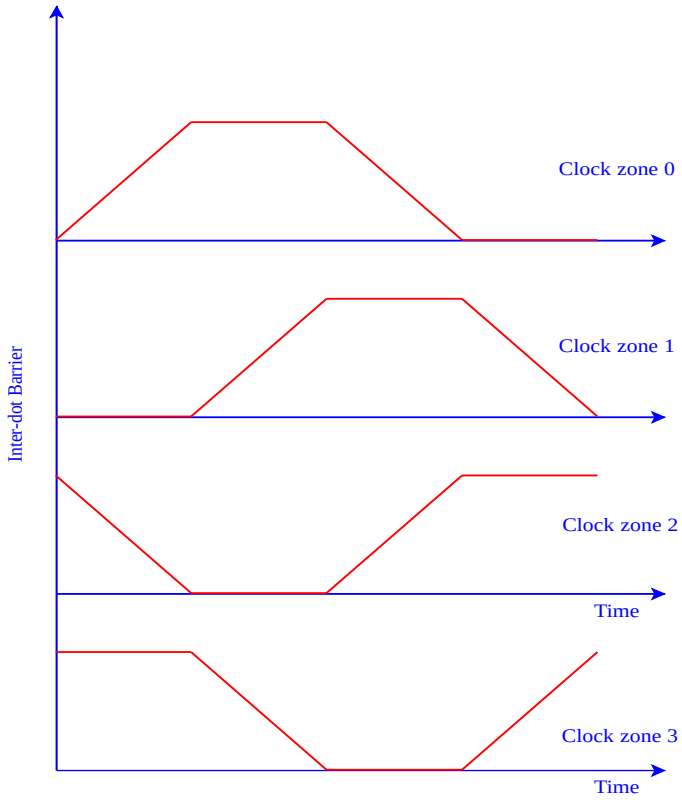


Figure 3. QCA clock zones.

Recently, extensive research efforts have been made on the design of various computational logic and memory structures, such as adders [15]-[18], multiplexers [19]-[20] and memory elements [21]-[23] based on QCA technology. More recently, Bahar et al. [11] have introduced effective single-layer binary discrete cosine transform (BinDCT) using QCA technology. In addition, the authors of [24] have proposed a QCA-based architecture of a new single-layer butterfly switching network (BSN) as a multistage interconnection network (MIN) for parallel computing. Moreover, the authors of [25] have implemented a bit-serial adder (BSA) using modified majority gate and E-shaped exclusive-OR gate. Additionally, Marshal et al. [26] have proposed novel and cost-efficient QCA-based configurable logic blocks and memory blocks, that can be used in Field Programmable Gate Array (FPGA) and embedded systems' designs. Furthermore, Song et al. [12] have suggested a novel loop-based RAM cell with asynchronous set and reset, based on a new 2-1 multiplexer and D-latch structures. Apparently, the variety of QCA-based logic and memory structures introduces QCA technology as a substantial candidate for a new computing paradigm.

A key factor in designing QCA-based structures is the reliability against structural defects. These defects can be categorized as dislocation defects that are caused by cells moving around their axis, dopant defects in which a QCA cell may have one or more extra or missing dots, interstitial (i.e., cell displacement) defects where cells may deviate from their intended horizontal or vertical orientation and vacancy defects (i.e., cell missing) caused by complete absence of cells. The presence of such defects in a computational structure may be manifested as an error that compromises the expected functionality of a design [27].

In recent years, asynchronous circuit designs have received considerable significance in the VLSI scientific community [28]-[30]. Such designs pose a great potential for low-power and high-performance computing and network-on-chip systems [31]. Asynchronous circuit designs can be used to ensure correct communication between different frequency domains. One of the most frequently used structures in constructing asynchronous circuits is the C-element, known as Muller gate [32]. This peculiar structure serves as a primitive building block in several asynchronous logic designs and is used in implementing the synchronization required by most handshaking schemes, which provides the basis for asynchronous communication, especially in micro-pipelines and some network-on-chip designs [31]-[34]. Figure 4 represents the symbolic representation of the 2-input C-element and Table 1 demonstrates its truth table. Its output (F) only changes when the inputs (A and B) have equal logical values. However, when the inputs (A and B) are different, the output (F) memorizes its previous logical state.

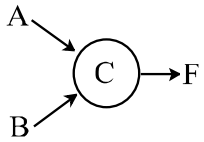


Figure 4. Symbolic representation of 2-input C-element.

Table 1. 2-Input C-element truth table.

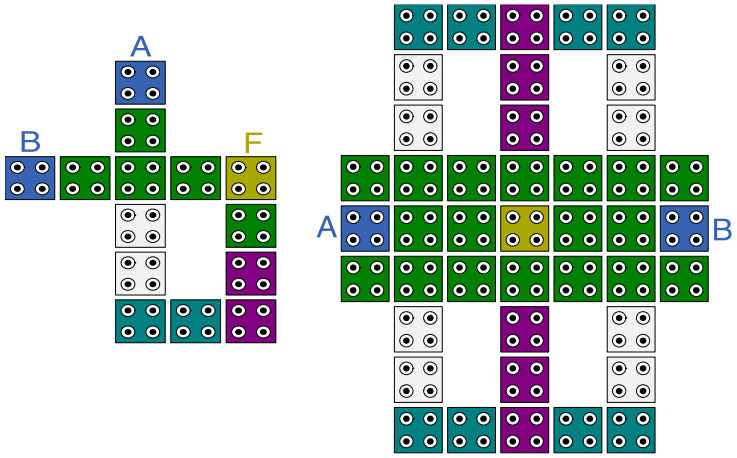
A	B	F_i
0	0	0
0	1	F_{i-1}
1	0	F_{i-1}
1	1	1

The purpose of this paper is to propose various QCA implementations of the C-element. As far as the authors could verify, there is no previous research efforts to tackle QCA-based C-element designs. This paper presents different QCA-based C-element structures with thorough analysis of their area, energy dissipation and robustness against structural defects. The proposed designs include 2-, 3- and 4-input C-elements.

The rest of this paper is organized as follows. Section 2 shows the proposed C-element structures. Section 3 presents the simulation results and compares the proposed structures in terms of their area, energy dissipation and robustness. Finally, Section 4 summarizes and concludes the paper.

2. PROPOSED QCA-BASED C-ELEMENT STRUCTURES

Figure 5a shows the proposed 2-Input QCA-based C-element structure. The simplest structure is mainly composed of a 3-input majority gate with a single feedback utilizing three different clock zones.



(a) (b)

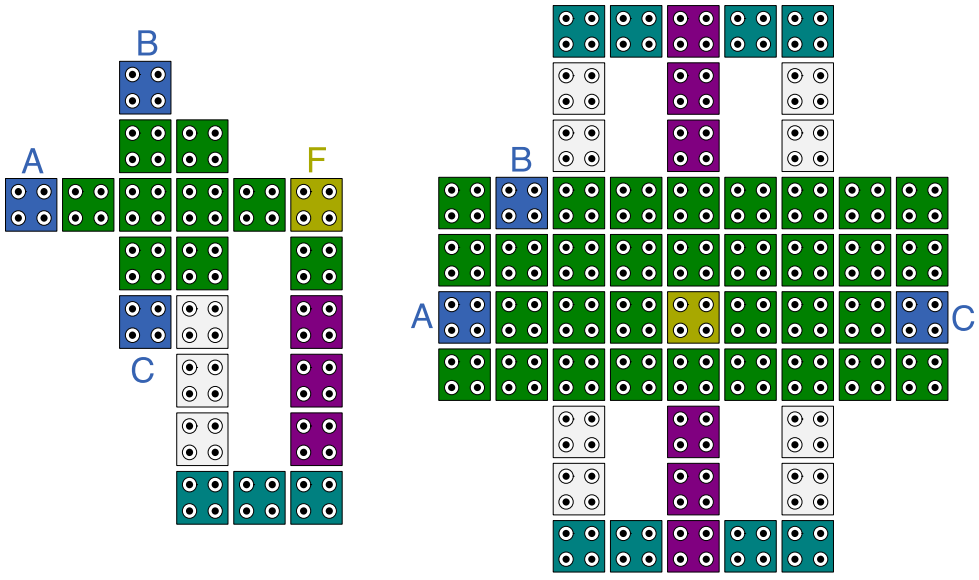
Figure 5. Proposed 2-input C-element structures: (a) Simple structure, (b) Complex structure.

As shown in Figure 5a, this structure consists of 14 cells with a total area of 0.013μm^2. The functionality of the proposed structure can be formulated as:

$$F = MAJ\ (A, B, F) = A.B + F.\ (A + B) \quad (1)$$

where A and B represent the inputs and F is the output of the C-element. In this structure, the device cell is responsible for computing the majority function between the inputs (A and B) and the output (F), while the feedback cells are responsible for controlling the flow of information from the output (F) to the device cell, allowing the proposed structure to achieve its intended functionality. Figure 5b shows an alternative design of the 2-input C-element, where the device cell is further duplicated to achieve a more robust design. In addition, the feedback has been duplicated to assure correct functionality in the presence of any structural defects. This design is composed of 43 cells with an area of 0.029μm^2.

Figure 6 shows the proposed 3−input QCA-based C-element structures. Table 2 demonstrates the truth table of the 3-input C-element.



(a) (b)

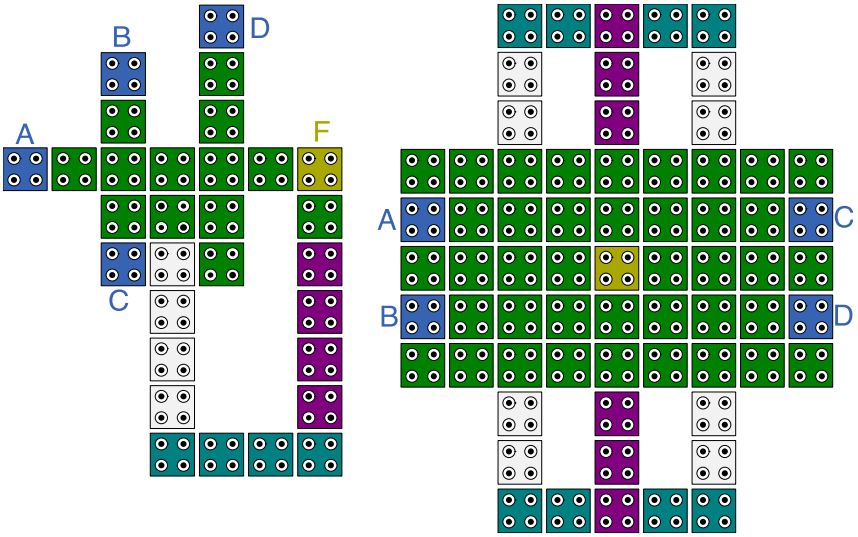
Figure 6. Proposed 3-input C-element structures: (a) Simple structure, (b) Complex structure.

Table 2. 3-Input C-element truth table.

A	B	C	F_i
0	0	0	0
0	0	1	F_{i-1}
0	1	0	F_{i-1}
0	1	1	F_{i-1}
1	0	0	F_{i-1}
1	0	1	F_{i-1}
1	1	0	F_{i-1}
1	1	1	1

As depicted in Figure 6a, the 3-input C-element can be implemented by modifying the simple 2-input C-element structure to accommodate more inputs while maintaining the clock zone sequence in the feedback. This design consists of 22 cells occupying an area of 0.021 μm^2. On the other hand, Figure 6b shows a 3-input C-element structure that is achieved by modifying the design shown in Figure 5b by adding 15 more cells to accommodate an extra input while improving the robustness against structural variations.

Figure 7 illustrates the proposed 4-input C-element structures. Table 3 shows the truth table of the 4-input C-element. Figure 7a shows the simple 4-input C-element that is achieved based on a 5-input majority gate with proper structuring of the feedback. As shown, the number of used cells is 31 with an area of 0.03μm^2. It is also possible to obtain a more robust C-element structure, as shown in Figure 7b, with a total of 67 cells.



(a) (b)

Figure 7. Proposed 4-input C-element structures: (a) Simple structure, (b) Complex structure.

Table 3. 4-Input C-element truth table.

A	B	C	D	F_i
0	0	0	0	0
0	0	0	1	F_{i-1}
0	0	1	0	F_{i-1}
0	0	1	1	F_{i-1}
0	1	0	0	F_{i-1}
0	1	0	1	F_{i-1}
0	1	1	0	F_{i-1}
0	1	1	1	F_{i-1}
1	0	0	0	F_{i-1}
1	0	0	1	F_{i-1}
1	0	1	0	F_{i-1}
1	0	1	1	F_{i-1}
1	1	0	0	F_{i-1}
1	1	0	1	F_{i-1}
1	1	1	0	F_{i-1}
1	1	1	1	1

3. RESULTS AND ANALYSIS

The QCADesigner-2.0.3 simulation tool was used to verify the functional correctness of the proposed C-element structures and assess their structural cost in terms of the occupied area [35]. The QCADesigner tool is a widely used layout and simulation tool in QCA technology to model and analyze the dynamics of QCA-based structures. In this work, simulation parameters are configured as shown in Table 4. Figures 8, 9, 10, 11, 12 and 13 show the simulation results of the proposed 2-, 3- and 4-input C-element structures, respectively under different input combinations.

As shown in Figures 8 and 9, when both inputs (A and B) are equal to (0), the output (F) is equal to (0) and maintains its logical value when one of the inputs toggles. However, when both inputs (A and B) are equal to (1), the output (F) is set to (1). Similarly, the output (F) keeps its state as long as the inputs change to distinct logical values. These observations validate the functional correctness of the proposed 2-input C-element structures. On the other hand, Figures 10, 11, 12 and 13 validate the intended functionality of the proposed 3- and 4-input structures. Apparently, the functionality of the C-element is correctly captured by the proposed structures; the output (F) only changes when the inputs have the same logic levels (0 or 1). However, when the inputs have distinct logic values, the output (F) memorizes its previous logic value.

Table 4. Simulation parameters.

Parameter	Value
Number of samples	12800
Cell Dimensions	18 nm x 18 nm
Quantum-dot diameter	5 nm
Cell separation	2 nm
Radius of effect	65 nm
Relative permittivity	12.9
Clock High	$9.8 * 10^{-22}$
Clock Low	$3.8 * 10^{-23}$
Clock shift	0
Clock amplitude factor	2
Layer separation	11.5 nm
Temperature	1K

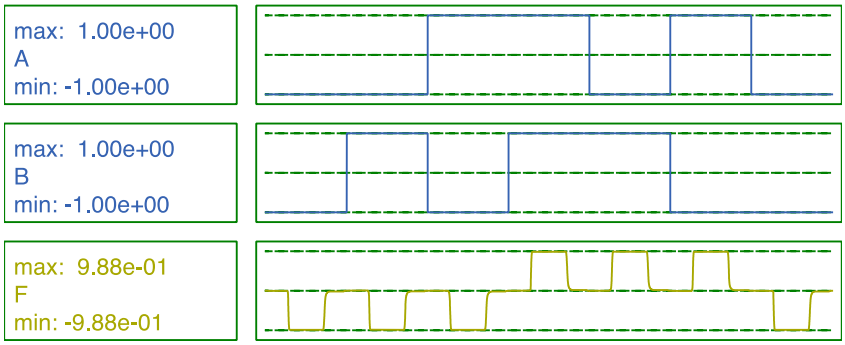


Figure 8. Simulation results of the proposed 2-input simple structure.

The only difference between simulation results of the proposed simple and complex structures is that the polarization level of the output cell (F) in the complex structures is slightly higher than that of the simple structures due to the increased level of cell interaction induced by the redundant paths to the output cell.

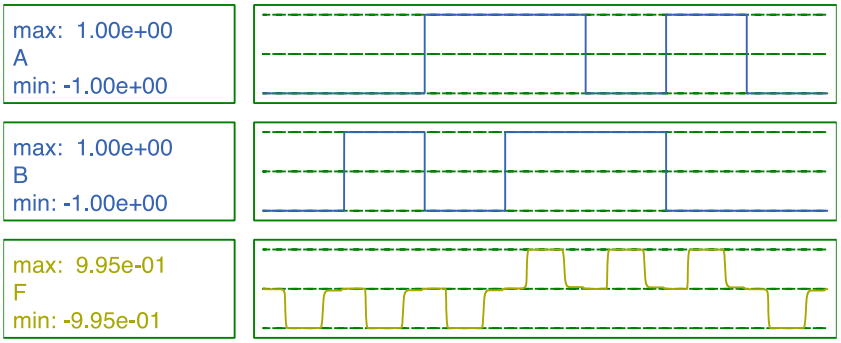


Figure 9. Simulation results of the proposed 2-input complex structure.

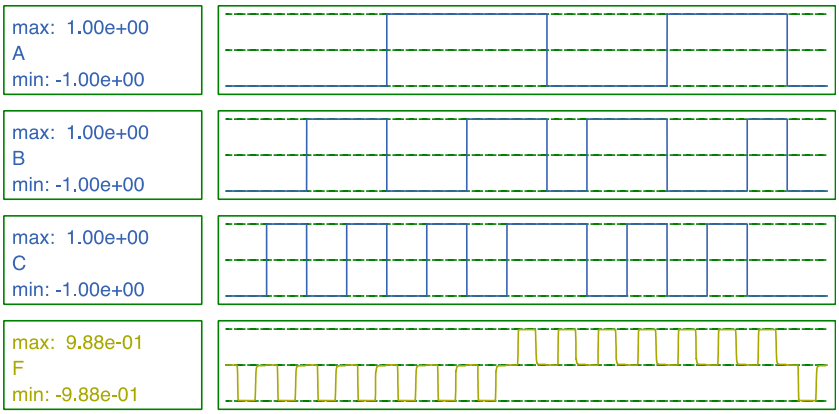


Figure 10. Simulation results of the proposed 3-input simple structure.

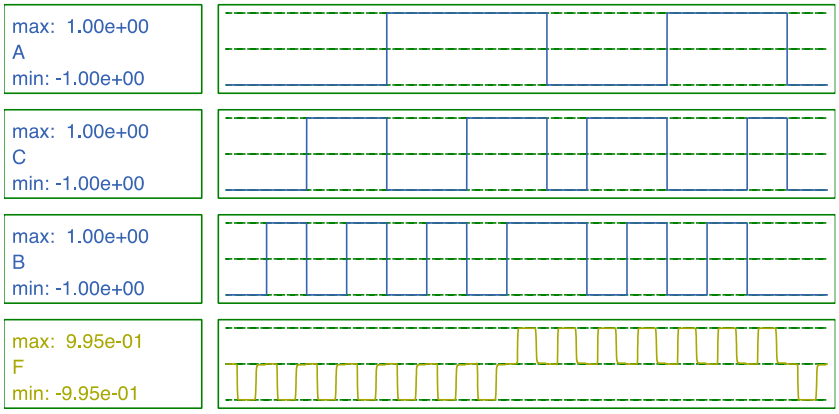


Figure 11. Simulation results of the proposed 3-input complex structure.

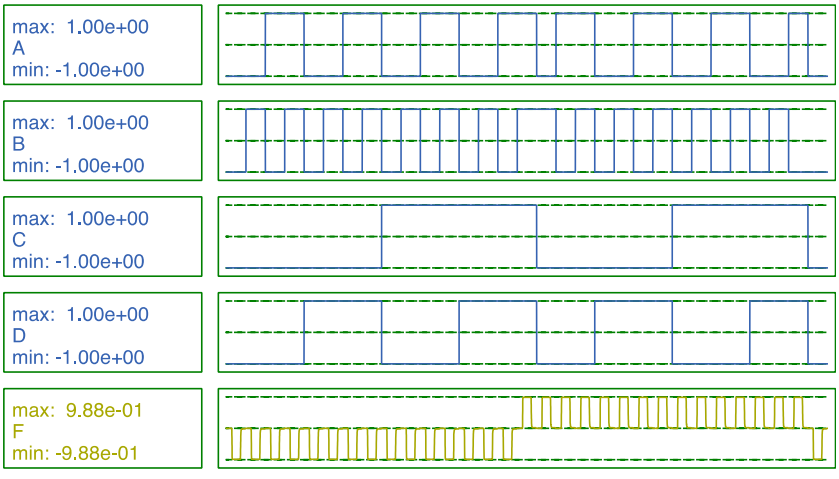


Figure 12. Simulation results of the proposed 4-input simple structure.

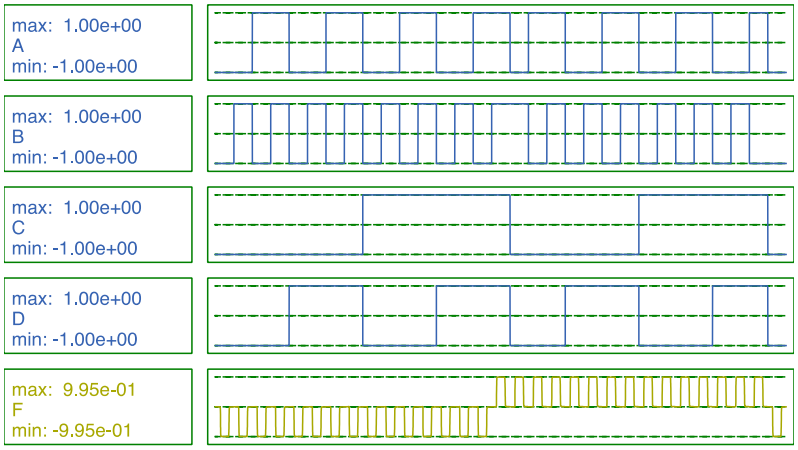


Figure 13. Simulation results of the proposed 4-input complex structure.

Figure 14 shows and compares the occupied area of the proposed C-element structures. As shown, the occupied area increases as the number of inputs is increased. In addition, the complex (i.e., with more redundant cells) structures occupy more area as compared to their simple counterparts. To estimate the energy dissipation of the various structures, the QCADesignerE tool has been used [36]. The QCADesignerE is a viable tool that models and estimates energy dissipation of QCA-based structures. Figures 15a and 15b illustrate the total energy dissipation and the average energy dissipation per clock cycle of the proposed C-element structures.

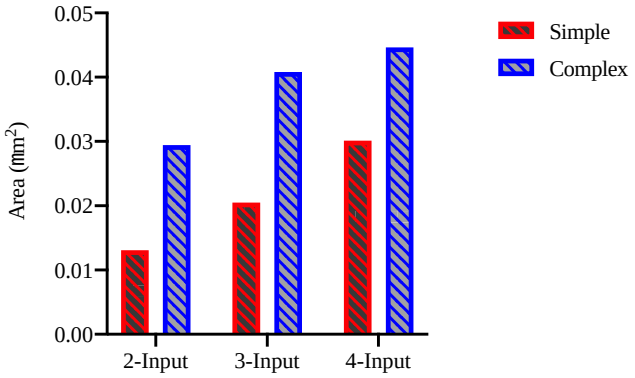
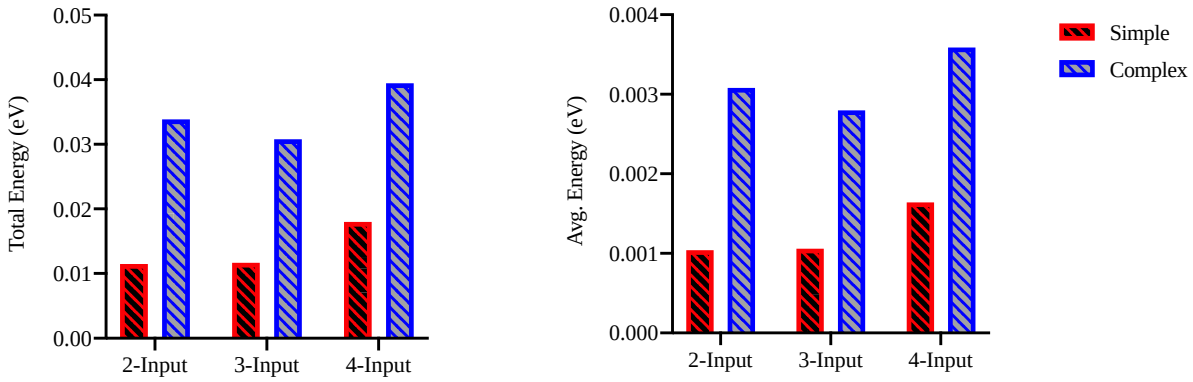


Figure 14. Area comparison of the proposed structures.



