Princess Sumaya University for Technology
Since 1991 تأسست
PSUT
جامعة الأميرة سمية للتكنولوجيا

صندوق دعم البحث العلمي
Scientific Research Support Fund

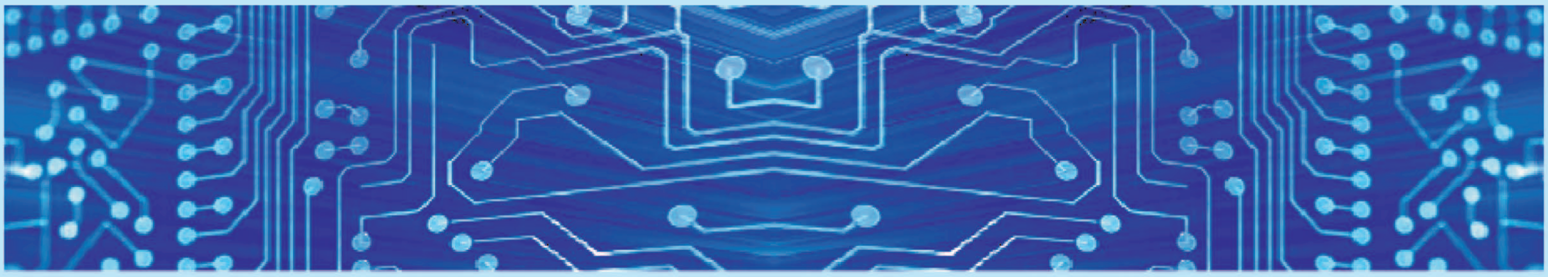
# Jordanian Journal of Computers and Information Technology

June 2021 VOLUME 07 NUMBER 02
ISSN 2415 - 1076 (Online)
ISSN 2413 - 9351 (Print)

JJCIT

| PAGES | PAPERS |
| --- | --- |


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. Researchers have the right to read, print, distribute, search, download, copy or link to the full text of articles. JJCIT permits reproduction as long as the source is acknowledged.

### 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:

ELSEVIER Scopus | DOAJ DIRECTORY OF OPEN ACCESS JOURNALS | crossref Text and Data Mining | OCLC WorldCat | BASE | ACADEMIA | Google scholar





### 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. 02, June 2021.

# ARBITER DESIGN BASED ON QUANTUM DOT CELLULAR AUTOMATA

Ziyad Altarawneh and Mutaz Al-Tarawneh

*(Received: 10-Sep.-2020, Revised: 1-Nov.-2020, Accepted: 18-Nov.-2020)*


## *ABSTRACT*

*The development of nano-scale Quantum-dot Cellular Automata (QCA) has been driven by the immense need for high-performance and energy-efficient computational systems. In this paper, 2- and 3-input QCA-based asynchronous arbiter designs are presented and investigated. A number of 2-input arbiter structures are introduced and compared with their majority-based counterpart. Simulation results show that the proposed structures outperform the majority-based arbiters in terms of number of cells, area and energy dissipation while achieving similar arbitration functionality. In addition, efficient resource utilization is obtained by configuring the proposed structures to consider the input priorities when making arbitration decision. Moreover, two 3-input arbiters are designed based on the proposed 2-input structures and proved to achieve the intended arbitration functionality. The proposed 3-input structures have surpassed their majority-based counterpart. Ultimately, the proposed arbiter designs can serve as basic building blocks in handling resource sharing in system-on-chip (SoC).*



## *KEYWORDS*

*QCA, System-on-chip, Arbiter, Resource utilization.*