(a) (b)

Figure 15. Energy dissipation of the proposed structures: (a) Total energy, (b) Average energy.

In order to evaluate the robustness of the proposed C-element structures, the QCADesigner-FS simulator has been used [37]. The QCADesigner-FS is a modified version of the QCADesigner tool with a capability of simulating the behavior of QCA-based structures under different structural defects. The

rationale behind this simulator is to inject a particular defect into the structure (Vacancy, Interstitial, Dopant or Dislocation) and compare the output of the defective structure to that of the defect-free structure. The structure is said to be error-free if its output in the presence of a defect matches its correct output. This process is repeated for an adequate number of simulations. The robustness of a design is quantified in terms of the percent of the error-free simulations from the total number of simulations. Figure 16 demonstrates the robustness of the various designs under different structural defects.

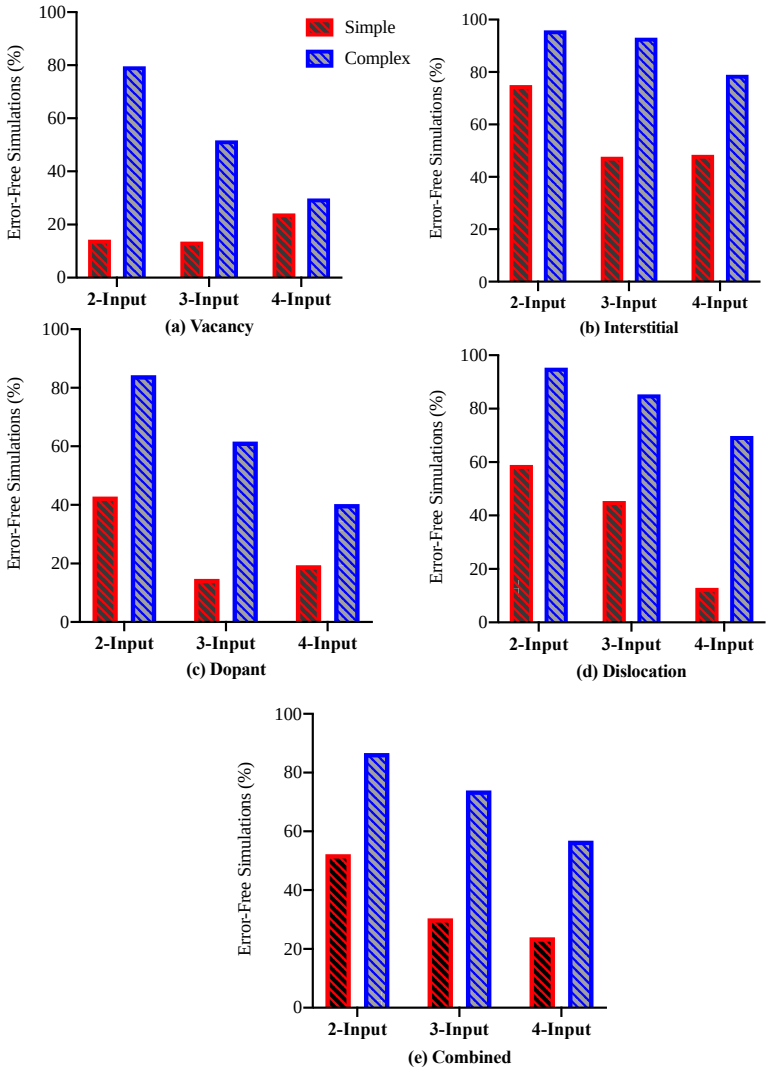


Figure 16. Robustness analysis of the proposed structures.

In Figure 16a, the robustness analysis is carried out under the vacancy (i.e., cell missing) structural defect. As shown, the proposed complex structures have higher robustness when compared to their simple counterparts. On the other hand, Figure 16b compares the robustness against interstitial defects for both simple and complex structures. It can be seen that the complex structures exhibit higher immunity against interstitial defects as compared to the simple ones. Similar trends can be observed for the dopant and dislocation structural defects, as shown in Figure 16c and 16d, respectively. Figure 16e shows the robustness analysis results under combined structural defects. The observations that can be drawn from this figure are two-fold. First, the robustness of the simple structures as well as the complex structures decreases as the number of inputs of the C-element is increased. Second, the percentages of enhancement achieved by the complex C-element structures as compared to their simple counterparts are 66%, 143% and 137% for the 2- , 3- and 4-input structures, respectively. Table 5 summarizes and compares the proposed structures in terms of cell count, area and energy dissipation, while Table 6 compares their robustness under different structural defects.

Table 5. Comparison of the proposed structures in terms of cell count, area and energy dissipation.

Structure	Inputs	Cell count	Area (μm^2)	Total Energy (eV)	Average Energy (eV)
Simple	2	14	0.013	1.15e-002	1.04e-003
	3	22	0.021	1.17e-002	1.06e-003
	4	31	0.030	1.80e-002	1.64e-003
Complex	2	43	0.029	3.39e-002	3.08e-003
	3	58	0.041	3.08e-002	2.80e-003
	4	67	0.045	3.95e-002	3.59e-003

Table 6. Comparison of the proposed structures in terms of robustness against different structural defects.

Structure	Inputs	Vacancy (%)	Interstitial (%)	Dopant (%)	Dislocation (%)	Combined (%)
Simple	2	14.29	75.00	42.86	58.93	52.23
	3	13.64	47.73	14.77	45.45	30.40
	4	24.19	48.39	19.35	12.90	23.96
Complex	2	79.65	95.93	84.30	95.35	86.63
	3	51.72	93.10	61.64	85.34	73.92
	4	29.85	78.95	40.30	69.78	56.83

It is worth noting that the simple structures provide easier reachability to the output cell while suffering from low immunity against structural defects. On the other hand, the complex structures achieve higher immunity against structural defects at the expense of requiring either coplanar or multilayer crossover wiring techniques to reach the output cell [38]-[39]. Moreover, the proposed QCA-based C-element structures have significant improvements in terms of area and energy dissipation as compared to previously reported CMOS-based C-element designs [40].

4. CONCLUSION

In this paper, 2-, 3- and 4-input QCA-based C-element structures were proposed and evaluated in terms of their functional correctness, area, energy dissipation and robustness against structural defects. The proposed structures can be classified as either simple or complex designs. Whereas simple structures have resulted in lower area and energy dissipation with up to 56% and 66% improvement, respectively, complex ones have shown significant immunity against structural defects and achieved up to 143% improvement when compared to simple structures. In addition, the number of inputs has a pronounced impact on the considered evaluation parameters. Ultimately, the proposed structures can serve as a basis for further research in the asynchronous circuits design.

REFERENCES

[1] N. B. Bousari, M. K. Anvarifard and S. Haji-Nasiri, "Improving the Electrical Characteristics of Nanoscale Triple-gate Junctionless Finfet Using Gate Oxide Engineering," AEU International Journal of Electronics and Communications, vol. 108, pp. 226 – 234, 2019.

[2] A. Razavieh, P. Zeitzoff and E. J. Nowak, "Challenges and Limitations of CMOS Scaling for FinFet and Beyond Architectures," IEEE Transactions on Nanotechnology, vol. 18, pp. 999–1004, 2019.

[3] W. Sung and Y. Li, "DC/AC/RF Characteristic Fluctuations Induced by Various Random Discrete Dopants of Gate-all-around Silicon Nanowire N-MOSFETs," IEEE Transactions on Electron Devices, vol. 65, no. 6, pp. 2638–2646, June 2018.

[4] D. E. Nikonov and I. A. Young, "Benchmarking of Beyond-CMOS Exploratory Devices for Logic Integrated Circuits," IEEE Journal on Exploratory Solid-State Computational Devices and Circuits, vol. 1, pp. 3–11, 2015.

[5] N. K. Chaubey and B. B. Prajapati, "Quantum Cryptography and the Future of Cyber Security," Hershey, PA, USA, pp. 1–343, 2020.

[6] B. Debnath, J. C. Das, D. De, S. P. Mondal, A. Ahmadian, M. Salimi and M. Ferrara, "Security Analysis with Novel Image Masking Based Quantum-dot Cellular Automata Information Security Model," IEEE Access, vol. 8, pp. 117 159–117 172, 2020.

[7] C. S. Lent, P. D. Tougaw, W. Porod and G. H. Bernstein, "Quantum Cellular Automata," Nanotechnology, vol. 4, no. 1, pp. 49–57, Jan. 1993.

[8] H. Adepuand and I. S. Rao, "Quantum-dot Cellular Automata Technology for High-speed High-data-rate Networks," Circuits, Systems and Signal Processing, vol. 38, no. 11, pp. 5236–5252, Nov. 2019.

[9] H. M. H. Babu, Quantum Computing: A pathway to quantum logic design, IOP Publishing, [Online], Available: http://dx.doi.org/10.1088/978-0-7503-2747-3, 2020.

[10] M. Gao, J. Wang, S. Fang, J. Nan and L. Daming, "A New Nano Design for Implementation of a Digital Comparator Based on Quantum-dot Cellular Automata," International Journal of Theoretical Physics, May 2020.

[11] A. N. Bahar and K. A. Wahid, "Design and Implementation of Approximate DCT Architecture in Quantum-dot Cellular Automata," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, pp. 1–10, (Early Access), 2020.

[12] Z. Song, G. Xie, X. Cheng, L. Wang and Y. Zhang, "An Ultra-low Cost Multilayer RAM in Quantum-dot Cellular Automata," IEEE Transactions on Circuits and Systems II: Express Briefs, (Early Access), pp. 1–1, 2020.

[13] M. Goswami, A. Mondal, M. H. Mahalat, B. Sen and B. K. Sikdar, "An Efficient Clocking Scheme for Quantum-dot Cellular Automata," International Journal of Electronics Letters, vol. 8, no. 1, pp. 83–96, 2020.

[14] R. Laajimi, "Nanoarchitecture of Quantum-dot Cellular Automata (QCA) Using Small Area for Digital Circuits," Chapter 3 in Book: Advanced Electronic Circuits-Principles, Architectures and Applications on Emerging Technologies, IntechOpen, 2018.

[15] M. Raj, L. Gopalakrishnan and S.-B. Ko, "Design and Analysis of Novel QCA Full Adder-subtractor," International Journal of Electronics Letters, vol. 0, no. 0, pp. 1–14, [Online], Available: https://doi.org/10.1080/21681724.2020.1726479, 2020.

[16] A. H. Majeed, M. S. B. Zainal, E. Alkaldy and D. M. Nor, "Full Adder Circuit Design with Novel Lower Complexity XOR Gate in QCA Technology," Transactions on Electrical and Electronic Materials, vol. 21, no. 2, pp. 198–207, Apr. 2020.

[17] D. Abedi and G. Jaberipur, "Decimal Full Adders Specially Designed for Quantum-dot Cellular Automata," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 65, no. 1, pp. 106–110, 2018.

[18] G. Cocorullo, P. Corsonello, F. Frustaci and S. Perri, "Design of Efficient BCD Adders in Quantum-dot Cellular Automata," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 64, no. 5, pp. 575–579, 2017.

[19] L. Xingjun, S. Zhiwei, C. Hongping and M. R. J. Haghighi, "A New Design of QCA-based Nanoscale Multiplexer and Its Usage in Communications," International Journal of Communication Systems, vol. 33, no. 4, p. e4254, 2020.

[20] J.-C. Jeon, "Designing Nanotechnology QCA–multiplexer Using Majority Function-based NAND for Quantum Computing," The Journal of Supercomputing, DOI: https://doi.org/10.1007/s11227-020-03341-8, May 2020.

[21] T. N. Sasamal, A. K. Singh and A. Mohan, "Design of Registers and Memory in QCA," Proc. of the Quantum-dot Cellular Automata Based Digital Logic Circuits: A Design Perspective, Part of the Studies in Computational Intelligence Book Series, vol. 879, pp 119-137, Springer, 2020.

[22] A. Sadhu, K. Das, D. De and M. R. Kanjilal, "Area-delay-energy Aware SRAM Memory Cell and m x n Parallel Read/write Memory Array Design for Quantum-dot Cellular Automata," Microprocessors and Microsystems, vol. 72, p. 102944, 2020.

[23] M. Patidar and N. Gupta, "An Efficient Design of Edge-triggered Synchronous Memory Element Using Quantum-dot Cellular Automata with Optimized Energy Dissipation," Journal of Computational Electronics, vol. 19, no. 2, pp. 529–542, Jun. 2020.

[24] A. N. Bahar and K. A. Wahid, "Design of an Efficient n x n Butterfly Switching Network in Quantum-dot Cellular Automata (QCA)," IEEE Transactions on Nanotechnology, vol. 19, pp. 147–155, 2020.

[25] A. N. Bahar and K. A. Wahid, "Design of QCA-serial Parallel Multiplier (QSPM) with Energy Dissipation Analysis," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 67, no. 10, pp. 1939–1943, 2020.

[26] R. Marshal, G. Lakshminarayanan, S. B. Ko, N. Naganathan and N. Ramasubramanian, "Configurable Logic Blocks and Memory Blocks for Beyond CMOS FPGA Based Embedded Systems," IEEE Embedded Systems Letters, pp. 1–1, 2020.

[27] S.-S. Ahmadpour, M. Mosleh and S. Rasouli Heikalabad, "Robust QCA Full-adders Using An Efficient Fault-tolerant Five-input Majority Gate," International Journal of Circuit Theory and Applications, vol. 47, no. 7, pp. 1037–1056, 2019.

[28] Z. Tabassam, S. R. Naqvi, T. Akram, M. Alhussein, K. Aurangzeb and S. A. Haider, "Towards Designing Asynchronous Microprocessors: From Specification to Tape-out," IEEE Access, vol. 7, pp. 33978–34003, 2019.

[29] B. Sparkman, S. C. Smith and J. Di, "Built-in Self-test for Multi-threshold Null Convention Logic Asynchronous Circuits," Proc. of the 38th IEEE VLSI Test Symposium (VTS), pp. 1–6, San Diego, USA, 2020.

[30] A. Motaqi, M. Helaoui, S. Aghli Moghaddam and M. R. Mosavi, "Detailed Implementation of Asynchronous Circuits on Commercial FPGAs," Analog Integrated Circuits and Signal Processing, vol. 103, no. 3, pp. 375–389, Jun. 2020.

[31] R. Ezz Eldin, M. A. El Moursy and H. F. A. Hamed, "Synchronous and Asynchronous NoC Design Under High Process Variation," Analysis and Design of Network-on-Chip under High Process Variation, Cham: Springer International Publishing, pp. 71–86, 2015.

[32] J. Spars and S. Furber, Principles of Asynchronous Circuit Design: A Systems Perspective, 1st Ed., Springer Publishing Company, Incorporated, 2010.

[33] A. Yakovlev, K. Gardiner and A. Bystrov, "A C-element Latch Scheme with Increased Transient Fault Tolerance for Asynchronous Circuits," Proc. of the 13th IEEE International On-Line Testing Symposium (IOLTS 07), pp. 223–230, Los Alamitos, CA, USA, 2007.

[34] A. de Gennaro, D. Sokolov and A. Mokhov, "Design and Implementation of Reconfigurable Asynchronous Pipelines," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 28, no. 6, pp. 1527–1539, 2020.

[35] K. Walus, T. J. Dysart, G. A. Jullien and R. A. Budiman, "QCADesigner: A Rapid Design and Simulation Tool for Quantum-dot Cellular Automata," IEEE Transactions on Nanotechnology, vol. 3, no. 1, pp. 26–31, 2004.

[36] F. Sill Torres, R. Wille, P. Niemann and R. Drechsler, "An Energy-aware Model for the Logic Synthesis of Quantum-dot Cellular Automata," IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems, vol. 37, no. 12, pp. 3031–3041, 2018.

[37] D. Reis and F. Sill Torres, "A Defects Simulator for Robustness Analysis of QCA Circuits," Journal of Integrated Circuits and Systems, vol. 11, pp. 86–96, Aug. 2016.

[38] M. Raj and L. Gopalakrishnan, "Cost Efficient Subtractor Designs in QCA," Proc. of the International Conference on Electronics and Sustainable Communication Systems (ICESC), pp. 1168–1172, Coimbatore, India, 2020.

[39] J. Maharaj and S. Muthurathinam, "Efficient Majority Logic Subtractor Design Using Multilayer Crossover in Quantum-dot Cellular Automata," Journal of Nanophotonics, vol. 14, no. 3, pp. 1 – 10.

[40] N. Soufi and S. C. Smith, "Analysis and Design of CMOS Resettable C-elements," Proc. of the 60th IEEE International Midwest Symposium on Circuits and Systems (MWSCAS), pp. 104–107, Boston, USA, 2017.

ملخص البحث:

إنّ الطلب المستمر في السوق على أنظمةٍ حاسوبيةٍ عالية الأداء وفعالة من حيث استهلاك الطاقة كان من شأنه أن يقود التّقنيات الحاسوبية في اتجاه آليّات التّشغيل الذاتي الخلويّة المستندة على النّقط الكمية (QCA) في نطاق تكنولوجيا النانو.

تقدم هذه الورقة بِنىً جديدةً لعنصر سي (بسيطة ومعقدة) قائمة على آليات التشغيل الذاتي الخلوية المستندة على النقط الكمية. وقد تم تحليل البِنى المقترحة بالتفصيل بناءً على متغيرات تصميمية أساسية ، مثل: المساحة، واستهلاك الطاقة، والحصانة ضد العيوب البِنيوية.

وقد أظهرت نتائج المحاكاة أن البِنى البسيطة المقترحة حققت تحسيناً وصل الى ما نسبته 56% و 66% فيما يتصل بالمساحة واستهلاك الطاقة، على الترتيب. من جهة أخرى، أظهرت البِنى المعقدة المقترحة حصانة راسخة ضد العيوب البِنيوية، محققة تحسيناً وصل الى 143% مقارنة بالبِنى البسيطة. ويمكن النظر الى بِنى عنصر سي المقترحة بوصفها وحداتٍ مهمة نافعة للتصاميم غير المتزامنة.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 07, No. 01, March 2021.

PHIBOOST- A NOVEL PHISHING DETECTION MODEL USING ADAPTIVE BOOSTING APPROACH

Ammar Odeh[1], Ismail Keshta[2] and Eman Abdelfattah[3]

(Received: 14-Sep.-2020, Revised: 6-Nov.-2020, Accepted: 15-Dec.-2020)

ABSTRACT

Every day, cyberattacks increase and use different strategies. One of the most common cyberattacks is Phishing, where the attacker collects sensitive and confidential information by pretending as a trusted party. Different traditional strategies have been introduced for anti-phishing, such as blacklisted, heuristic search and visual similarity. Most of these traditional methods have a high false rate and take a long time to detect the phishing website. New modes have been introduced using machine learning techniques which improve the detection's accuracy. Machine learning techniques require a huge amount of data called features that are collected from different websites. These collected features are classified into four categories. This paper introduces a novel detection model by utilizing features' selection to pick up the highly correlated features with the class label. The phase of features' selection employs independent significance features library from MATLAB and heat-map from Python to find the highly correlated features. Then, the proposed model uses an adaptive boosting approach which consists of multiple classifiers to increase the model's accuracy. The proposed model produces an extremely high predictive accuracy of approximately 99%.

KEYWORDS

Adaptive boost, Feature selection, Correlation-based feature, Machine learning.

1. INTRODUCTION

Phishing is against the law. It uses social engineering and technical trick to thieve Internet users' non-public identity facts and financial account credentials. Social engineering schemes prey on unwary sufferers with the means of not only fooling them into believing they're managing a trusted and a legitimate party, but also using misleading electronic mail addresses and electronic mail messages [1]-[2].

Disasters have continually been a good chance for different types of criminals' special cyberattacks. The phishers have created violations to take advantage of hurricanes, recessions and different challenging times, merchandising fake charitable giving possibilities and nonexistent services or products. One of the most recent world catastrophes in 2020 is the COVID-19 pandemic. Anti-Phishing Working Group (APWG) classifies four cybercriminal methods that represent more complicated scenarios to lure their victims [3]-[4].

Several types of research introduced phishing attack problems and their consequences on customer trust in e-commerce and online services [5]. The phishing attackers create a website that pretends as a trusted website to collect valuable and sensitive Internet user information. At the same time, different anti-phishing software models for phishing detections are introduced. The phishing detection strategies are classified into seven categories [6] as follows:

1. User education: this category depends on the educated Internet users to distinguish between a legitimate and a phishing website [7].
2. Create a blacklist: this strategy creates centralized phishing websites and compares an URL with the list to find out if the URL is legitimate or not [8].
3. Heuristic blacklist methods: in this strategy, the system identifies the signature of the phishing URL and blacklists it for the future use of intrusion detection systems [9].
4. Visual similarity: These techniques use URL features to find out the similarity between websites (page source code, images, textual content, text formatting, HTML tags, CSS, website logo).

1. A. Odeh is with Computer Science Department, Princess Sumaya Uni. for Technology, Amman, Jordan. Email: a.odeh@psut.edu.jo
2. I. Keshta is with Comp. Sci. and Information Systems Department, AlMaarefa Uni., Riyadh, KSA. Email: imohamed@mcst.edu.sa
3. E. Abdelfattah is with School of Theoretical & Applied Science, NJ, USA.

After that, the system compares the new website with previously visited ones and distinguishes whether it is a legitimate or a phishing website [10].

5. Search engine-based techniques: in this mode, the system uses the search engine and extracts the website features, then checks the website legitimacy. However, the search engine does not give precise output for the non-English search query [11].
6. Supervised Machine Learning detection system uses supervised machine learning models on phishing datasets with predefined features [12].
7. Deep learning techniques: these techniques include Gated Recurrent Neural Network (GRU) and Convolutional Neural Network (CNN). Based on these techniques, the system automatically extracts the features from generic URL, file directory, ...etc. [13].

Table 1 shows a summary of phishing detection strategies and their main drawbacks.

Table 1. Phishing detection strategies.

	Phishing detection strategies	Problem
1	User education	• Fail to detect a new phishing attack
2	Create a blacklist	• Produce high false positive rate
3	Heuristic blacklist methods	
4	Visual similarity	• Complicated • Slow in nature
5	Search engine-based techniques	• Not fit for real-time environment • Language dependence
6	Supervised machine learning detection	• The achieved performance depends on the features' selection and the classification algorithms
7	Deep learning techniques	

The rest of the paper is organized as follows: In Section 2, a review of related work is presented. In Section 3, the proposed methodology is described. In Section 4, the experimental results are reported. The conclusion of the paper is included in Section 5.

2. LITERATURE REVIEW

Different research papers have conducted an intensive work on website security, some of which manipulated the routing security [14], while others dealt with intrusion detection, intrusion prevention and smart grid security [15].

Pawan Parakash et al. proposed two methods to identify phishing websites, where first proposed method introduced five heuristics to enumerate the combination of the known phishing websites to find out the new phishing websites. The second method used matching algorithms to find out the new phishing websites [16].

Samuel Marchal et al. analyzed and evaluated the URL of the websites and extracted the features of the URL. Based on the several queries through Google and Yahoo search engines, the authors determined the keywords for each website. Then, the keywords with the extracted features are used in a machine learning classification algorithm to find out the phishing websites from the real dataset [17]. In [18], the authors introduced models using machine learning and data mining algorithms for detecting website phishing.

The authors in [19] used the artificial neural network to spot phishing websites. The proposed work used 17 neurons as input for 17 characteristics and one hidden layer level and two neurons as output to decide whether or not the website is phishing. The dataset was divided into 80 percent as a training set and 20 percent as a test set. The suggested model achieved 92.48 percent accuracy.

Authors in [20] introduced a model relying on a machine learning technique called PLIFER. This model requires an age of the URL domain. Also, ten features are extracted and Random Forest (RF) model is

used to identify the phishing website. 96 percent of phishing e-mails were correctly identified by this model. Classification models are also used to identify phishing utilizing labeled datasets. Different classification methods used features, like URL-based and text-based applications.

A proposed software collection model hybrid set of features (HEFS) to identify phishing websites relying on machine learning algorithms is presented in [21]. A cumulative distribution gradient technique is used to extract the primary feature set. Then, the second set of features is extracted using a method called data perturbation ensemble. Random Forest (RF), an ensemble learner, is subsequently implemented to identify phishing websites. The results indicated that HEFS identified phishing features with a precision of up to 94.6 percent.

In 0, The authors selected the most suitable components to identify website phishing and proposed two new selection methods or detection techniques based on machine learning algorithms. The two methods include the AdaBoost classifier and the LightGBM classifier. When combined, they form a hybrid classifier. These two algorithms have proved to be effective and efficient in improving the accuracy of single classifiers in detecting web phishing attacks.

In 0, The authors investigated agreeing on the final conclusion of the features used to detect phishing on webpages. Using three standard datasets, the authors used the Fuzzy Rough Set (FRS) theory as a tool to select the most significant features to identify intrusion on webpages. The chosen features were then fed into three standard classifiers to detect phishing. When Random Forest classification was used, the maximum accuracy gained by Fuzzy Rough Set (FRS) feature selection was 95%. The Fuzzy Rough Set (FRS) had used three sets of data to come up with nine universal features of detecting phishing. When these versatile features were used to measure the accuracy value, the accuracy was about 93%, which is comparable to the Fuzzy Rough Set performance, with only a slight difference of 2%.

The authors of 0 proposed three ensemble learning models based on Forest Penalizing Attributes (Forest PA) algorithm. The algorithm exploited the prowess of all attributes in a given set of data using a weight increment and weight assignment strategy to build highly resourceful decision trees. The results of the experiment showed highly efficient meta-learners with an accuracy of 96.26%.

3. MOTIVATION AND MAIN CONTRIBUTION

All phishing attacks have some salient features; however, these attacks exhibit some similarities and patterns. Thus, using machine learning methods to detect these similar patterns and recognize phishing websites has become possible 00.

In this paper, an inventive detection model is introduced that utilizes feature selection to pick up similar features on phishing websites with the class label. The independent significant features library from MATLAB and heat-map from python are employed in the features' selection to find the associated features on phishing websites. The proposed novel detection model consists of multiple classifiers incorporated in an adaptive boosting technique to increase the model's accuracy.

The adaptive AdaBoost classifier is selected as an efficient technique for detecting website phishing, because it is flexible and straightforward, yet it has a high generalization performance 0-0. The fact that it is based on several weak classifiers makes it flexible and straightforward to implement. Also, it doesn't use large sets of features that may be unnecessary sometimes, but it treats each class's attributes separately 00. Moreover, the AdaBoost classifier achieves much high accuracy, as it regulates the errors of weak classifiers; therefore, it needs much fewer settings as compared to other robust classifiers 0-0.

4. PRELIMINARIES

This section provides a brief description of the phishing dataset used in the experimental comparison, as well as a background about the dataset, feature selection and the classification model used in this study.

4.1 Dataset

The dataset used is collected from the PhishTank archive [22], MillerSmiles archive [23] and Google searching operators. The phishing dataset consists of 30 features, as listed in Table 2. All of these features were classified into four categories: Address Bar Features (1-12), Abnormal Based Features

(13-18), HTML and JavaScript-based Features (19-23) and Domain-based Features (24-30). The last feature is the label column, which represents the class of the website as either phishing or legitimate.

Table 2. Feature classes of the dataset.

Feature class	Description
Address Bar	Feature of Uniform Resource Locator (URL) such as IP address
Abnormal Based	Feature of abnormal activities such as URL of tag (Anchor)
HTML and JavaScript-based	Feature of HTML and Jscript embedded in the page source code
Domain-based	Feature of third party

For example, the feature number 28 is Google_Index, which examines whether a website is in Google's index or not.

$$\text{Rule: IF}\begin{cases}\text{Webpage Indexed by Google} \rightarrow \text{Legitimate}\\ \text{Otherwise} \rightarrow \text{Phishing}\end{cases}$$

4.2 Feature Selection

A subset of features that work well together is selected. The selection process aims to minimize the time needed to build the machine learning model and produce high accuracy. Selection features' process keeps features that have low correlation to each other, but have high correlation to the label feature [28]. The rest of the highly correlated features are dropped.

Table 3. URL features.

#	Feature name	#	Feature name
1	having_IP_Address	17	Submit_to_email
2	URL_Length	18	Abnormal_URL
3	Shortining_Service	19	Redirect
4	having_At_Symbol	20	on_mouseover
5	double_slash	21	RightClick
6	Prefix_Suffix	22	popUpWidnow
7	having_Sub_Domain	23	Iframe
8	SSLfinal_State	24	age_of_domain
9	Domain_registeration	25	DNSRecord
10	Favicon	26	web_traffic
11	port	27	Page_Rank
12	HTTPS_token	28	Google_Index
13	Request_URL	29	Links_pointing
14	URL_of_Anchor	30	Statistical_report
15	Links_in_tags	31	Result
16	SFH		

4.3 Adaptive Boosting

AdaBoosting is the decision tree on binary classification problems. AdaBoosting is usually used for a discrete dataset, so it's more related to classification than to regression. The AdaBoosting algorithm updates the weight to minimize error, which leads to minimize the misclassification rate. It is necessary to highlight that Freund, Schapire and Abe 0 developed the AdaBoost algorithm to increase the efficiency of binary classifiers. AdaBoost uses an ensemble learning method approach to learn from weak classifiers' mistakes and turn them into strong ones. AdaBoost generates a weak learner through

primary training data. The data is then adjusted according to the foreseen performance for the next round of weak learner training. It is good to note that the training samples with the lowest predicting accuracy in the preceding step are approached with more attention in the step that follows. The weak learners with different weights are finally combined to create a strong learner 0-0.

5. PROPOSED MODEL

Figure 1 shows the system's flow diagram to recognize the URL. The proposed system reads the URL from the dataset, then the URL is classified into multidimensional features according to the dataset components. The model's detection accuracy is improved by selecting the most correlated features and eliminating the irrelevant features. The filtered data is split into the training set and testing data. Machine learning model is applied by using an adaptive boost classifier to create the adaptive boost knowledge base. The testing dataset is used as the input for the detection model to evaluate it.

The proposed model uses Weka 3.6, Python and MATLAB. Table 4 shows the experimental parameters, such as the evaluator, the search algorithm and the batch size, the classifier, the number of iterations and the weight threshold.

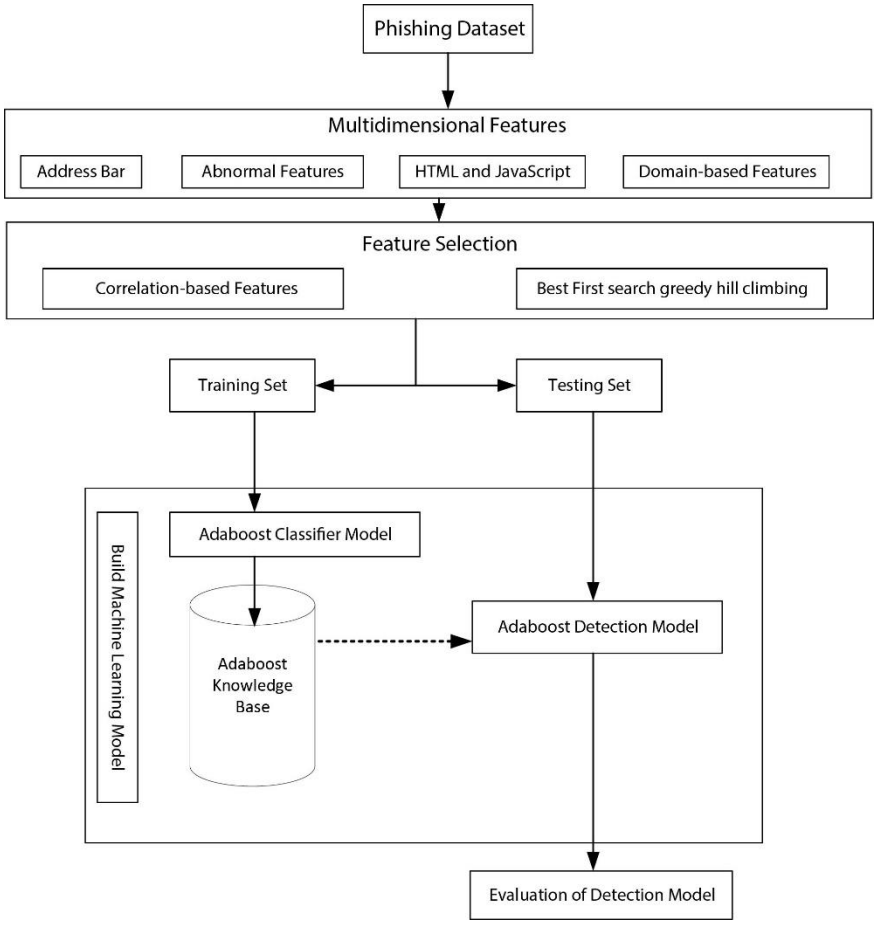


Figure 1. PhiBoost structure.

Table 4. Experimental parameters.

Feature Selection	
Parameters	Value
Evaluator	Correlation-based Features
Search model	Best First search greedy hill-climbing
Adaptive Boost Classifier	
Parameters	Value
Batch size	100
Classifier	Decision Stump
Number of iterations	10
Weight threshold	100

6. DISCUSSION OF RESULTS

The proposed model classifies the features into four categories by utilizing the correlation relationship between features and the class label (phishing or legitimate).

The output from the feature selection process is nine features as follows: having_IP_Address, having_Sub_Domain, SSLfinal_State, web_traffic, Google_Index, Request_URL, URL_of_Anchor, Links_in_tags and SFH. In the next feature selection phase, MATLAB built-in procedure called independent significance features test (IndFeat()) is invoked. Figure 2 shows the Python heat map of the output of the independent significance features test.

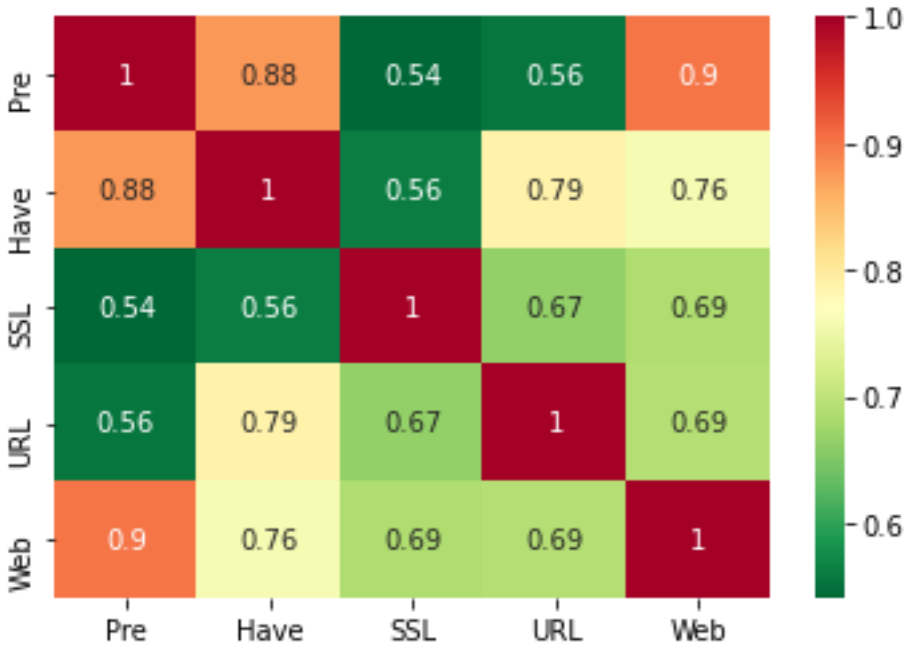


Figure 2. Heat map after applying attribute selector.

Four popular statistical measures were utilized to determine the efficiency of the proposed model. Table 5 lists these performance measures and their effects on the model performance. In our experiments, we evaluate the proposed system by using the accuracy to evaluate the ratio of correctly predicated observations to the total observations of the proposed system. Precision measure enables us to evaluate the ratio of correctly predicated observations to the total of positive observations. The recall measure evaluates the ratio of correctly predicated positive observations to all observations in the actual class. F-measure is a weighted average precision and recall.

Table 5. Popularly statistical measures.

Statistical measures	Formula
Precision	$\frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$
Recall	$\frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$
Accuracy	$\frac{\text{True Positive+True Negative}}{\text{Total Number of Instance}}$
F-measure	$2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$

Table 6 shows the experiments conducted on a different percentage split. The minimum accuracy achieved in the proposed model is 97.7% and the F-measure is 97.5% after training the model in 50%

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 07, No. 01, March 2021.

of the dataset. The best performance is obtained when the training percentage is 70%, where both accuracy and F-measure are approximately 99%.

Table 6. The performance of the proposed algorithm.

Experiment #	Training Percentage	Precision	Recall	Accuracy	F-measure
1	50 %	97.8 %	97.1 %	97.7 %	97.5 %
2	60 %	98.2 %	97.6 %	98.1 %	97.9 %
3	70 %	99.0 %	98.6 %	98.9 %	98.8 %
4	80 %	98.4 %	97.8 %	98.3 %	98.1 %
5	90 %	98.8 %	98.2 %	98.7 %	98.5 %

Figure 3 shows the efficiency of the PhiBoost model which explores the precision and accuracy with different percentages of training and testing to avoid any overfitting problem. The minimum accuracy that PhiBoost achieved was when the training test is 50% of the dataset. On the other side, the performance of the PhiBoost model improves if the training set is 70%.

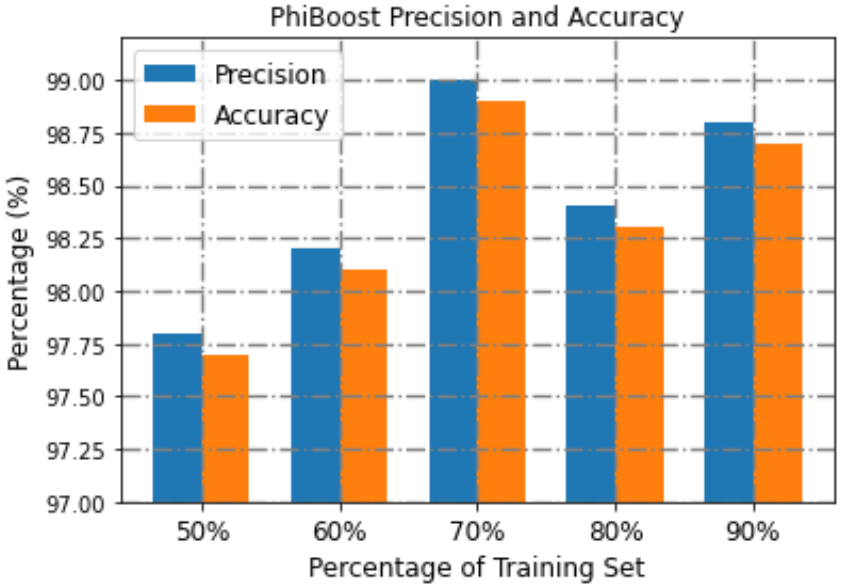


Figure 3. PhiBoost precision and accuracy.

In Table 7, the proposed model is compared with different detection machine learning models. As demonstrated in the results obtained, the proposed model enhances the accuracy of the detection system. In [27], the authors introduced a phishing detection model by utilizing feature selection and combining

Table 7. Comparison with the PhiBoost model.

Paper	Machine learning algorithm	Accuracy
[14]	NN	94.07%
[15]	multi-label rule-based	94.8%
[18]	NN	84%
[19]	FFNN	92.48%
[21]	Feed-forward NN	97.40%
[24]	Logistic regression classifier	98.40%
[25]	Naïve Bayesian classifier	90%
[26]	HNB and J48	96.25%
[27]	Multilayer perceptron neural network	98.5%
PhiBoost model		98.9 %

as a pre-processing step for the dataset. After that, they employed a multilayer perceptron neural network as a classifier function. In our proposed work, we tried to optimize the accuracy by minimizing the number of selected features and utilizing the adaptive boosting classifier.

7. CONCLUSION

This paper aims to introduce an outstanding solution to the threat of phishing in our modern community. As a result, this research proposed implementing feature selection and adaptive boosting for an efficient model for detecting phishing websites. The results of this study explored the best splitting rate for the dataset to train the machine learning model, which was 70%. The results achieved a high accuracy and a high F-measure with high predictive capability as well as with low false-positive rates and low false-negative rates. The proposed model minimizes the time to build the training model by picking up the most correlated features and produces an extremely high predictive accuracy of approximately 99%. Conclusively, the application of the implemented methods of this research in a real-time environment remains pivotal in future work. In the future, the system's capability will be investigated by testing it over a real-time environment.

REFERENCES

[1] G. Varshney, M. Misra and P. K. Atrey, "A Survey and Classification of Web Phishing Detection Schemes," Security and Communication Networks, vol. 9, pp. 6266-6284, 2016.

[2] A. Aleroud and L. Zhou, "Phishing Environments, Techniques and Countermeasures: A Survey," Computers & Security, vol. 68, pp. 160-196, 2017.

[3] C. Singh, "Phishing Website Detection Based on Machine Learning: A Survey," Proc. of the 6th IEEE International Conference on Advanced Computing and Communication Systems (ICACCS), pp. 398-404, Coimbatore, India, 2020.

[4] L. Jelovčan, S. L. Vrhovec and A. Mihelič, "A Literature Survey of Security Indicators in Web Browsers," Elektrotehniski Vestnik, vol. 87, pp. 31-38, 2020.

[5] Y. Al-Hamar, H. Kolivand and A. Al-Hamar, "Phishing Attacks in Qatar: A Literature Review of the Problems and Solutions," Proc. of the 12th IEEE International Conference on Developments in eSystems Engineering (DeSE), 2019, pp. 837-842, Kazan, Russia, 2019.

[6] M. Sánchez-Paniagua, E. Fidalgo, V. González-Castro and E. Alegre, "Impact of Current Phishing Strategies in Machine Learning Models for Phishing Detection," Proc. of the 13th International Conference on Computational Intelligence in Security for Information Systems (CISIS 2020), Part of the Advances in Intelligent Systems and Computing Book Series (AISC), vol. 1267, pp. 87-96, 2020.

[7] A. S. Onashoga, O. E. Ojo and O. O. Soyombo, "Securix: A 3D Game-based Learning Approach for Phishing Attack Awareness," Journal of Cyber Security Technology, vol. 3, pp. 108-124, 2019.

[8] K. Hynek, T. Čejka, M. Žádník and H. Kubátová, "Evaluating Bad Hosts Using Adaptive Blacklist Filter," Proc. of the 9th IEEE Mediterranean Conf. on Emb. Comp. (MECO), pp. 1-5, Budva, Montenegro, 2020.

[9] S. Sarika, "A Heuristic Model to Detect Malicious URLs Using Case-based Reasoning," Journal of Information and Computational Science, vol. 9, no. 11, pp. 1066–1079, 2019.

[10] S. Abdelnabi, K. Krombholz and M. Fritz, "VisualPhishNet: Zero-Day Phishing Website Detection by Visual Similarity," Proc. of the ACM SIGSAC Conference on Computer and Communications Security (CCS '20), pp. 1681–1698, [Online], Available: https://doi.org/10.1145/3372297.3417233, Oct. 2020.

[11] B. B. Gupta and A. K. Jain, "Phishing Attack Detection Using a Search Engine and Heuristics-based Technique," Journal of Information Technology Research (JITR), vol. 13, pp. 94-109, 2020.

[12] E. Anthi, L. Williams, M. Słowińska, G. Theodorakopoulos and P. Burnap, "A Supervised Intrusion Detection System for Smart Home IoT Devices," IEEE Internet of Things J., vol. 6, pp. 9042-9053, 2019.

[13] B. Wei, R. A. Hamad, L. Yang, X. He, H. Wang, B. Gao and W. L. Woo, "A Deep Learning-driven Light-weight Phishing Detection Sensor," Sensors, vol. 19, p. 4258, 2019.

[14] P. Yi, Y. Guan, F. Zou, Y. Yao, W. Wang and T. Zhu, "Web Phishing Detection Using a Deep Learning Framework," Wireless Communications and Mobile Computing, vol. 2018, [Online], available: https://doi.org/10.1155/2018/4678746, 2018.

[15] T. Alves, R. Das and T. Morris, "Embedding Encryption and Machine Learning Intrusion Prevention Systems on Programmable Logic Controllers," IEEE Embedded Sys. Letters, vol. 10, pp. 99-102, 2018.

[16] P. Prakash, M. Kumar, R. R. Kompella and M. Gupta, "PhishNet: Predictive Blacklisting to Detect Phishing Attacks," Proceedings of IEEE INFOCOM, pp. 1-5, San Diego, USA, 2010.

[17] S. Marchal, J. François, R. State and T. Engel, "PhishStorm: Detecting Phishing with Streaming Analytics," IEEE Transactions on Network and Service Management, vol. 11, pp. 458-471, 2014.

[18] A. Subasi, E. Molah, F. Almkallawi and T. J. Chaudhery, "Intelligent Phishing Website Detection Using Random Forest Classifier," Proc. of the IEEE International Conference on Electrical and Computing Technologies and Applications (ICECTA), pp. 1-5, Ras Al Khaimah, United Arab Emirates, 2017.

[19] S. Smadi, N. Aslam and L. Zhang, "Detection of Online Phishing Email Using Dynamic Evolving Neural Network Based on Reinforcement Learning," Decision Support Systems, vol. 107, pp. 88-102, 2018.

[20] N. Abdelhamid, F. Thabtah and H. Abdel-jaber, "Phishing Detection: A Recent Intelligent Machine Learning Comparison Based on Models Content and Features," Proc. of the IEEE International Conference on Intelligence and Security Informatics (ISI), pp. 72-77, Beijing, China, 2017.

[21] K. L. Chiew, C. L. Tan, K. Wong, K. S. Yong and W. K. Tiong, "A New Hybrid Ensemble Feature Selection Framework for Machine Learning-based Phishing Detection System," Information Sciences, vol. 484, pp. 153-166, 2019.

[22] P. PhishTank, "Join the Fight against Phishing," [Online], Available: http://phishtank.org, 2016.

[23] R. K. V. Penmatsa and P. Kakarlapudi, "Web Phishing Detection: Feature Selection Using Rough Sets and Ant Colony Optimization," International Journal of Intelligent Systems Design and Computing, vol. 2, pp. 102-113, 2018.

[24] O. S. Qasim and Z. Y. Algamal, "Feature Selection Using Particle Swarm Optimization-based Logistic Regression Model," Chemometrics and Intelligent Laboratory Systems, vol. 182, pp. 41-46, 2018.

[25] N. A. Azeez and A. Oluwatosin, "CyberProtector: Identifying Compromised URLs in Electronic Mails with Bayesian Classification," Proc. of the IEEE International Conference on Computational Science and Computational Intelligence (CSCI), pp. 959-965, Las Vegas, USA, 2016.

[26] S. Zaman, S. M. U. Deep, Z. Kawsar, M. Ashaduzzaman and A. I. Pritom, "Phishing Website Detection Using Effective Classifiers and Feature Selection Techniques," Proc. of International Conf. on Innovation in Engineering and Technology (ICIET), vol. 23, p. 24, DOI: 10.13140/RG.2.2.24043.08483, 2019.

[27] A. Odeh, I. Keshta and E. Abdelfattah. "Efficient Detection of Phishing Websites Using Multilayer Perceptron," International J. of Interactive Mobile Technologies (iJIM), vol. 14, no. 11, pp. 22- 31, 2020.

[28] Y. Freund, R. Schapire and N. Abe, "A Short Introduction to Boosting," Journal-Japanese Society for Artificial Intelligence, vol. 14, no. 5, pp. 771-780, 1999.

[29] D. C. Feng, Z. T. Liu, X. D. Wang, Y. Chen, J. Q. Chang, D. F. Wei and Z. M. Jiang, "Machine Learning-based Compressive Strength Prediction for Concrete: An Adaptive Boosting Approach," Construction and Building Materials, vol. 230, ID no. 117000, [Online], Available: https://doi.org/10.1016/j.conbuildmat.2019.117000, 2020.

[30] S. Abdulhamit and E. Kremic. "Comparison of Adaboost with MultiBoosting for Phishing Website Detection," Procedia-Computer Science, vol. 168, pp. 272-278, 2020.

[31] V. Shahrivari, M. M. Darabi and M. Izadi, "Phishing Detection Using Machine Learning Techniques," arXiv preprint arXiv:2009.11116, [Online], Available: https://arxiv.org/pdf/2009.11116.pdf, Sep. 2020.

[32] V. Ramanathan and H. Wechsler, "Phishing Website Detection Using Latent Dirichlet Allocation and AdaBoost," Proc. of the IEEE International Conference on Intelligence and Security Informatics, pp. 102-107, Arlington, USA, 2012.

[33] B. Alotaibi and M. Alotaibi, "Consensus and Majority Vote Feature Selection Methods and A Detection Technique for Web Phishing," Journal of Ambient Intelligence and Humanized Computing, [Online], Available: https://doi.org/10.1007/s12652-020-02054-3, 2020.

[34] M. Zabihimayvan and D. Doran, "Fuzzy Rough Set Feature Selection to Enhance Phishing Attack Detection," Proc. of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp. 1-6, DOI: 10.1109/FUZZ-IEEE.2019.8858884, June 2019.

"PhiBoost- A Novel Phishing Detection Model Using Adaptive Boosting Approach", A. Odeh, I. Keshta and E. Abdelfattah.

[35] Y. A. Alsariera, A. V. Elijah and A. O. Balogun, "Phishing Website Detection: Forest by Penalizing Attributes Algorithm and Its Enhanced Variations," Arabian Journal for Science and Engineering, vol. 45, pp. 10459–10470, 2020.

ملخص البحث:

تزداد كل يومٍ الهجمات السيبرانية التي تستخدم استراتيجياتٍ مختلفة. ومن أكثر الهجمات السيبرانية شيوعاً ما يعرف بـ "التلصُّص" على البيانات؛ إذ يقوم المهاجم بجمع معلوماتٍ حساسةٍ وسريةٍ بينما يحاول إظهار نفسه كطرفٍ موثوقٍ به. وقد ابتكرت استراتيجياتٌ مختلفة لمقاومة هذه الظاهرة، مثل: الوضع على القائمة السوداء، والبحث الموجه لكشف الهجمات، والتشابه المرئي. ولكن هذه الطرق التقليدية لها في الغالب معدلات خطأ عالية وتستغرق الكثير من الوقت لكشف الموقع الإلكتروني المهاجم. وقد استخدمت نماذج جديدة تستفيد من تقنيات تعلم الآلة التي من شأنها أن تحسن من دقة الكشف.

وتحتاج تقنيات تعلم الآلة الى كمياتٍ ضخمةٍ من البيانات تسمى "السِّمات"، ويتم جمعها من مواقع الكترونية مختلفة. وتصنف هذه السِّمات التي يجري جمعها ضمن أربعة أصناف او فئات.

تقدم هذه الورقة نموذج كشفٍ مبتكراً يستند على الاستفادة من انتقاء السِّمات لالتقاط السِّمات ذات الارتباط العالي بعلامة الصنف. وتوظف مرحلة انتقاء السِّمات مكتبة السِّمات ذات الأهمية المستقلة من ماتلاب (MATLAB) والخريطة الحرارية من بايثون (Python) لإيجاد السِّمات ذات الارتباط العالي. بعدئذٍ، يستخدم النموذج المقترح طريقة تعزيزٍ تكيّفيةً تشتمل على عدّة مُصنِّفات لزيادة دقة النموذج. ويحقق النموذج المقترح دقةً تنبُّؤيّةً عاليةً جداً تصل الى ما يقرب من 99%.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 07, No. 01, March 2021.

A NOVEL INSTANCE SEGMENTATION ALGORITHM BASED ON IMPROVED DEEP LEARNING ALGORITHM FOR MULTI-OBJECT IMAGES

Suhaila Farhan Ahmad Abuowaida[1], Huah Yong Chan[1], Nawaf Farhan Funkur Alshdaifat[1] and Laith Abualigah[2]

(Received: 26-Oct.-2020, Revised: 15-Dec.-2020 and 17-Jan.-2021, Accepted: 2-Feb.-2021)

ABSTRACT

A Deep Learning (DL) algorithm is highly common and attractive in recent years because of its encouraging achievements in many areas. DL lies in image-based detection and instance segmentation of an entity, which is a critical issue that needs further investigation. This paper aims to study the fundamental challenges in using object instance segmentation of images. This paper proposes a novel algorithm for multi-object image instance segmentation algorithm in three stages. A novel backbone approach improves the image recognition algorithm by extracting low and high characteristic levels from the given images in the first stage. The ResNet is the fundamental building block and connects with the Squeeze-and-Excitation Network (SENet) for each ResNet block. The Region Proposal Network (RPN) is used to determine the object item's placement, followed by the third stage, which suggests an average position RoI layer to choose the optimal boundaries of the instance segmentation. The experiments are conducted and validated using a standard benchmark image dataset, called COCO. The proposed algorithm's performance is validated using standard evaluation criteria and compared against the recent image segmentation algorithms that use object instances. The results show that the proposed algorithm gets better results than other well-known instance segmentation algorithms in terms of average accuracy over IoU (AP) threshold measures using various thresholds.

KEYWORDS

Deep learning, Multi-object detection, Recognition, Instance segmentation, Average position RoI layer.

1. INTRODUCTION

DL as a branch of machine learning is given this term, since it uses a Deep Neural Network (DNN) [1]. DNN has attracted significant interest and attention over the years due to its ability to handle complex data by its very nature and having a high-level of dimensions [2], such as computer vision [3]-[4], speech recognition [5]-[6], Rate Control (RC) [7], depth estimation from singe image [8] and neural language processing [9]-[10]. One of the essential characteristics of the DNN is dealing with a broad set of data in its various forms in the training phase through optimization algorithms. Within a short period of time, the vision community has been improved rapidly for object recognition [11]-[12], object detection [13]-[14], semantic segmentation [15] and instance segmentation [16]-[17] based on DL. The object detection algorithms, which are used to get object information, have two main problems. First, the traditional algorithms cannot solve the object detection [18]-[19] and recognition problems [20]-[21] effectively. This problem mainly focuses on distinguishing the object from the background and addressing labels of the object class. Second, addressing the bounding boxes of each object is a critical issue to solve the object localization. The development process has been motivated by a powerful baseline algorithm. A Region-Convolution Neural Network (R-CNN) [22] is used to solve the problem of multiple-object detection by generating a particular region search, which can draw bounding boxes over all of the objects. Afterward, the algorithm applies the VGG backbone with a modified Fully Connected (FC) layer using Support Vector Machine (SVM), which extracts a feature map for each region and image detection. R-CNN's main drawbacks are that its detection process is slow, requires multiple stages and is computationally costly. Nevertheless, the Fast R-CNN enhances the R-CNN to solve the low accuracy and slow detection problems by sharing computation of the convolution layers of various proposals and running the CNN [23]. This is the primary motivation to generate proposals for the region using selective

1. S. F. A. Abuowaida, H. Y. Chan and N. F. F. Alshdaifat are with School of Computer Sciences, Universiti Sains Malaysia, 11800, Pulau Pinang, Malaysia. Emails: suhilaowida@student@usm.my, hychan@usm.my and nawaf@student@usm.my

2. L. Abualigah is with Faculty of Computer Science and Informatics, Amman Arab University, Amman 11953, Jordan. Email: Aligah.2020@gmail.com

search. Henceforth, the proposal region is sent to Region of Interest (RoI), which is a proposed region from the input image and RoI pooling that converts the feature inside each region to make small feature maps using max-pooling. The main issue of the algorithm is its slow detection, since using selective search produces bottlenecks. Notably, Faster R-CNN [24] is the enhanced version of Fast R-CNN by combining the RPN and Fast R-CNN. Furthermore, transforming the Faster R-CNN image into a convolution network requires output, which is a set of feature maps on the last convolution layer used. Hence, a sliding window is implemented for each feature map. The Faster R-CNN uses 3×3 as the sliding window size and a set of nine anchors, which computes how much these anchors have overlapped with the ground-truth bounding boxes. Finally, each feature map extracted from the convolution layer has fed a smaller network with two tasks: classification and regression.

The instance segmentation is an essential task of the object recognition system in recognising each object based on the image. The instance segmentation task is challenging due to several challenges, including diversity, difference between the colours, sizes of the object items and overlapping between the objects. Li et al. in [25] have proposed a new algorithm for image segmentation, called Full Convolution Instance Segmentation (FCIS). However, in perdition edges, FCIS suffers from overlapping instances and errors. Another research [26] has proposed a Multi-task Network Cascade (MNC) algorithm for instance segmentation. The MNC comprises of three stages; each stage has a particular task to predict the instance level for each object. The first stage proposes the bounding box for each object in the image, the second stage presents a mask for each bounding box and the third stage distinguishes between instances. However, the MNC, which has numerous predicting instance segmentation gaps, is inflexible and takes much time when predicting instance segmentation. Also, the main problem in MNC is that the three stages do not work in a parallel way and require many parameters for each stage, leading to prolonged time to predict instance segmentation. Mask R-CNN is used to predict instance-level segmentation [27]. The proposed algorithm utilizes the Faster R-CNN to predict each object's mask by adding a branch for each bounding box after the Faster R-CNN. The Mask R-CNN works in parallel to decrease training and testing time. Moreover, the main contribution of Mask R-CNN is that it uses RoI align and provides highly accurate results. However, it has taken a long time in the training stage and lost some features at the instance level. A real-time algorithm is proposed in [28], called You Only Look at CoefficienTs (YOLACT), where it uses the parallel concept in its main procedure. However, this algorithm does not receive satisfying feedback because of instance segmentation's accuracy values. Regarding the architecture of YOLACT [28], the researcher in [29] has constructed a Cascade R-CNN that minimizes the overfitting using a sequential threshold. However, the Cascade R-CNN increases the threshold training and testing time. The architecture of Cascade R-CNN consists of many stages, where each stage extends Faster R-CNN for the localization of the object for each input image and extends Mask R-CNN to instance segmentation. The main advantage of Cascade R-CNN is decreasing overfitting through a sequence of detectors trained with increasing IoU thresholds and is sequentially more selective against close false positives.

The main goal of this paper is to build a new algorithm able to perform the instance segmentation process effectively. Therefore, it combines elements from the classical computer vision object detection tasks. The objective is to classify individual objects, locate each using a bounding box and instance segmentation. The objective is to classify each pixel into a fixed set of categories without differentiating object instances. Normally, a complex algorithm is required to achieve good results. We demonstrate a surprisingly simple, versatile and fast algorithm that can overcome the results of well-known instance segmentation algorithms. This paper has proposed a new algorithm for instance segmentation of objects to solve the problems mentioned above. Hence, a novel image instance segmentation algorithm is proposed; namely, multi-object instance segmentation is divided into three phases. In the first phase, a novel backbone architecture is proposed, aiming to extract the object's feature maps with high precision value and less time. In the second phase, the RPN is adapted to identify multiple objects. In the third phase, the Fully Convolution Network (FCN) [30] is used to generate instance segmentation to prevent the overlapping problem between objects. This prevention manages the various sizes of RPN's feature maps using the average position RoI layer. So, the instance segmentation of multiple-object instances is the main aim of this research. The proposed multi-object instance segmentation algorithm is evaluated using two measures; the AP with different thresholds and time. The obtained results of the proposed algorithm are compared against other well-known instance segmentation algorithms published in the literature, such as MNC, FCIS, Mask R-CNN, YOLACT and Cascade R-CNN. The experimental results

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 07, No. 01, March 2021.

have shown that the proposed image instance segmentation algorithm has obtained better results compared to other well-known image instance segmentation algorithms. The rest of this paper is organized as follows. Section 2 presents the full details of the proposed image instance segmentation algorithm and its main procedures. Section 3 shows the experiments and discussion. Finally, the conclusions and future work directions are given in Section 4.

2. THE PROPOSED ALGORITHM

This section presents the proposed instance segmentation algorithm based on the improved DL algorithm for multi-object detection. We have focused on the multi-object instance segmentation by identifying multi-object tasks.

Figure 1 shows the proposed algorithm which consists of several improved algorithms, including the backbone, detection and instance segmentation algorithms in order to obtain the most significant results by accurately presenting the multi-object image. The mathematical presentation of the given problem is explained as follows.

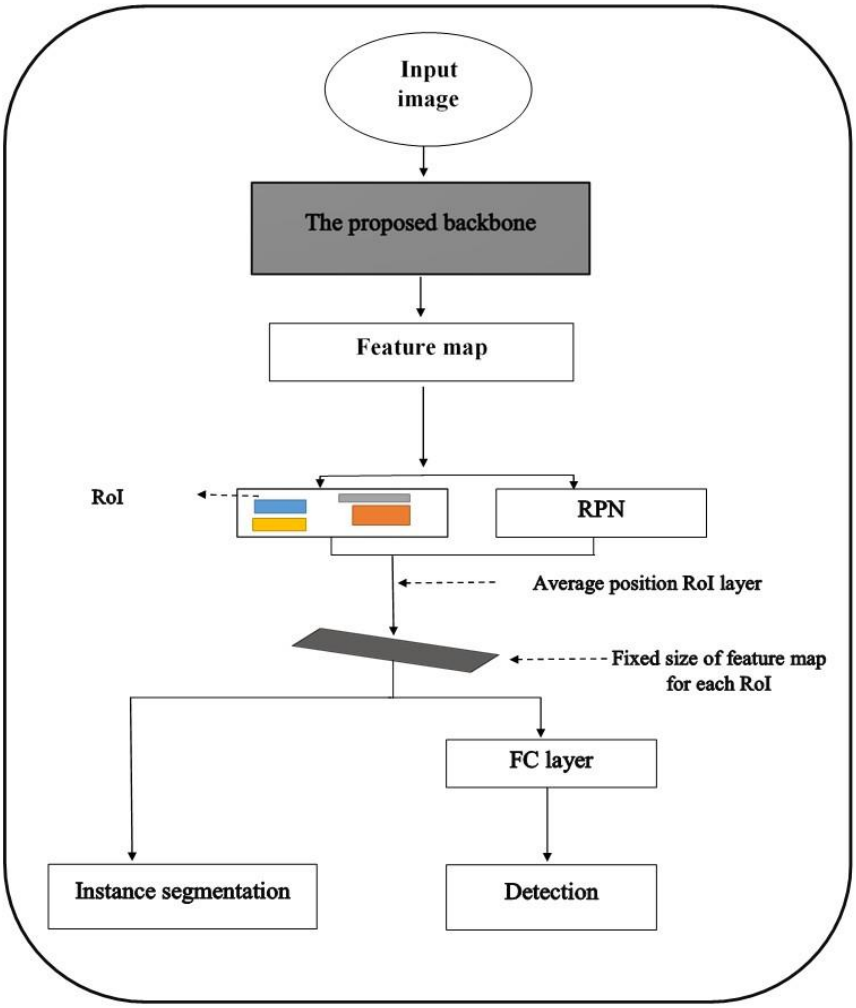


Figure 1. The proposed algorithm.

For each object, the overall loss function of the multi-object instance segmentation is calculated using Equation (1).

$$L = L_{detection} + L_{instance\ segmentation} \tag{1}$$

where $L_{detection}$ loss function is given according to Equation (2).

$$L(p_i, t_i) = \frac{1}{N}\sum_{i=1}|p_i^* - p_i|^2 + \lambda\frac{1}{N}\sum_{i} p^* L_{loc}(t_i, t_i^*) \tag{2}$$

where *pi* is the predicted probability of anchor *i*, p_i^* is the ground truth of anchor *i*, t_i stands for the coordinates predicted, t_i^* is the coordinates ground truth, *N* is the normalization term and λ = *a* is the balancing parameter.

The $L_{instance\ segmentation}$ loss function is identified using the per-pixel sigmoid and average binary cross-entropy to generate boundaries for each class, as shown in Equation (3).

$$L_{instance\ segmentation} = -\frac{1}{s^2}\sum_{1\le i,j\le s}[y_{ij}logy_{ij}^k + (1-y_{ij}\log(1-y_{ij}^k)] \quad (3)$$

where y_{ij} is the ground truth of boundaries of size region (s^2), y_{ij}^k is the predicted value of boundaries and *k* is the ground truth class.

The pseudocode of the proposed algorithm is as shown in Algorithm 1.

Algorithm 1: Pseudocode of the proposed algorithm

```
block 1, block 2, block 3, block 4, block 5 = build proposed backbone()
anchors = generate-anchors()
rpn = build-rpn()
rois = Proposal-Layer(rpn, anchors)
if mode == 'training': then
    ground-truth-values = values from the training dataset
    bbox, classes = classifier(rois)
    target-detection = Detection-Target-Layer(ground-truth-values)
    instance-segmention = instance-segmention (rois from target
    detection)
    loss = loss-functions(target-detection, bbox, classes, instance-segmention )
    algorithm = [bbox, classes, instance-segmention, loss]
else
    bbox, classes = classifier(rois)
    target-detection = Detection-Layer(bbox,
    classes) instance-segmention = instance-
    segmention (rois) algorithm = [bbox, classes,
    instance-segmention]
end
return algorithm
```

2.1 The Proposed Backbone

The ResNet has attracted significant interest and attention due to its ability to handle complex data and high accuracy compared to other backbones [31]; it consists of a series of blocks to overcome the problem of the vanishing of gradient [31]. Therefore, there are problems with ResNet backbone, such as: 1) determination of ResNet block that has failed to receive sufficient training, 2) determination of ResNet block that has received more than sufficient training and 3) adopting a large filter size in the first convolution layer. In this paper, ResNet is improved in the proposed backbone as the fundamental building block based on the proposed Equation 4. The ResNet block is fed forward directly and linked to all other layers, which consist of a series of blocks to overcome the vanishing of the gradient.

$$H(y) = \sum_{I=1}^{n} F(y,\{y_i, I\}) + y \quad (4)$$

where *y* is the building block's input, *H(y)* is the block's output, *F(y, W)* is the remaining mapping that you acquired during the training stage and *I* represents the number of iterations to every ResNet block. In the case of a layer of insufficient training, *I* should be raised, while additional training must be decreased. The chosen filter size in the first convolution layer has been smaller than ResNet due to the feature map's extraction. In the proposed backbone, the first convolution layer uses a 5x5 filter size accompanied by max-pooling of the 2x2 matrix to obtain more features, as shown in Figure 2. In order

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 07, No. 01, March 2021.

to further enhance information flow across layers, the performance of the convolution layer is used for the input to the ResNet block of the proposed ResNet backbone.

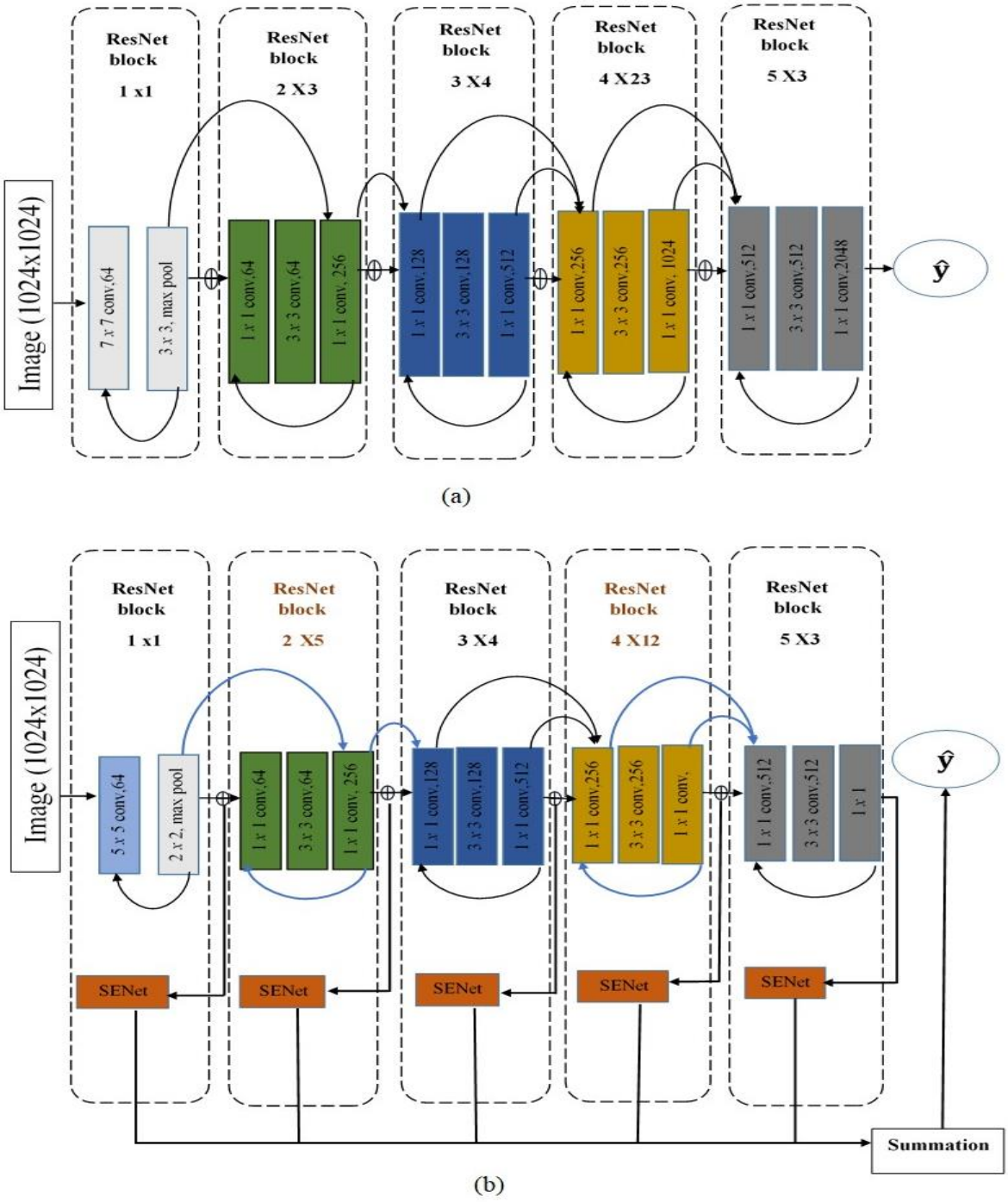


Figure 2. (a) The existing ResNet-101 backbone [31] (b) The improvement ResNet backbone.

For incorporating a content-aware mechanism to weigh each channel adaptively, the production of each proposed ResNet block is transmitted *via* the SENet network to provide a linear scalar of how relevant each proposed ResNet block is, which can be written as:

$$F_{feature\ map} = \sum_{i=1}^{W} \sum_{j=1}^{H} y_q(i,j) \tag{5}$$

where y_q is the element of the feature map with spatial dimension $H \times W$, where H is the height and W is the width.

Sequentially, we have obtained five samples of the building blocks consisting of ResNet blocks and a network SENet. Outputs are integrated from each of the SENet networks to combine all features from various depth levels by summation of the characteristics of feature maps derived from the five couples of the ResNet blocks, as shown in Figure 2. In the following sub-section, the architecture of SENet will be addressed.

2.1.1 The architecture of SENet

The feature-generating maps from the ResNet improvement have been fed to the SENet network [32] to obtain further channel information and enhance the sharing of information, as shown in Figure 3. It selectively uses global information to illustrate and eliminate less valuable features by using weights on each feature map's layer. It contains five operations, including a global average pooling, an FC layer, an ReLU function, an FC layer and the sigmoid function. The role of the sigmoid activation for channel weights is suited to the input. The SENet architecture is illustrated in Figure 3. As represented in Figure 3, the SENet architecture mainly consists of two processes, which are:

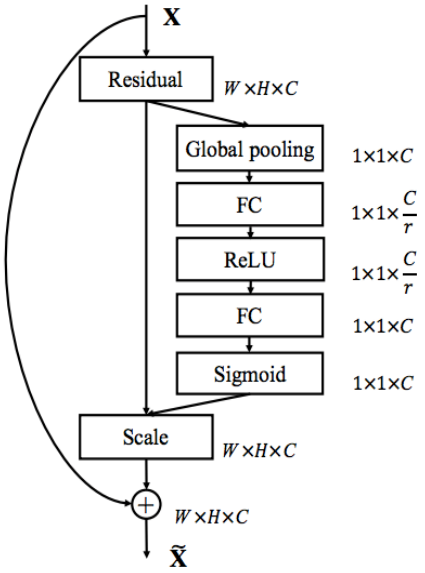


Figure 3. The SENet architecture [32].

- The squeezing process: It produces channel-wise statistics (Se $\in R^D$) through global average pooling, which can be written as:

$$Se = F_{SENet}(y_q) = \frac{1}{H \times W} \sum_{i=1}^{W} \sum_{j=1}^{H} y_q(i,j) \tag{6}$$

where $F_{SENet}(.)$ is the function of squeezing. y_q is the element of the feature map with spatial dimensions $H \times W$, where y_q is the q^{th} element of Se and $q = 1, 2, ..., D$.

- Excitation process: It provides and identifies channel-wise dependencies and significantly minimizes the number of parameters through FC layers, sigmoid and ReLU functions, as shown in the following formula.

$$T = F_{excitation}(Se, W) = \sigma\big(G(Se, W)\big) = \sigma\big(W_2 \delta(W_1 Se)\big) \tag{7}$$

where T = t1, t2, ..., tD , Fexcitation is the function of excitation and $t_q \in R^{H \times W}$. $\delta(x) = MAX(x, 0)$ is reference to ReLU function, G(.,.) is the reference to global function and $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigma mode function.

$$\hat{P} = F_{scale}(Se_q, y_q) = Se_q . y_q, \tag{8}$$

where $y_q \in R^{H \times W}$ and F_{scale} is the reference to channel-wise multiplication between the scalar Se_q and the feature map y_q.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 07, No. 01, March 2021.

2.2 Localization

The RPN is implemented in the multi-object form to decide the location of multiple objects in the input image [24]. Besides, the RPN approved any sizes of the feature map that would act as the output. In the meantime, the proposed CNN functions as the input to produce multiple proposals for rectangular objects. The object is illustrated in the current technique for rectangular objects, while the sliding window is seen in all the feature maps collected by the proposed CNN's last convolution layer. The sliding window in RPN comprises nine anchors, which are the center points of the sliding window. In particular, the location for each anchor is calculated based on the input image, which gives the sliding window different Aspect Ratio (AR) and Scale (S) values, as seen in Figure 4.

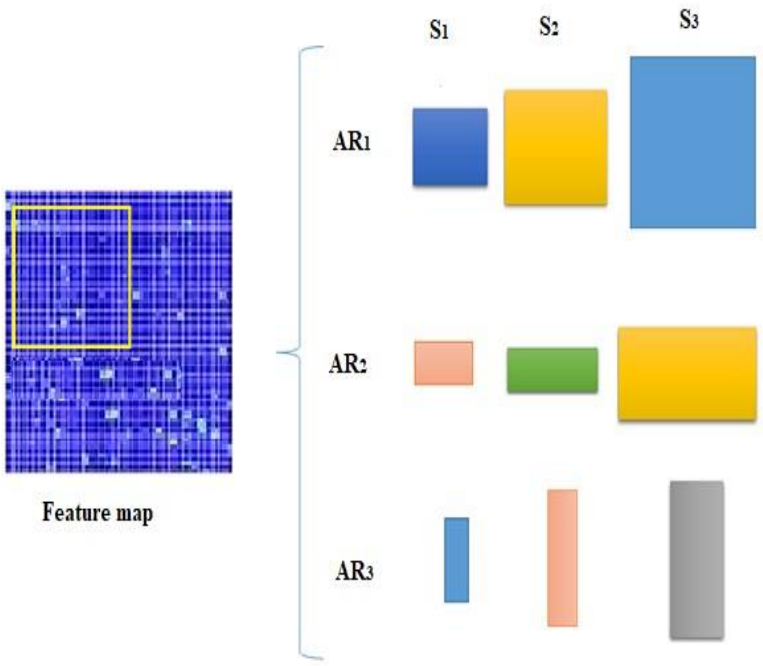


Figure 4. The sliding window with different aspects in ratio and scale.

As a consequence, the value of p^* to every anchor is determined on the two parameters that are as follows:

1) The anchors with the largest intersection-over-union overlap and a ground truth box.
2) For each anchor, the overlap intersection-over-union (IoU) is higher than 0.7.

The IoU is defined by the following formula:

$$IoU = \frac{Anchor \cap ground\ truth\ box}{Anchor \cup ground\ truth\ box} \tag{9}$$

2.3 Average Position Region of Interest Pooling Layer (Average Position RoI Layer)

Several propositions of rectangular objects are created on feature maps from RPN, as represented in Figure 1. Consequently, various map size features are designed, which affected the instance segmentation accuracy. This paper, therefore, has suggested a novel layer for handling the feature map's different sizes. The function map has been reduced over the following two steps to a fixed scale, known as the average position RoI layer. Suppose that the feature map's size is 5x5, where the rectangular object proposals are encoded in red colour as represented in Figure 5.

The first move is to preserve the position of feature maps by stopping implemented quantification to each RoI boundary *via* the RoI Pool [25], as represented in Figures 6 and 7. Nevertheless, because of the strong quantization levels for every pixel and success in order to achieve optimal performance in segmentation, low performance is found in the RoI Pool segmentation [25]. This paper has solved the

question of misalignment by annulling quantization. For each bin, the second step has utilized average pooling to reduce computational complexity and extract low-level features from the neighbourhood, as represented in Figure 8.

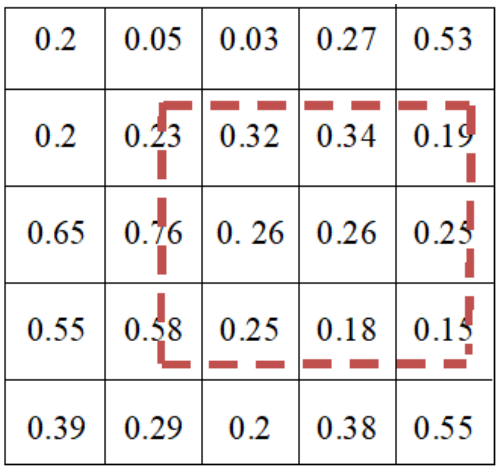


Figure 5. The rectangular object proposals in red colour on the feature map.

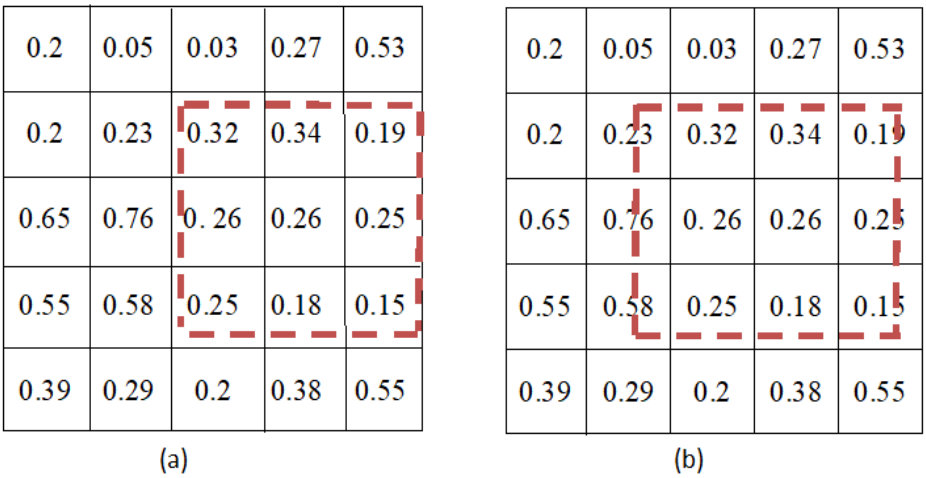


Figure 6. (a) The RoI Pool after implementing of quantization (b) The prevention of quantization by average position RoI.

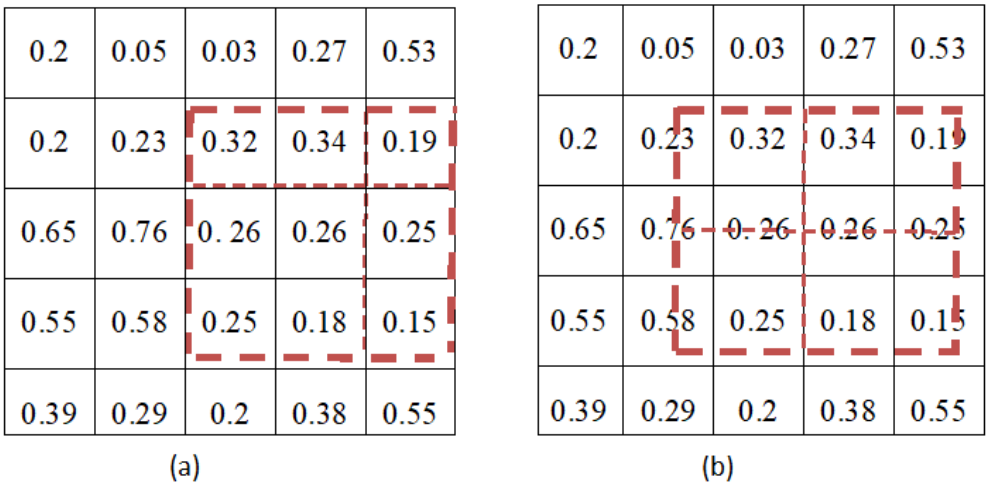


Figure 7. (a) The RoI Pool after the second implementation of quantization (b) The second prevention of quantization by average position RoI.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 07, No. 01, March 2021.

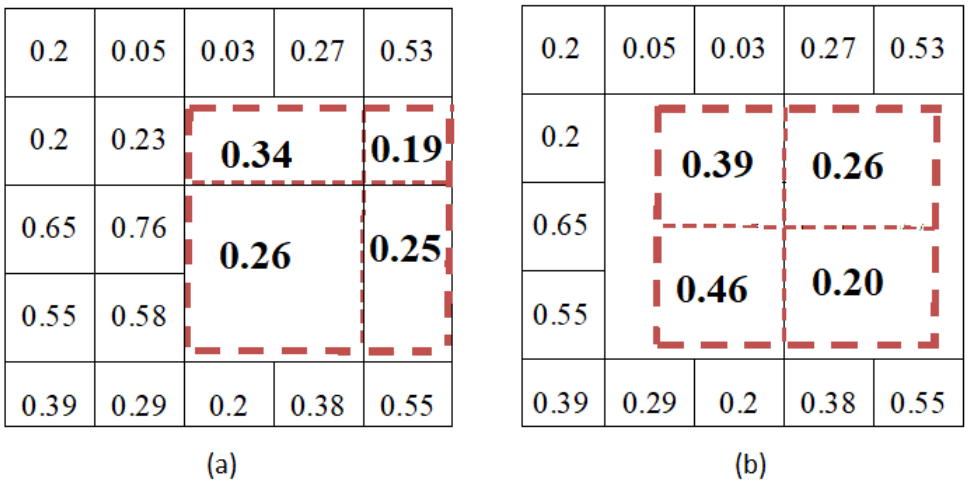


Figure 8. (a) The RoI Pool result (b) The average position RoI result.

After that, the result of the average position RoI layer is passed to the FC layer. This process consists of detection for each object element, as shown in Figure 1.

2.4 Instance Segmentation

The FCN [30] is done to differentiate between class levels, while the segmentation of the instance has returned the boundaries of each object element. This process consists of three stages, which are as follows:

1) The first stage consists of translating the average position RoI layer performance for the object item into a sequence of 3 x 3 convolution stages, multiple times after applying ReLU for the creation of limits for each area from the average position RoI layer.
2) The second stage requires a 1 x 1 convolution layer to every feature map acquired from the last convolution.
3) The third stage converts the segmentation dimension based on the input image by bi-linear interpolation.

3. Results and Discussion

The experimental findings and the proposed algorithm evaluation from the preceding sections are summarized in this section. The multi-object instance segmentation algorithm assessment is carried out using the following measurements: AP with specific thresholds is used to evaluate the multi-object instance segmentation algorithms, including improved CNN, RPN and instance segmentation. To ensure a judgment on the enhanced segmentation of multi-object instances, the outcome is compared with those of other well-known algorithms that have been used for the multi-object instance.

3.1 Benchmark Datasets

The experiments are conducted on MS-COCO dataset [33], which includes 1118k images for training, 5k for validation (Val) and 20k for annotated testing (test-dev). The calculation of COCO AP was done from 0.5 to 0.95, with an interval of 0.05. All models have been trained on the COCO training set and tested on the Val set. For a fair comparison, the final results are compared with the state-of-the-art instance segmentation algorithm on the test-dev package.

3.2 Experimental Specifications

A new algorithm for the instance segmentation of multi-object instances is introduced using TensorFlow [34]. The algorithms are tested with the GPU-Us-Tesla V100 16 GB and the VCPUs-8 cores 61 GB on Amazon Web Services (AWS) and Amazon Machine Image (AMI). In the training stage, the RoI is determined positive if it has IoU with a ground-truth box of at least 0.5 and negative otherwise and the L segmentation loss function is defined on positive RoIs. The instance segmentation is the intersection between an RoI and its associated ground-truth of instances segmentation. We train on GPU-Us-Tesla

V100 16 GB and the VCPUs-8 cores 61 GB on Amazon Web Services (AWS) and Amazon Machine Image (AMI). The weight decay for 50 epochs was 0.0001 with a learning momentum of 0.9 and a learning rate of 0.001. Every one of the epochs is an iteration of 1000. Besides, the optimization algorithm used in the context of this analysis is Stochastic Gradient Descent (SGD) [27].

3.3 Comparison with the State-of-the-Art Layers

In this sub-section, a comparison of the proposed algorithm with the state-of-the-art layers is conducted to validate the average position RoI layer ability. Table 1 compares the performance of the average position RoI layer with state-of-the-art layers.

Table 1. Evaluation results of the proposed backbone with various RoI layers.

RoI layers	Instance segmentation			Detection		
	AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}
RoI pooling [25]	31.0	53.6	30.12	35.6	60.1	34.3
RoI wrap [26]	28.3	50.2	28.1	31.8	53.7	31.6
Average position RoI layer	**45.8**	**66.7**	**47.1**	**47.6**	**68.6**	**47.1**

Based on the results, the average position RoI layer ability has obtained high accuracy with different AP values. The feature maps are reduced to a fixed size while maintaining the map's location obtained from the previous algorithms by avoiding quantization, while RoI Wrap and RoI pooling are still considered quantization in the RoI boundary, which leads to losing alignment with the input image. As a result, significant results in the detection and instance segmentation are obtained due to the impact on the detection and instance segmentation process for each pixel value in the feature maps.

3.4 Comparison with the State-of-the-Art Detection Algorithms

We compare our proposed algorithm with the state-of-the-art algorithms on COCO detection to validate the proposed algorithm's ability.

Table 2. Multi-object detection algorithm performance with different thresholds (0.5, 0.75, Small (S), Medium (M), Large (L)).

Algorithms	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Fast R-CNN	VGG	19.7	35.9	-	-	-	-
Faster R-CNN	VGG	21.9	42.7	-	-	-	-
Faster R-CNN	ResNet-101	34.9	55.7	37.4	15.6	38.7	50.9
Mask R-CNN	ResNet-101	38.2	60.3	41.7	20.1	41.1	50.2
Cascade R-CNN	ResNet-101	42.8	62.1	46.3	23.7	45.5	55.2
Proposed algorithm	**ResNet-50**	**40.5**	**62.7**	**43.3**	**23.5**	**42.8**	**51.5**
Proposed algorithm	**ResNet-101**	**44.8**	**66.0**	**46.6**	**30.2**	**48.1**	**56.8**
Proposed algorithm	**Proposed backbone without SENet**	**45.5**	**66.4**	**46.8**	**30.4**	**48.9**	**57**
Proposed algorithm	**Proposed backbone with SENet**	**47.6**	**68.6**	**47.1**	**32.8**	**50.8**	**58.9**

The efficiency of the proposed algorithm indicates better results with different AP values. The proposed algorithm AP accuracy has amounted to 47.6, 68.6, 47.1, 32.8, 50.8 and 58.9. Particularly, these values are considerably higher than comparable algorithm values due to the recommendation of a new backbone in this paper, which addresses insufficient training and determines the best possible filter size.

Additionally, inserting an SENet network for each block helps increase the efficiency of the method's productivity by capturing several local and precise features from the input image.

3.5 Comparison with the State-of-the-Art Instance Segmentation Algorithms

In this sub-section, a comparison of the proposed algorithm with the state-of-the-art algorithms is conducted to validate the proposed algorithm's ability.

It is clear from Table 3 that the proposed algorithm is compared with the state-of-the-art algorithms, including MNS [26], FCIS [25], Mask R-CNN [27], YOLACT [28] and CASCADE R-CNN [29].

Table 3. Multi-object instance segmentation algorithm performance with different thresholds (0.5, 0.75, Small (S), Medium (M), Large (L)).

Algorithms	**Backbone**	**AP**	**AP**$_{50}$	**AP**$_{75}$	**AP**$_S$	**AP**$_M$	**AP**$_L$
MNC	ResNet-101	24.6	44.3	24.8	4.7	25.9	43.6
FCIS	ResNet-101	29.2	49.5	-	7.1	31.3	50.0
Mask R-CNN	ResNet-101	35.7	58.0	37.8	15.5	38.1	52.4
YOLACT	ResNet-101	31.2	50.6	32.8	12.1	33.3	47.1
CASCADE R-CNN	ResNet-101	42.8	62.1	46.3	23.7	45.5	55.2
Proposed algorithm	ResNet-50	33.8	55.1	36.4	19.4	42.2	51.1
Proposed algorithm	ResNet-101	36.3	57.4	38.7	18.8	40.5	53.8
Proposed algorithm	**Proposed backbone without SENet**	**43.9**	**64.8**	**45.3**	**22.4**	**44.4**	**54**
Proposed algorithm	**Proposed backbone with SENet**	**45.8**	**66.7**	**47.1**	**24.3**	**46.2**	**55.9**

The performance of the proposed algorithm has been found to exhibit better results with different AP values. Our algorithm's AP accuracy has amounted to 45.8, 66.7, 47.1, 24.3, 46.2 and 55.9. Notably, these values are significantly higher than the comparative algorithms' values due to the suggestion of a new backbone in this paper, which is essential to achieve better results.

The proposed algorithm has solved gradient vanishing *via* an identity shortcut based on a new gradient equation while taking into account the efficiency training for each convolution block.

This function is considered to remove a certain amount of feature from the input image and transfer it to other layers by means of a similar duplicate increase or decrease in training and reduction in the filter size. Also, the SENet network has increased the performance of the feature selection process. The proposed backbone incorporates deeper and shallow feature maps. Because there are several local and accurate feature maps on the shallow layers, there are also rich feature maps on the deep network layers. Consequently, with the proposed backbone, we should effectively collect feature maps in different improved ResNet blocks and transfer them to the SENet for additional spatial feature maps.

Compared to other algorithms, the results obtained from the produced function are positive results. In addition to the structure, this paper has also proposed a novel layer for managing different sizes of the feature map created from the RPN, known as the average position RoI. Furthermore, the average position RoI layer has reduced the feature maps to a fixed size while preserving the map's position obtained from previous algorithms by avoiding quantization.

As a result, significant results in the instance segmentation are obtained due to the impact on the instance segmentation process of every pixel value in the feature maps. Given the significant success of the proposed algorithm in the multi-object segmentation setting, it is expected that the proposed algorithm obtains the potential for substantial results. The visual experimental results are presented in Figure 9.

3.6 Time Measure

In this sub-section, the time measure is used to evaluate the proposed algorithm's training time and frame per second in the testing process. The training and frame per second are significant factors in evaluating algorithm efficiency. The comparative algorithms are evaluated in terms of the training time in second per image and frame per second, as shown in Table 4.

Figure 9. The visual experimental results from the proposed algorithm for instance segmentation.

Table 4. Evaluation of training time in second per image and frame per second of multi-object segmentation using different state-of-the-art algorithms.

Algorithms	Training time in second	Frame per second
MNC	0.21	5.45
FCIS	0.14	5.75
Mask R-CNN	0.11	6.07
YOLACT	0.27	29.5
CASCADE R-CNN	0.41	8.03
Proposed algorithm	0.10	8.71

Based on the evaluation frame per second in the testing process of multi-object instance segmentation algorithm with different state-of-the-art algorithms, the proposed algorithm has produced 8.71 frames per second, making it the second algorithm following YOLACT algorithm despite of the YOLACT algorithm's high-speed characterization in this process. At the same time, the precision is omitted in

AP50, AP75, AP90, APS, APM and APL due to its reliance on the proposed backbone, which has resulted in the avoidance of the issue in ResNet. This trend has happened with the reduction of the filter size and the emphasis on the layer, allowing further training and reduction in the amount of excessive layer testing to obtain good results in the shortest possible time. In contrast, the MNS, FCIS, Mask R-CNN and CASCADE R-CNN algorithms have produced less frames per second due to ResNet101 implementation as the backbone network. ResNet has faced many problems, including using a large filter size affecting the increased consumption time parameters, as stated earlier. In addition, several layers have not undergone preparation, which has resulted in a substantial time investment in the training cycle, due to the three-fold repetition of the bounding box by the algorithm.

4. CONCLUSION

Multi-object image is improved by extracting features low and large *via* creating a novel backbone by several connected copies of the ResNet blocks of enhancement ResNet network connected with SENet to obtain additional channel features, improve the use of a more significant feature in images and provide a linear scalar of how relevant each proposed ResNet block is. The second phase is to adopt RPN to locate the object item and create a new layer called the average position RoI layer, which manages to map various features to obtain the best boundaries for the multi-object image. In the third phase, the FCN is used to generate instance segmentation to prevent the overlapping problem between objects. This prevention manages the various sizes of RPN's feature map using the average position RoI layer. So, the segmentation of multiple-object instances is the main aim of this research. The proposed multi-object instance segmentation algorithm is evaluated using two measures; AP with different thresholds and training time and frames per second. The proposed algorithm's obtained results are compared against other well-known segmentation algorithms published in the literature, such as MNC, FCIS, Mask R-CNN, YOLACT and Cascade R-CNN. Better performance regarding the accuracy AP with different thresholds; namely, AP50, AP75, AP90, APS, APM and APL, is observed from the proposed algorithm. Such thresholds are higher than those of the state-of-the-art instance segmentation algorithms. Notably, the proposed system has rapidly and reliably defined, located and segmented the multi-object precision, preparation, training and frames per second. The proposed algorithm can be enhanced in future work by adding an edge detection algorithm to capture multiple objects' fine details. Also, we will implement variations of CNN architectures to investigate instance segmentation.

REFERENCES

[1] Q. Zhang, L. T. Yang, Z. Chen and P. Li, "A Survey on Deep Learning for Big Data," Information Fusion, vol. 42, pp. 146–157, 2018.

[2] L. Liu, W. Ouyang et al., "Deep Learning for Generic Object Detection: A Survey," International Journal of Computer Vision, vol. 128, pp. 261–318, 2020.

[3] N. F. F. Alshdaifat, A. Z. Talib and M. A. Osman, "Improved Deep Learning Framework for Fish Segmentation in Underwater Videos," Ecological Informatics, vol. 59, p. 101121, DOI: 10.1016/j.ecoinf.2020.101121 2020.

[4] Z.-C. He, L.-Y. An et al., "Comment on "Deep Learning Computer Vision Algorithm for Detecting Kidney Stone Composition"," World Journal of Urology, DOI: 10.1007/s00345-020-03181-4, April 2020.

[5] A. B. Nassif, I. Shahin et al., "Speech Recognition Using Deep Neural Networks: A Systematic Review," IEEE Access, vol. 7, pp. 19143–19165, 2019.

[6] J. Jiang and H. H. Wang, "Application Intelligent Search and Recommendation System Based on Speech Recognition Technology," International Journal of Speech Technology, pp. 1–8, DOI: 10.1007/s10772-020-09703-0, April 2020.

[7] M. Zhou, X. Wei, S. Kwong et al., "Rate Control Method Based on Deep Reinforcement Learning for Dynamic Video Sequences in HEVC," IEEE Transactions on Multimedia, pp. 1-1, DOI: 10.1109/TMM.2020.2992968, May 2020.

[8] S. F. A. Abuowaida and H. Y. Chan, "Improved Deep Learning Architecture for Depth Estimation from Single Image," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 6, no. 4, pp. 434-445, 2020.

[9] R. S. T. Lee, "Natural Language Processing," in: Artificial Intelligence in Daily Life Book, pp. 157–192, ISBN 978-981-15-7695-9, Springer, 2020.

[10] K. Shuang, Z. Zhang et al., "Convolution–deconvolution Word Embedding: An End-to-end Multi-prototype Fusion Embedding Method for Natural Language Processing," Information Fusion, vol. 53, pp. 112–122, DOI: 10.1016/j.inffus.2019.06.009, 2020.

[11] Spyridon Thermos et al., "Deep Sensorimotor Learning for RGB-D Object Recognition," Computer Vision and Image Understanding, vol. 190, p. 102844, DOI: 10.1016/j.cviu.2019.102844, 2020.

[12] N. Wang, Y. Wang and M. J. Er, "Review on Deep Learning Techniques for Marine Object Recognition: Architectures and Algorithms," Control Engineering Practice, p. 104458, DOI: 10.1016/j.conengprac.2020.104458, 2020.

[13] Qiaoyong Zhong et al., "Cascade Region Proposal and Global Context for Deep Object Detection," Neurocomputing, vol. 395, pp. 170–177, 2020.

[14] Francisco Pérez-Hernández et al., "Object Detection Binary Classifiers Methodology Based on Deep Learning to Identify Small Objects Handled Similarly: Application in Video Surveillance," Knowledge-based Systems, vol. 194, p. 105590, DOI: 10.1016/j.knosys.2020.105590, 2020.

[15] M. Rezaei, H. Yang and C. Meinel, "Recurrent Generative Adversarial Network for Learning Imbalanced Medical Image Semantic Segmentation," Multimedia Tools and Applications, vol. 79, pp. 15329–15348, DOI: 10.1007/s11042-019-7305-1, 2020.

[16] B. Xu, W. Wang, G. Valzon et al., "Automated Cattle Counting Using Mask R-CNN in Quadcopter Vision System," Computers and Electronics in Agriculture, vol. 171, p. 105300, 2020.

[17] M. Bellver, A. Salvador, J. Torres et al., "Mask-guided Sample Selection for Semi-supervised Instance Segmentation," Multimedia Tools and Applications, vol. 79, pp. 25551–25569, DOI: 10.1007/s11042-020-09235-4, 2020.

[18] D. Larlus, J. Verbeek and F. Jurie, "Category Level Object Segmentation by Combining Bag-of-words Models with Dirichlet Processes and Random Fields," International Journal of Computer Vision, vol. 88, pp. 238–253, DOI: 10.1007/s11263-009-0245-x, 2010.

[19] X. Zhao, Y. Satoh et al., "Object Detection Based on a Robust and Accurate Statistical Multipoint-pair Model," Pattern Recognition, vol. 44, no. 6, pp. 1296–1311, 2011.

[20] J. Walsh, N. O'Mahony et al., "Deep Learning *vs.* Traditional Computer Vision," Proc. of the Science and Information Conference (CVC), pp. 128–144, DOI: 10.1007/978-3-030-17795-9_10, Springer, Las Vegas, USA, 2019.

[21] Z. Xue, D. Ming et al., "Infrared Gait Recognition Based on Wavelet Transform and Support Vector Machine," Pattern Recognition, vol. 43, no. 8, pp. 2904–2910, DOI: 10.1016/j.patcog.2010.03.011, 2010.

[22] R. Girshick, J. Donahue et al., "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587, DOI: 10.1109/CVPR.2014.81, Columbus, USA, 2014.

[23] R. Girshick, "Fast R-CNN," Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1440–1448, DOI: 10.1109/ICCV.2015.169, Santiago, Chile, 2015.

[24] S. Ren, K. He et al., "Faster R-CNN: Towards Real-time Object Detection with Region Proposal Networks," Advances in Neural Information Processing Systems, pp. 91–99, [Online], Available: https://arxiv.org/pdf/1506.01497.pdf, 2015.

[25] Yi Li et al., "Fully Convolutional Instance-aware Semantic Segmentation," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2359–2367, DOI: 10.1109/CVPR.2017.472, Honolulu, USA, 2017.

[26] J. Dai, K. He and J. Sun, "Instance-aware Semantic Segmentation *via* Multi-task Network Cascades," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3150–3158, DOI: 10.1109/CVPR.2016.343, Las Vegas, 2016.

[27] K. He et al., "Mask R-CNN," Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 2961–2969, DOI: 10.1109/ICCV.2017.322, Venice, Italy, 2017.

[28] D. Bolya, C. Zhou et al., "YOLACT: Real-time Instance Segmentation," Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 9157–9166, DOI: 10.1109/ICCV.2019.00925,

Seoul, Korea (South), 2019.

[29] Z. Cai and N. Vasconcelos, "Cascade R-CNN: High Quality Object Detection and Instance Segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019.

[30] J. Long, E. Shelhamer and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3431–3440, DOI: 10.1109/CVPR.2015.7298965, Boston, USA, 2015.

[31] K. He, et al., "Deep Residual Learning for Image Recognition," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778, DOI: 10.1109/CVPR.2016.90, Las Vegas, USA, 2016.

[32] J. Hu, L. Shen and G. Sun, "Squeeze-and-excitation Networks," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7132–7141, DOI: 10.1109/CVPR.2018.00745, Salt Lake City, USA, 2018.

[33] T.-Y. Lin, M. Maire et al., "Microsoft COCO: Common Objects in Context," Proc. of the European Conference on Computer Vision (ECCV), pp. 740–755, DOI: 10.1007/978-3-319-10602-1_48, Part of the Lecture Notes in Computer Science Book Series (LNCS, vol. 8693), Springer, 2014.

[34] M. Abadi, A. Agarwal, P. Barham et al., "TensorFlow: Large-scale Machine Learning on Heterogeneous Distributed Systems," Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation (OSDI'16), pp. 265-283, arXiv preprint arXiv: 1603.04467, 2016.

ملخص البحث:

تعدّ خوارزميات التعلُّم العميق واسعة الانتشار وجذّابة في السنوات الأخيرة، وذلك بالنظر الى ما تحقّق بفضلها من إنجازاتٍ في العديد من المجالات. ويكمن التعلُّم العميق في الكشف المستند الى الصُّور وتجزئة المراحل الخاصة بكيانٍ او شيء ما، وتلك مسألة حرجة في حاجةٍ الى المزيد من البحث والاستقصاء.

تهدف هذه الورقة الى اقتراح خوارزمية مبتكرة من أجل تجزئة المراحل في الصور متعددة المواضع، من ثلاث مراحل؛ بهدف البحث في التحديات التي تواجه استخدام خوارزميات تجزئة المراحل في الصور. في المرحلة الأولى، التي تشكل العمود الفقري أو الجزء الأساسي، يجري تحسين خوارزمية تمييز الصور عبر استخلاص المستويات الدنيا والعليا المميزة من الصور موضوع المعالجة. ووحدة البناء الأساسية هي تلك الشبكة المسماة (ResNet)، وهي تتصل مع شبكة الضغط والإثارة التي تعرف باسم (SENet) لكل وحدةٍ من وحدات شبكة (ResNet). وتستخدم شبكة اقتراح المنطقة (RPN) للعمل على تحديد موضع العُنصر الهدف، وتتبعها المرحلة الثالثة التي تقترح طبقة الموضع (RoI) من أجل اختيار الحدود المثالية للتجزئة.

تم إجراء التجارب وتقييمها باستخدام قاعدة بياناتٍ مرجعية للصور تُعرف باسم (COCO). وجرى تقييم الخوارزمية المقترحة من حيث الأداء باستخدام معايير تقييم مرجعية، ومقارنتها بعددٍ من خوارزميات تجزئة المراحل الحديثة الخاصة بالصور. وبينت النتائج أن الخوارزمية المقترحة في هذه الورقة كانت أفضل من خوارزميات تجزئةٍ أخرى معروفة جيداً تستخدم المراحل، وذلك من حيث متوسط الدقة باستخدام قياسات العتبة (AP)، وذلك لعتباتٍ مختلفة.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 07, No. 01, March 2021.

LIVE BIG DATA ANALYTICS RESOURCE MANAGEMENT TECHNIQUES IN FOG COMPUTING FOR TELE-HEALTH APPLICATIONS

Ragaa Shehab, Mohamed Taher and Hoda K. Mohamed

(Received: 20-Nov.-2020, Revised: 11-Feb.-2021, Accepted: 23-Feb.-2021)

ABSTRACT

Enhancing the IoT health monitoring systems used in various environments, such as smart homes and smart hospitals, imply lively analyzing the patients' critical streams (e.g. ECG stream). Conducting these tele-health applications over the traditional cloud violates the deadline constrains of the stream analytics applications, which results not only in performance degradation, but also in inaccurate analytics results due to patient's stream loss. Fog computing can take place within the patient's vicinity and is considered as the best candidate for critically analyzed stream applications. Fog nodes are geo-distributed and are poor in resources, thus a scalable and fault-tolerant resource management platform for stream analytics in fog computing is a must. Current Stream Processing (SP) resource managers are designed for massive resource nodes, deploying them over the poor resource edge fog nodes greatly decreasing the fog infrastructure utilization. Innovative SP resource managers that cope with the fog nature are needed. We propose Fog Assisted Resource Management (FARM) platform based on Apache Hadoop2 resource manager (YARN) for compatible stream/batch analytics. Static FARM (S-FARM) represents two YARN schedulers; per-user and per-module. Results indicate that per-user scheduler overcomes the lack of resources issues of the edge fog nodes, fully utilizes the fog infrastructure and allows the system to expand safely up to its double size. In addition, Differentiated S-FARM scheduler is proposed to support per-user control to the analytic results' accuracy and speed. Stream CardioVascular Disease (S-CVD) application for patient's ECG analytics is simulated in iFogSim to judge the proposed YARN schedulers. The research is pioneer in enhancing the poor resource edge fog node utilization, supporting per-user control to live big data analytics IoT applications and utilizing iFogSim to implement and evaluate the resource manager performance of a stream analytics platform.

KEYWORDS

Fog/Edge computing, Big data analytics, Stream analytics, Apache Hadoop2 YARN schedulers, Per-user control, Analytics accuracy, Fog infrastructural management, Patient monitoring, Smart hospital.

1. INTRODUCTION

According to the National Council on Aging, up to 80% of older adults have at least one chronic health condition that requires continual treatment management. This fact increases the burden on the world's health care systems. The evolution of the IoT technologies and health informatics systems aims to realize the remote patient's monitoring (tele-health) with high quality of care and to make the management of these populations more cost-effective [1]. Tele-health applications allow patients to live more independently and improve their quality of life while reducing the cost of medical care and hospital re-admissions. In addition, online patients' data aids caregivers in early patient state classification, emergency situation management and following up patient adherence to the given treatments.

Data analytics plays an important role in tele-health ecosystems, especially for smarter decision-making within the time constrains; i.e., patient's critical state detection. For this fully distributed data sources, cloud data processing fails to meet the requirement of delay sensitive applications, which results not only in performance degradation, but also in inaccurate analytics results due to patient's stream loss [2]. Fog computing, also known as edge computing [3]-[4], is a distributed computing paradigm that aims to tackle the issue by offloading data analytics and sensitive delay tasks to the edge of the network closer to the data sources, leaving the delay tolerant highly computational tasks to be performed at the cloud. Resource management in fog computing is a challenging issue [5]-[6]. This is because fog allows application modules to be distributed along the fog tiers to provide an enhanced

R. Shehab, M. Taher and H. K. Mohamed are with Computer and Systems Engineering Department, Ain Shams University, Cairo, Egypt.
Emails: rashehab@yahoo.com, mohamed.taher@eng.asu.edu.eg and Hoda.Korashy@eng.asu.edu.eg

application delay and network usage [2], [7]. However, deployment of critical tele-health stream applications in such manner degrades the application performance, because edge fog nodes are geo-distributed, poor in resources and sustain to failure that will affect the patient's experience and prevent achieving the main purpose of tele-health applications [2]. A scalable and fault tolerant Stream Processing (SP) platform in fog computing overcomes these issues. The on-market SP resource managers are designed for massive resource nodes. Deploying these resource managers over poor resource edge fog nodes degrades the fog infrastructure utilization. Innovative SP resource managers in fog computing are needed. All the reviewed literature depends on real cluster implementation. To the best of our knowledge, the research is pioneer in utilizing iFogSim simulator [8] to implement and evaluate the performance of SP platform resource manager. This contribution may guide researchers in implementing and judging the resource management performance of other various on-market SP platforms; i.e., Apache (Samza, Flink, Spark,...).

This work proposes a Fog Assisted Resource Management (FARM) platform based on YARN for compatible short-term and long-term big data analytics. Static FARM (S-FARM) represents YARN schedulers. Two schedulers are proposed to control the fog nodes CPU load: per-user and per-module. Per-user scheduler is a YARN scheduler that copes with the edge fog nodes lack of resources. In addition, Differentiated S-FARM scheduler is proposed to allow per-user control to the analytics results QoS. Analytics results are controlled by the analytics tuples' Million Instructions Per Second (MIPS) to represent accurate *versus* fast results. Stream CardioVascular Disease (S-CVD) application is modelled. It lively analyzes the patient's ECG streams to conduct the patient's state using a linear classifier machine learning tool. IFogSim is used to judge the application and the fog infrastructure performance under the proposed YARN schedulers.

The paper is organized as follows: Section 2 introduces a brief background about live big data analytics. Section 3 provides a literature review. Section 4 presents the research methodology. Section 5 presents S-CVD application and system model. Section 6 presents FARM platform and YARN to fog mapping. Section 7 presents the YARN schedulers (S-FARM algorithms). Section 8 analyzes the application and infrastructure performance. Finally, Section 9 concludes the paper and presents suggestions for future work.

2. LIVE BIG DATA ANALYTICS

Big data is characterized by its volume, variety, veracity, velocity and value. Big data could be analyzed either in stream or batch mode [11]-[12], see Table 1.

Table 1. Comparison between stream and batch data analytics Modes.

Differences	**Stream Mode**	**Batch Mode**
Mode	Short-term (live) analytics	Long-term analytics
Management target	Transient streams	Persistent data
Amount of data	Unknown in advance	Finite
Processing model	On the fly	Store then process
Query model	Continuous	One-time query
Access model	Sequential access	Random access
Result repeatability	Nearly impossible	Easy
Result update	Incremental update	Global update
Focus of processing	Low latency	High accuracy
Platforms	Storm, Spark, Samza, Flink,…	Apache Hadoop,…

In the domain of healthcare, IoT big data has several challenges, including high data rate with variable volume, semi-structured or unstructured format (i.e., echo image, voices), correlation across several dimensions (i.e., time, location) and its social relations among related healthcare devices [1]. Analyzing the healthcare IoT streams at the fog network has a set of advantages, including real-time handling, user-centric processing, user's mobility support and geo-distribution, location and context awareness applications support [10].

Stream Analytics for Critical Healthcare Decision-making: A stream is defined as a sequence of

data elements ordered by time. Each data element has a time stamp that measures the data order. Stream processing SP is a one-pass data processing that aims to achieve low processing latency by keeping data in motion. The complete stream analytics data life cycle includes [12]:

1. Data generation stage.
2. Data collection and aggregation stage: from different distributed sources.
3. Messaging and buffering stage: IoT streams are gathered into a centralized buffer.
4. Continuous Logic Processing (CLP) stage: processes data according to the designed continuous logic. Current CLP Systems (CLPS) are scalable and fault-tolerant.
5. Presentation and storage stages: deliver the insights to end users and store them.

Table 2 presents a comparison between Apache's Stream Processing (SP) platforms that represent the third-generation CLPS. SP platforms are characterized by [12]-[13]:

- Programming components of the CLPS: graph name, nodes and edges.
- Type of process: client (graph builder), task scheduler and task executer.
- CLPS capability of accurate recovery of the same processing results when system failures occur.
- State consistency of all participating components during processing, which is related to the fault recovery methods implemented by the system.

Apache Hadoop2 YARN (Yet Another Resource Management Negotiator) [18]-[19] can serve as the core of various Apache's SP platforms. YARN can serve as the core of various Apache's SP platforms. Thus, the proposed FARM platform is based on YARN for compatible stream/batch analytics.

Table 2. Comparison between open-source stream analytics platforms.

Apache's SP Platform	**Storm [14]**	**Spark [15]**	**Samza [16]**	**Flink [17]**
Processing Type	Stream	Stream-Batch	Stream-Batch	Stream-Batch
Type of processes: 1-Client 2-Task scheduler 3-Task executer	1-Topology builder 2-Nimbus and Zookeeper 3-Workers	1-Spark DAG 2-YARN scheduler or standalone 3-Workers	1-User-defined 2-YARN scheduler or Zookeeper 3-Workers	1-Graph builder 2-YARN scheduler or standalone 3-operators executer
Accurate Recovery	Yes	Yes	Yes	Yes
State Consistency	No	Yes	Yes	Yes
Adopted by	Twitter, Yahoo	eBay Inc.	LinkedIn	Research gate

3. LITERATURE REVIEW: STREAM ANALYTICS IN FOG COMPUTING

Stream analytics research in fog computing could be classified as:

- **Stream Analytics Platform Deployed Stream Applications:** [20]-[24]. See Table 3. In this literature, performance was measured for general-purpose applications. It did not consider healthcare stream applications with high sensor rates and critical reading that concerns patient's safety and security. In addition, no study considered a single platform for both short/long-term analytics that is required for accurate remote patient monitoring.
- **Healthcare Stream Analytics Platforms:** proposed to deploy healthcare stream applications only. The healthcare application modules are placed on the fog network according to the type of the analytics task; i.e., [25]-[27] for healthcare data critical analysis task and [28]-[29] for healthcare data critical control task. See Table 4. In this literature, three tiers of IoT data network are used for permanent task allocation regardless of the encountered application performance: smart watch or smart phone tier was used for data collection tasks, fog/cloud tier was used for data computations tasks and the cloud tier was used for data storage and long-term analytics tasks.
- **Stream Analytics Platform Deployed Healthcare Stream Applications**: [30]-[32]. See Table 5. In this literature, the evaluation method depends on real cluster implementation only. In addition, the lack of resources of the edge fog nodes has not been addressed.

To the best of our knowledge, no research addressed the edge fog node lack of resources, provided

per-user control to the accuracy and speed of the analytic results or utilized iFogSim as an innovative tool to implement and judge the performance of a stream analytics platform resource manager.

Table 3. Stream analytics platform deployed general-purpose application in fog computing.

REF.	Platform	Perf. Metric	Scheduler	Imp.
[20]	Storm	Utilization, latency, inter-node traffic	distributed Storm scheduler that adapts to fog network's changes	R
[21]	Storm	Comm. latency to external IoT actuators or databases	modified Storm with a decision module that decides whether to place selected tasks on edge devices at run time	R
[22]	Storm, Nimbus	Latency, average inter-node traffic	modified Nimbus by adding offline and online schedulers that analyze the topology and monitor the effectiveness of the schedule at run time	R
[23]	Spring cloud dataflow	Optimization problem	resource elasticity mechanism to deal with changing rates of streaming data	R
[24]	Spark	Job completion time, scalability, power consumption	Modified Spark to utilize the whole processing capacity of all the available edge devices	R

R: Real Implementation.

Table 4. Healthcare stream analytics platform in fog computing.

REF.	Perf. Metric	Per. Task Allocation	Application	Imp.
[25]	Data size, processing and transmitting time	At LAN level	ECG monitoring	R
[26]	Processing time, system reliability	At PAN level for transport scenario	ECG feature extraction	R
[27]	End-to-end data delay, mobile battery life time	At PAN level for user mobile scenario	Patient monitoring	R
[28]	Patient deterioration and re-admission incidence rate	User mobile scenario	Oxygen level control	R
[29]	Stable patient heart beat	At LAN level for hospital scenario	Pacemaker monitoring	R

Per.: Permanent Task Allocation.

Table 5. Stream analytics platform deployed healthcare stream applications in fog computing.

REF	Platform	Perf. Metric	Scheduler	Application	Imp.
[30], [31]	Storm, Kafka	-	Kafka and Storm cluster architecture for S/B analytics	i.e., pervasive health	-
[32]	Flink, Kafka	CPU, memory usage, average data loss	Stream computing at Kafka's broker and Flink's cluster processing layer	Anomaly detection REALDISP dataset	R

4. RESEARCH METHODOLOGY

In this research, we aim to develop a FARM platform for critical tele-health stream analytics applications. Figure 1 presents our research methodology.

5. S-CVD APPLICATION MODEL

Stream CardioVascular Disease (S-CVD) application analyzes an ordered Electrocardiogram (ECG) stream with a sensor transmission rate of (25ms: 1000ms). The application is modelled as a Directed Acyclic Graph (DAG), where modules are represented as vertices and inter-module communications are represented as edges, see Figure 2. Client module accepts the sensor ECG stream, adding any

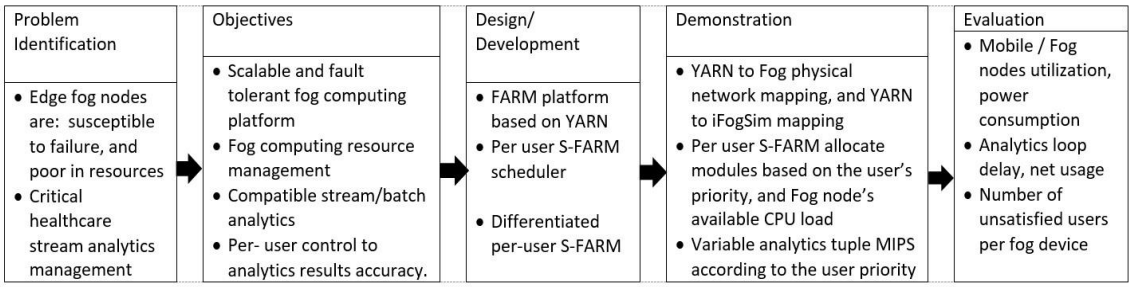


Figure 1. Research methodology.

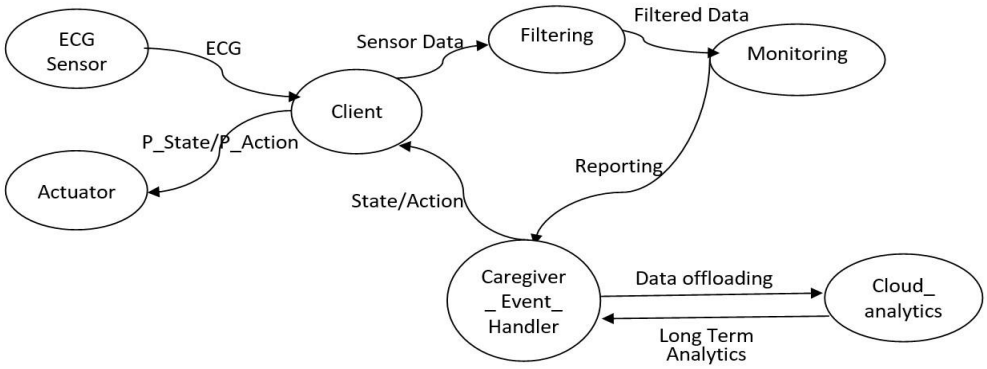


Figure 2. S-CVD application model.

related information and securing the packets. Filtering module cleans the data, eliminates inaccurate and out-of-range readings. Monitoring module is a linear classifier machine learning tool that continuously classifies the patient's state. Caregiver module gets the analytics results and reports the online consultant about the current patient state. Online consultant has a complete access to the patient's electronic health record along with a knowledge base that aids him/her in decision-making. In emergency situations, the consultant can send an ambulance or contact the patient's family for help. After analyzing the data, the patient is informed by his state or with the ongoing actions. Cloud analytics module is located at the remote cloud and is responsible for long-term analytics and managing the analytical operators supervised machine learning for the patient state classifiers.

System Model

- Body Area Network of ECG sensors are connected to the patient and through WiFi or Bluetooth to the smart e-health gateway 1.
- Smart e-health gateway 1 (Mobile) is the patient's smart phone that carries out the data collection task; i.e., S-CVD (Client module).
- Smart e-health gateway 2 (Dept) is located at the patient's vicinity (smart home, smart vehicle or smart hospital ward) and carries out the data analysis tasks of the tele-health applications. It is connected to the healthcare center.
- Remote healthcare center (Proxy-Server) is located at the smart hospital and carries out the decision-making and permanent data storage tasks; i.e., S-CVD (Caregiver module). Data management tasks; i.e., S-CVD (Filtering, Monitoring modules) could be utilized at smart gateway or at the proxy-server according to the required application and fog infrastructure performance.
- Remote cloud is responsible for long-term data analytic; i.e., S-CVD (Cloud analytics module).

Deploying YARN over this system model enables a scalable and fault-tolerant platform for the tele-health application. Also, it preserves the patient's security and privacy, because his/her sensory data is processed locally at his/her vicinity or at his/her smart hospital.

"Live Big Data Analytics Resource Management Techniques in Fog Computing for Tele-health Applications", R. Shehab, M. Taher and H. K. Mohamed.

6. FOG ASSISTED RESOURCE MANAGEMENT (FARM) PLATFORM

Based on the complete life cycle of the IoT stream analytics systems [12] and the Apache Hadoop2 YARN architecture [18]-[19], Figure 3 is proposed to represent the FARM platform based on YARN for compatible stream/batch analytics in the fog/cloud system. For a smart hospital system model, the fog nodes (i.e., smart e-health gateway 2 and the remote healthcare center) represent the YARN nodes that carry out the stream analytics. A remote cloud represents YARN nodes that carry out batch analytics tasks for multiple hospital branches that belong to the same owner. For the S-CVD application, patients' streams are queued in messaging system (i.e., Kafka) before turning into the hospital's stream analytics platform. The Hadoop Distributed File System (HDFS) can comprise a large number of directly connected individual fog nodes at the smart hospital. For decision-making, caregiver queries are sent to HDFS in fog network and cloud data center.

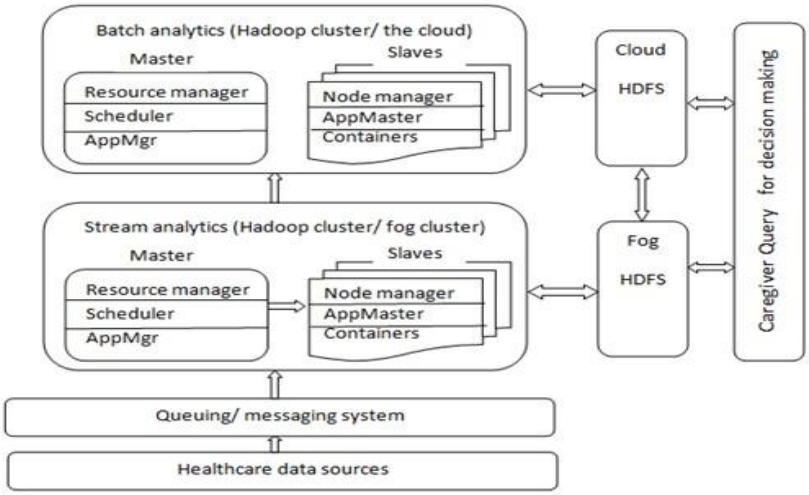


Figure 3. FARM platform based on YARN for compatible stream and batch analytics.

YARN to Fog Mapping: Table 6 proposes YARN to fog physical entity mapping, where Fog Manager Server and Fog Domain Servers are presented to carry out the main YARN responsibilities. Also, it presents YARN to iFogSim mapping.

Table 6. YARN concept to fog physical entity components and iFogSim mapping.

YARN Concept	Fog Physical Entity	iFogSim Mapping
ResourceManager (Master): **Scheduler:** global resource scheduler **Application Manager:** follows up the progress for executing the application's specific ApplicationMaster	**Fog Manager Server (Master):** Scheduler: determines how application's modules are placed across fog devices upon submission of the application Application Manager: monitors the performance and reschedule resources of each host fog device to the application modules **Fog Master Server:** Console for user interface and modules	**Scheduler:** represented by: controller class, per-user-Basic-MP class, per-module-MP class, per-user-Diff-MP class, StreamOperatorScheduler, TupleScheduler class **ApplicationManager**: CheckDelay-Enh-Diff method (FogDevice class), MY-updateAllocatedMips method (FogDevice class)
NodeManager (slaves): Host: tracks the running Virtual Machines (VMs) Container: VMs ApplicationMaster: global monitoring daemon, negotiates resources for VMs.	**Fog Domain Server:** one or multiple tiers **Wireless End Fog Nodes:** light-weight cluster slaves ApplicationMaster: allocated over certain host or centralized	Host: FogDevice class. Container: AppModule class ApplicationMaster: main class

7. FARM ALGORITHMS

This work proposes Static FARM (S-FARM) representing YARN ResourceManager(Scheduler). Other YARN components will be studied as future work, see Table 6.

7.1 Basic S-FARM Algorithm

S-FARM schedulers determine how the application's modules are placed across the fog nodes upon the application submission. Two modes are used:

- Per-user Mode, where each user has his own Virtual Machines VMs carrying out his own workload. Resources are managed by placing the user's modules (VMs) individually and one by one according to the user's priority, until the required RM objectives are reached. This mode allows explicit user differentiation according to the user's Service Level Agreement (SLA).
- Per-module Mode, where all users' workloads are placed together and carried out over one VM. Each VM represents one application module. Resources are managed by placing the whole module according to the application DAG until the required RM objectives are reached.

Per-user S-FARM scheduler, Algorithm 1, places the high-priority user modules at its closest fog nodes first; if there is a shortage in the closest fog node computational resources, the remaining users' (the lower priorities) modules are shifted up to the next fog node in the fog node path hierarchy. Per-module S-FARM scheduler, Algorithm 2, places and shifts up the modules that carry the whole user instances according to the application's DAG.

UsrMIPS in algorithm 1 and ModMIPS in algorithm 2 represent the analytics tuple's CPU length. They are permanently assigned by caregivers based on the requested analytics accuracy, where heavy-weight tuples represent accurate results and light-weight tuples represent fast results. They are used across all the module's edges to calculate the required CPU load for this module $CPULoad_{Mod}^{Req}$.

Algorithm 1: S-FARM Per-user Scheduler Algorithm

```
1: procedure SFARM-PERUSER(PATHS,App,Usrprio,UsrMIPS)
2:     Arrange all FogDevices within PATHS (leaf to root traversal)
3:     Arrange PATHS according to the Usrprio (high to low priority users).
4:       for (P ∈ PATHS) do
5:           for (endFogDev ∈ P) do
6:               for (UsrModule ∈ App) do
7:                   Assign the UsrMIPS (analytics tuple's CPUlength).
8:                       if CPULoad_Dev^Curr + CPULoad_Mod^req .≤ CPULoad_Dev^max av . then
9:                       Place the UsrModule on the endFogDev
10:                  else
11:                      Shift up the UsrModule to ParentFogDev
```

Algorithm 2: S-FARM Per-module Scheduler Algorithm

```
1: procedure SFARM-PERMODULE(PATHS,App,ModMIPS)
2: Arrange all FogDevices within PATHS (leaf to root traversal)
3: for (P ∈ PATHS) do
4:     for (endFogDev ∈ P) do
5:         for (Module ∈ App) do
6:         Assign the ModMIPS (analytics tuple CPUlength).
7:         if (Module is Placed on this endFogDev )
8:             if ( CPULoad_Dev^Curr + CPULoad_Mod^req .≤ CPULoad_Dev^max av .) then
9:                 Place this Module instance on this endFogDev
10:            else
11:                Shift up the whole Module to ParentFogDev
12:        else
13:            if CPULoad_Dev^Curr + CPULoad_Mod^req .≤ CPULoad_Dev^max av .then
14:                Place the Module's first instance on this endFogDev.
```

In addition, in both algorithms, the modules are placed at a fog device until the device's maximum available $CPULoad_{Dev}^{\max av}$ is reached. In this study, dept and proxy fog devices are loaded up to 100% of their maximum CPU load, while the user's mobile is loaded to <= 0.15% of its maximum CPU load. Increasing the allowable mobile's CPU load permits more application modules to be placed at the user's mobile, which is not a desired performance.

Both schedulers shift up the modules to the next level on the fog devices' path hierarchy if:
$CPULoad_{Dev}^{Curr} + CPULoad_{Mod}^{req} \geq CPULoad_{Dev}^{\max av}$.

7.2 Differentiated S-FARM Algorithm

Basic S-FARM assigns a constant analytics tuples' MIPS to all users. Differentiated S-FARM assigns a variable analytics tuples' MIPS to each user, according to the user's requested analytics accuracy and speed.

8. RESULTS

Simulation parameters are the same used in [2]. S-CVD are tested for heavy-weight processing modules (10MB RAM, 2000 MIPS) and tuples (2000 MIPS and 500 Byte network length). Both modules (VMs) and tuples (Tasks) are time-shared scheduled, with device scheduling interval (10 ms) and application scheduling interval (300 ms). Fog devices' CPU processing is: Mobile (1000 MIPS), Dept (2800 MIPS) and ProxyServer (16800 MIPS) to express the edge fog node lack of resources. Simulation time is up to 600000ms. The system is configured by the number of depts and the number of mobiles per each dept. The simulated (Depts/Mobile) are: 1D/3M, 2D/4M, 2D/6M, 2D/8M and 3D/10M. Performance Metrics include:

- Stream Analytics Loop Delay (ALD-Mean): the average delay for all tuples within the analytic loop for all users. The analytic loop executes the modules: ECG, client, filtering, monitoring and caregiver, in order.
- Stream Analytics Loop Delay User (ALD-User): the average delay for all tuples within the analytics loop for a single user.
- Standard Deviation of the Analytics Loop Delay (ALD-SD): for N users, the standard deviation is the root of variance:

$$\sigma^{\wedge 2} = \frac{1}{N}\sum_{i=1}^{N}(ALD_{Useri} - ALD_{Mean})^2$$

- Percentage of unsatisfied users per fog device: percentage of users with analytics loop delay greater than the delay threshold at this device. For SP, the max. allowed analytics delay for all devices should be less than or equal to the sensor's transmission rate.
- Device's Power Consumption (Watt/hour): measured by the device's utilization percentage over the simulation period.
- Total Network Usage (kByte).

8.1 Basic S-FARM Performance

8.1.1 At Various System Configurations

Figure 4 shows the analytics loop delay for ECG with a sensors' transmission rate of 50ms, at two user's Mobile CPU load percents (0.1% and 0.15%) of its max. CPU load. Results indicate that at 1D3M, per-user and per-module modes have the same analytics delay; where the modules are placed similarly in both algorithms. Analytics delay is acceptable (<50ms) under all system configurations for per-user and per-module modes with Mobile CPU load of 0.15%. Expanding the system to 3D8M, both per-user and per-module modes with Mobile CPU load of 0.1% encounter unacceptable analytics delay (>50ms), but per-user mode delay is within the reasonable limit.

Figure 5 shows the energy consumption of dept and proxy fog devices. Starting from 2D/6M configuration, per-module dept device flushes all its load to the proxy device and works by its idle power. At 3D/8M per-module mode, proxy device reaches its maximum allowable CPU load, flushes

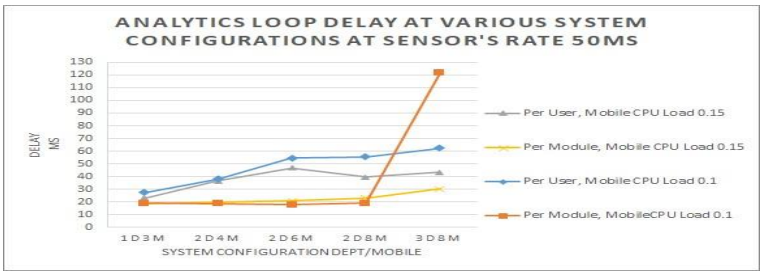


Figure 4. Analytics loop delay under various system configurations.

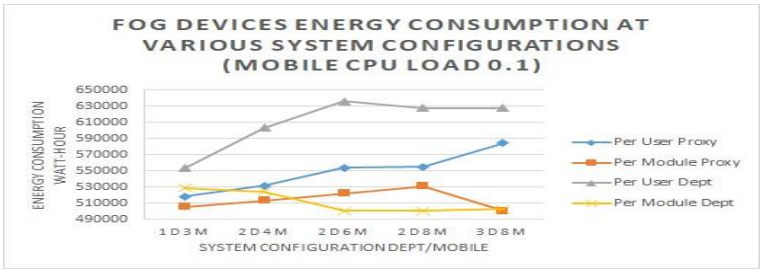


Figure 5. Fog devices energy consumption under various system configurations.

all its workload to the remote cloud and works also by its idle power, leaving the smart hospital fog system completely un-utilized. Per-user mode dept and proxy devices work under all system configurations; even if the device's maximum CPU load is reached, the device shifts up selected user's modules only to the higher fog device. Thus, per-user mode permits for safer system expansion than per-module mode. The same information is deduced from Figure 6 that shows the dept and proxy fog devices CPU utilization at 3D/8M configuration. Per-user devices have 100% CPU utilization, while pe- module devices have 0% CPU utilization. Cloud datacenter carries out the heavy cloud analytics module for batch analytics, consumes higher energy in per-user mode than in per-module mode. Thus, cloud datacenter by its massive resources is not preferable to work in per-user mode. We suggest a hybrid mode of operation, where the limited resource fog nodes can work in per-user mode to save the system expandability, where the massive resource nodes like the cloud datacenter should work in the pe- module mode to save its power consumption. Also, within the same fog node, the hybrid mode could be studied to optimize the fog infrastructure energy consumption while allowing for safe system expansion.

Figure 7 shows Mobile energy consumption. Results indicate that Mobile devices consume lower energy with per-user mode under all system configurations. Also, more energy is consumed when the mobile CPU load is 0.15% of its max. available CPU load.

8.1.2 At Various Sensor Transmission Rates

Figure 8 shows analytics loop delay under various sensor transmission rates, at 2D/4M and 3D/8M configurations. AT 2D/4M, all sensor transmission rates >=25ms are acceptable for per-module mode, while sensor transmission rates >= 40 ms are acceptable for pe- user mode. At 3D/8M, per-module analytics delay is acceptable for sensor rates >=500ms, while per-user analytics delay is acceptable for rates >= 100ms.

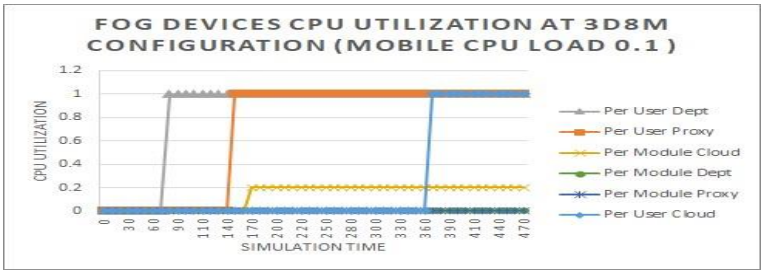


Figure 6. Fog devices CPU utilization at 3D/8M configurations.

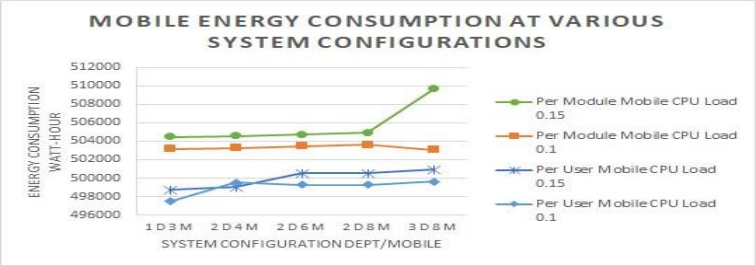


Figure 7. User mobile energy consumption under various system configurations.

Figure 8. Analytics delay at various sensor transmission rates.

Figure 9 shows that at 2D/4M, per-user network usage is lower than that of per-module mode under all the sensor transmission rates. Same result is obtained under various system configurations and tuple MIPS.

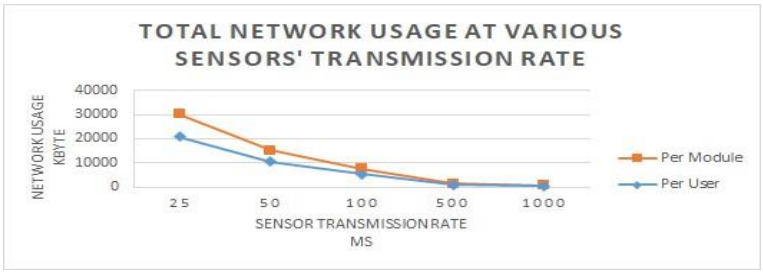


Figure 9. Total network usage at various sensor transmission rates.

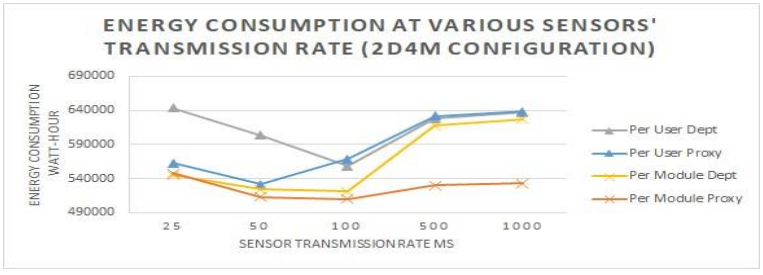


Figure 10. Fog devices energy consumption at various sensor transmission rates.

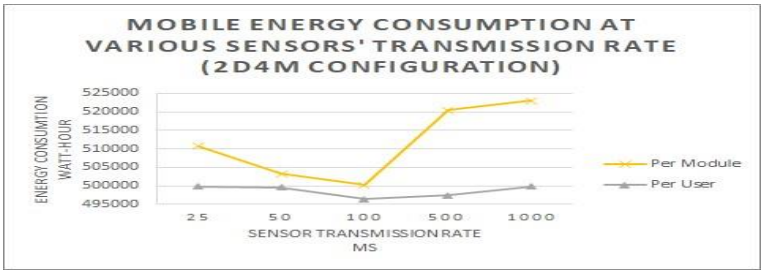


Figure 11. User mobile energy consumption at various sensor transmission rates.

Figure 10 shows the energy consumption of dept and proxy devices. Sensor transmission rate is not linearly affecting fog device energy consumption. Sensor rates 100ms gives the lowest devices' energy consumption under both per-user and per-module modes. The same information is deduced from Figure 11, where the minimum Mobile energy consumption is obtained at a sensor transmission rate of 100ms. Under all rates, per-user mode saves Mobile energy more than per-module mode.

8.1.3 At Various Analytics Tuples' MIPS

Varying the analytics modules (Filtering, Monitoring and Caregiver) and their analytics tuples between 500 and 4000 MIPS, while keeping the remaining modules constant to their original MIPS, Figure 12 shows delay under two situations. Situation1: transmission rate 50ms and 2D4M configuration, we found that delay is acceptable (<50ms) under all analytics tuples' MIPS for per -ser and per-module modes. Situation2: transmission rate 100ms and 3D8M configuration. In per-user mode, the variation in analytics tuples' MIPS doesn't linearly affect the delay and the delay is acceptable (<100ms) under all analytics tuples' MIPS. In per-module mode, delay is acceptable only for analytics tuples' MIPS <= 1000 MIPS.

Figure 13 shows the analytics tuples' MIPS limit that affects the dept utilization. Situation1, analytics tuples' MIPS >= 3000, makes per-module dept CPU utilization=0. Situation2, analytics tuples' MIPS >= 1000, makes per-module dept CPU utilization=0. All tuples' MIPS make per-user mode dept work by its full utilization.

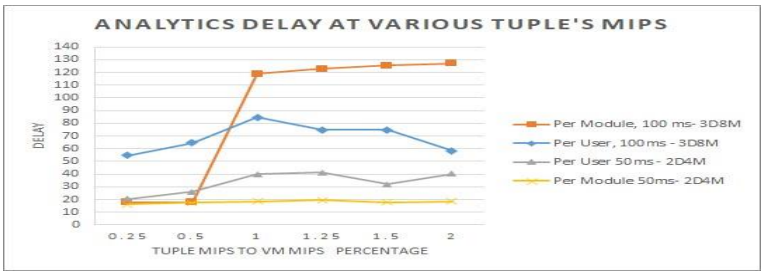


Figure 12. Analytics delay under various Tuple MIPS.

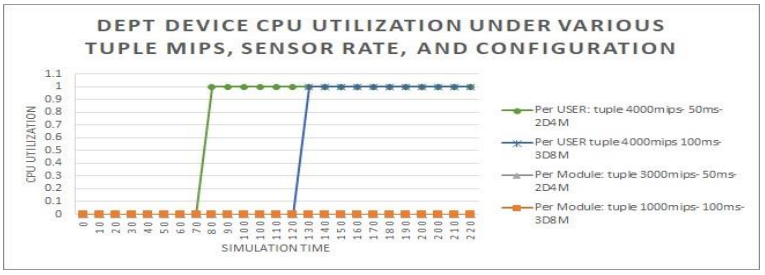


Figure 13. Dept CPU utilization limits under various Tuple MIPS.

Safe Stream Analytics under S-FARM at 3D8M Configuration

Tracing the analytics tuples' MIPS limit that is necessary to obtain an acceptable analytics delay while keeping the dept device working, we found that:

- Rate 50ms: per-user tuple MIPS <= 400; per-module tuple MIPS <= 1000.
- Rate 100ms: per-user tuple MIPS <= 4000; per-module tuple MIPS <= 1000.

8.2 Differentiated S-FARM Performance

Differentiated S-FARM is studied by considering the safe analytics delay limit at 3D8M for per-user basic S-FARM. Two situations have been simulated. Situation1: a random analytics tuples' MIPS between (500:4000) with sensor rate 100ms. Situation2: a random analytics tuples' MIPS between (50:400) with sensor rate 50ms. The average of 10 simulation runs is figured out at each case to test the dept device safe capacity under four configurations: 1D4M, 1D6M, 1D8M, 1D10M.

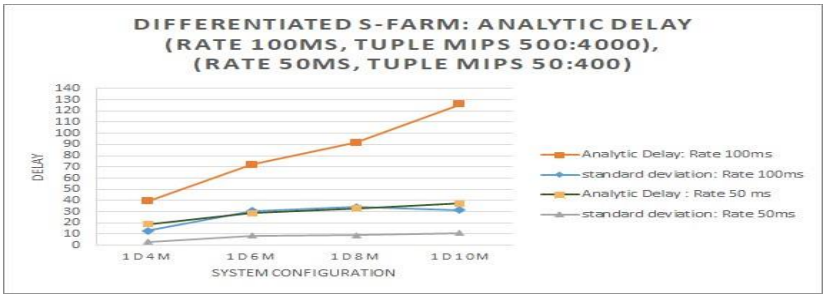


Figure 14. Differentiated S-FARM: analytics delay.

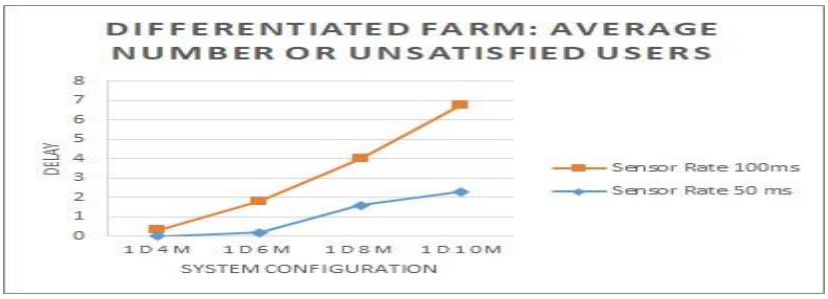


Figure 15. Differentiated S-FARM: average number or unsatisfied users.

Figure 14 shows that average analytics delay is acceptable up to 8 Mobiles per dept for 100 ms sensor rate, but the standard deviation is very high (30 ms) due to the variation in the users' analytics tuples MIPS and the delay is not acceptable for 4 users (50%) on average, as seen in Figure 15, while the average analytics delay is acceptable up to 10 Mobiles per dept for 50 ms sensor rate, but the standard deviation is 10 ms and the delay is not acceptable for only 2 users (20%) on average, as seen in Figure 15.

WORK LIMITATION

To minimize the number of unsatisfied users, performance monitoring should be done on the application run. Analytics delay should be calculated at each fog device to discover the risky users and the risky devices that may cause a problem. If a user's delay at a device exceeds his allowable analytics delay at this device, the device resources should be managed by reallocating more resources to that user (enhanced time-shared scheduling). This will be implemented by the Dynamic FARM (D-FARM) algorithm that represents the YARN ResourceManager (ApplicationManager). It monitors the performance and reschedules resources either locally or by migration to the risky user VMs.

9. CONCLUSIONS AND FUTURE WORK

Fog Assisted Resource Management FARM platform based on YARN for compatible short-term/long-term data analytics is presented. S-FARM is proposed using per-user and per-module modes; it represents the YARN ResourceManager (Schedulers). S-FARM schedulers are simulated over iFogSim. Results indicate that although per-module scheduler minimizes the analytics delay and energy consumption, it is a risky scheduler. It leaves fog devices un-utilized in case that it encounters a shortage in its CPU resources. Per-module scheduler shifts up the whole module to a higher fog device, if there is an increase in: the number of users, the sensor transmission rate or the analytics tuple MIPS. In addition, per-module consumes the user's mobile energy and the network usage more than the per-user scheduler under all the simulated scenarios.

Conducting stream analytics over the poor resources fog nodes, per-user scheduler allows for safer system expansion. If there is a shortage in the device's CPU resources, selected users' modules only could be shifted up to the higher fog device in the path hierarchy. Although being better for the fog infrastructure utilization, per-user scheduler has an average analytics loop delay higher than in per-module mode. Per-user analytics tuples' MIPS should be adjusted carefully under the variable system configurations and transmission rates to allow for a safe stream analytics platform that avoids losing

any sensor reading or violating the stream analytics restrictions of the continuous query result.

Managing the fog network resources, differentiated S-FARM scheduler is the best methodology for the per-user control to the live analytic results, as it allows users to request their specified analytics tuple MIPS and thus their analytics QoS. Heavy-weight analytics tuples allow accurate analytics results, while light-weight analytics tuples allow fast analytics results and allows accommodating more users per fog device.

The application's mean analytics delay and the standard deviation are not sufficient parameters to judge the resource management algorithms' performance. Maximum users' analytics delay should also be figured out at each simulation run, in order to figure out whether there is any loss in the user's stream.

The future work is to minimize the number of unsatisfied users of the differentiated S-FARM algorithm by monitoring the stream analytics application and fog infrastructure performance, as well as to propose Dynamic FARM (D-FARM) that represents the YARN ResourceManager (ApplicationManager) with enhanced time-shared scheduling algorithm to support per-user differentiation.

REFERENCES

[1] F. A. Kraemer, A. E. Braten, N. Tamkittikhun and D. Palma, "Fog Computing in Healthcare: A Review and Discussion," IEEE Access, vol. 5, pp. 9206–9222, 2017.

[2] R. A. Shehab, M. Taher and H. K. Mohamed, "Fog Enabled Health Informatics System for Critically Controlled Cardiovascular Disease Applications," Proceedings of the International Conference on Health Informatics & Medical Systems, pp. 35-41, ISBN: 1-60132-500-2, Copyright ' 2019 CSREA Press, United States of America, 2019.

[3] K. H. Abdulkareem, M. A. Mohammed, S. S. Gunasekaran, M. N. Al-Mhiqani, A. A. Mutlag, S. A. Mostafa, N. S. Ali and D. A. Ibrahim, "A Review of Fog Computing and Machine Learning: Concepts, Applications, Challenges and Open Issues, " IEEE Access, vol. 7, pp. 153123–153140, 2019.

[4] S. Parveen, P. Singh and D. Arora, "Fog Computing Research Opportunities and Challenges: A Comprehensive Survey," Proceedings of the 1st International Conference on Computing, Communications and Cyber-Security (IC4S 2019), Part of the Lecture Notes in Networks and Systems Book Series LNNS, vol. 121, pp. 171–181, DOI: 10.1007/978-981-15-3369-3_13, Springer Singapore, 2020.

[5] R. A. Shehab, M. Taher and H. K. Mohamed, "Resource Management Challenges in the Next Generation Cloud Based Systems: A Survey and Research Directions," Proc. of the 13th International Conference on Computer Engineering and Systems (ICCES), pp. 139–144, Cairo, Egypt, Dec. 2018.

[6] A. A. Mutlag, M. K. Abd Ghani, N. Arunkumar, M. A. Mohammed and O. Mohd, "Enabling Technologies for Fog Computing in Healthcare IoT Systems," Future Generation Computer Systems, vol. 90, pp. 62 – 78, 2019.

[7] A. A. Mutlag, M. Khanapi Abd Ghani, M. A. Mohammed, M. S. Maashi, O. Mohd, S. A. Mostafa, K. H. Abdulkareem, G. Marques and I. de la Torre D´ıez, "MAFC: Multi-agent Fog Computing Model for Healthcare Critical Tasks Management," Sensors, vol. 20, no. 7, Article no. 1853, DOI: 10.3390/s20071853, 2020.

[8] H. Gupta, A. Vahid, A. Dastjerdi, S. K. Ghosh and R. Buyya, "IFOGSIM: A Toolkit for Modeling and Simulation of Resource Management Techniques in the Internet of Things, Edge and Fog Computing Environments," Software: Practice and Experience, vol. 47, no. 9, pp. 1275–1296, 2017.

[9] M. Marjani, F. Nasaruddin, A. Gani, A. Karim, I. A. T. Hashem, A. Siddiqa and I. Yaqoob, "Big IoT Data Analytics: Architecture, Opportunities and Open Research Challenges," IEEE Access, vol. 5, pp. 5247–5261, 2017.

[10] H. Dubey, J. Yang, N. Constant, A. M. Amiri, Q. Yang and K. Makodiya, "Fog Data: Enhancing Tele-health Big Data through Fog Computing," Proceedings of the ASE Big Data & Social Informatics, DOI: 10.1145/2818869.2818889, New York, USA, Association for Computing Machinery, 2015.

[11] S. K. Sharma and X. Wang, "Live Data Analytics with Collaborative Edge and Cloud Processing in Wireless IoT Networks," IEEE Access, vol. 5, pp. 4621–4635, 2017.

[12] X. Liu, A. Dastjerdi and R. Buyya, "Chapter 8 - Stream Processing in IoT: Foundations, State-of-the-art and Future Directions," Internet of Things By: R. Buyya and A. V. Dastjerdi, Eds., pp. 145 – 161, Morgan Kaufmann, 2016.

[13] S. Kamburugamuve and G. C. Fox, "Survey of Distributed Stream Processing for Large Stream Sources," Proc. of SPIDAL: CIF21 DIBBs: Middleware and High Performance Analytics Libraries for Scalable Data Science, DOI: 10.13140/RG.2.1.3856.2968, 2016.

[14] Apache Storm, "Storm Apache," [Online], Available: http://storm.apache.org.

[15] Apache Spark, "Spark Apache," [Online], Available: https://spark.apache.org.

[16] Apache Samza, "Samza Apache," [Online], Available: https://samza.apache.org.

[17] Apache Flink, "Flink Apache," [Online], Available: https://ink.apache.org.

[18] Apache Hadoop, "Apache Hadoop," [Online], Available: http://hadoop.apache.org.

[19] Edureka, "Hadoop Tutorials," [Online], Available: https://www.edureka.co/blog/hadoop-tutorial.

[20] V. Cardellini, V. Grassi, F. L. Presti and M. Nardelli, "On QoS-aware Scheduling of Data Stream Applications over Fog Computing Infrastructures," Proc. of the IEEE Symposium on Computers and Communication (ISCC), pp. 271–276, Larnaca, Cyprus, July 2015.

[21] A. Papageorgiou, E. Poormohammady and B. Cheng, "Edge-Computing-Aware Deployment of Stream Processing Tasks Based on Topology-external Information: Model, Algorithms and A Storm-based Prototype," Proc. of the IEEE International Congress on Big Data (BigData Congress), pp. 259–266, San Francisco, USA, June 2016.

[22] L. Aniello, R. Baldoni and L. Querzoni, "Adaptive Online Scheduling in Storm," Proceedings of the 7th ACM International Conference on Distributed Event based Systems, (DEBS 13), pp. 207–218, New York, USA, ACM, 2013.

[23] C. Hochreiner, M. Vogler, S. Schulte and S. Dustdar, "Elastic Stream Processing for the Internet of Things," Proceedings of the IEEE 9th International Conference on Cloud Computing (CLOUD), pp. 100–107, San Francisco, USA, June 2016.

[24] N. Maleki, M. Loni, M. Daneshtalab, M. Conti and H. Fotouhi, "SoFA: A Spark-oriented Fog Architecture," Proc. of the 45th Annual Conference of the IEEE Industrial Electronics Society (IECON 2019), vol. 1, pp. 2792–2799, Lisbon, Portugal, 2019.

[25] T. N. Gia, M. Jiang, A. Rahmani, T. Westerlund, P. Liljeberg and H. Tenhunen, "Fog Computing in Healthcare Internet of Things: A Case Study on ECG Feature Extraction," Proc. of the IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, pp. 356–363, Liverpool, UK, Oct. 2015.

[26] H. Chen and H. Liu, "A Remote Electrocardiogram Monitoring System with Good Swiftness and High Reliability," Computers & Electrical Engineering, vol. 53, pp. 191 – 202, 2016.

[27] K. Wac, M. S. Bargh, B. F. V. Beijnum, R. G. A. Bults, P. Pawar and A. Peddemors, "Power -and delay- Awareness of Health Telemonitoring Services: The Mobihealth System Case Study," IEEE Journal on Selected Areas in Communications, vol. 27, pp. 525–536, May 2009.

[28] X. Masip-Bruin, E. Mar´ın-Tordera, A. Alonso and J. Garcia, "Fog-to-cloud Computing (F2C): The Key Technology Enabler for Dependable e-Health Services Deployment," Proc. of the Mediterranean *Ad Hoc* Networking Workshop (Med-Hoc-Net), pp. 1–5, Vilanova i la Geltru, Spain, June 2016.

[29] C. Rotariu, V. Manta and H. Costin, "Wireless Remote Monitoring System for Patients with Cardiac Pacemakers," Proc. of the International Conference and Exposition on Electrical and Power Engineering, pp. 845–848, Iasi, Romania, Oct. 2012.

[30] G. W. Nkabinde, "Big Data Stream Computing in Healthcare Real-time Analytics," Proc. of the IEEE International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), pp. 37–42, Chengdu, China, July 2016.

[31] E. Badidi and K. Moumane, "Enhancing the Processing of Healthcare Data Streams Using Fog Computing," Proc. of the IEEE Symposium on Computers and Communications (ISCC), pp. 1113–1118, Barcelona, Spain, 2019.

[32] L. Greco, P. Ritrovato and F. Xhafa, "An Edge-stream Computing Infrastructure for Real-time Analysis of Wearable Sensors Data," Future Generation Computer Systems, vol. 93, pp. 515 – 528, 2019.

ملخص البحث:

يتضمّن تحسين أنظمة المراقبة الصحية القائمة على إنترنت الأشياء والمستخدمة في بيئات متنوعة -مثل المنازل الذّكيّة والمستشفيات الذّكيّة- التّحليل الحيّ لِسَيْل البيانات الحرج (مثل سَيْل بيانات مخططات القلب الكهربائية). والجدير بالذّكر أن إجراء هذه التطبيقات الصحية عن بُعد بواسطة السّحابة التّقليدية أمْرٌ يخرق المحدِّدات النهائية للتّطبيقات القائمة على تحليل سِيْول البيانات، الأمر الذي ليس من شأنه أنْ يؤدي الى تدهور الأداء فحسب، بل أيضاً الى نتائج غير دقيقة لعمليات التّحليل نظراً لفقدان بيانات المريض. ويمكن للحوسبة الضبابيّة أن تتم على مقربة من المريض، وهي أبرز التقنيات المفضلّة لهذا النوع من التّطبيقات.

يتم توزيع العُقَد الضّبابيّة جغرافياً وتكون فقيرةً في الموارد؛ لذا فإنّ الحاجة الى منصّةٍ لإدارة الموارد قابلة للتّوسعة وقادرة على التّعامل مع الأخطاء تُصبح أمراً واجباً. وفي الوقت الرّاهن، يتم تصميم أنظمة إدارة لمعالجة سُيول البيانات للتّعامل مع أعداد ضخمة من عُقد الموارد وتوظيفها بدلاً من العُقد الضّبابيّة فقيرة الموارد؛ ممّا يعالج النّقص في استغلال بنية النّظام. وهناك حاجة ماسّة الى تصميم منصات إدارة موارد قادرة على مواكبة طبيعة الحوسبة الضّبابيّة.

نقترح في هذه الورقة منصّة إدارة موارد مبتكرة تستند الى الحوسبة الضّبابيّة بناءً على إدارة الموارد باستخدام نظام يارن (YARN) من (Apache Hadoop2) للتّحليلات المتوافقة مع سُيول/حُزم البيانات وتمثل منصّة (S-FARM) مُجَدْوِلين من نوع (YARN)؛ أحدهما يعمل وفْق المستخدِمين والثاني يعمل وفْق الوحدات.

وتُظهر النّتائج أنّ المجَدْوِل الذي يعمل وفْق المستخدِمين يتغلّب على مشكلة نقص الموارد التي تعاني منها العُقَد الضّبابيّة، ويستغلّ بشكلٍ كامل البنية التّحتيّة الضّبابيّة، ويسمح للنّظام بالتّمدّد بأمانٍ الى مِثْلَي حجمه. إضافة الى ذلك، يتم اقتراح مُجَدْوِل YARN من طراز (S-FARM) التفاضليّ لدعم التحكّم في نمط الاستخدام وفْق المستخدِمين من حيث دقّة نتائج التّحليل وسرعة الحصول عليها. ويستخدم تطبيق (S-CVD) لتحليل مخطّطات القلْب الكهربائية للمرضى، وقد تم إجراء محاكاةٍ له في (iFogSim) للتّحقق من أداء مُجَدْوِلي (YARN) المقترحين.

ويُعدّ هذا البحث رائداً في تحسين الاستغلال الفقير للموارد للعُقد الضّبابيّة، ودعم التّحكّم في نمط الاستخدام وفْق المستخدِمين في تطبيقات تحليل البيانات الضخمة استناداً الى إنترنت الأشياء، والاستفادة من (iFogSim) في تقييم أداء منصّات إدارة الموارد في أنظمة تحليل سيول البيانات.

المجلة الأردنية للحاسوب وتكنولوجيا المعلومات (JJCIT) مجلة علمية عالمية متخصصة محكمة تنشر الأوراق البحثية الأصيلة عالية المستوى في جميع الجوانب والتقنيات المتعلقة بمجالات تكنولوجيا وهندسة الحاسوب والاتصالات وتكنولوجيا المعلومات.

تحتضن جامعة الأميرة سمية للتكنولوجيا (PSUT) المجلة الأردنية للحاسوب وتكنولوجيا المعلومات، وهي تصدر بدعم من صندوق دعم البحث العلمي في الأردن. وللباحثين الحق في قراءة كامل نصوص الأوراق البحثية المنشورة في المجلة وطباعتها وتوزيعها والبحث عنها وتنزيلها وتصويرها والوصول إليها. وتسمح المجلة بالنسخ من الأوراق المنشورة، لكن مع الإشارة إلى المصدر.

الأهداف والمجال

تهدف المجلة الأردنية للحاسوب وتكنولوجيا المعلومات (JJCIT) إلى نشر آخر التطورات في شكل أوراق بحثية أصيلة وبحوث مراجعة في جميع المجالات المتعلقة بالاتصالات وهندسة الحاسوب وتكنولوجيا المعلومات وجعلها متاحة للباحثين في شتى أرجاء العالم. وتركز المجلة على موضوعات تشمل على سبيل المثال لا الحصر: هندسة الحاسوب وشبكات الاتصالات وعلوم الحاسوب ونظم المعلومات وتكنولوجيا المعلومات وتطبيقاتها.

الفهرسة

المجلة الأردنية للحاسوب وتكنولوجيا المعلومات مفهرسة في كل من:

فريق دعم هيئة التحرير

عنوان المجلة

الموقع الإلكتروني: www.jjcit.org
البريد الإلكتروني: jjcit@psut.edu.jo
العنوان: جامعة الاميرة سمية للتكنولوجيا، شارع خليل الساكت، الجبيهة، عمان، الأردن.
صندوق بريد: 1438 عمان 11941 الأردن
هاتف: 5359949-6-962+
فاكس: 7295534-6-962+

المجلة الأردنية للحاسوب و تكنولوجيا المعلومات

آذار ٢٠٢١ المجلد ٧ العدد ١

ISSN 2415 - 1076 (Online)
ISSN 2413 - 9351 (Print)

JJCIT

www.jjcit.org jjcit@psut.edu.jo

مجلة علمية عالمية متخصصة محكمة

تصدر بدعم من صندوق دعم البحث العلمي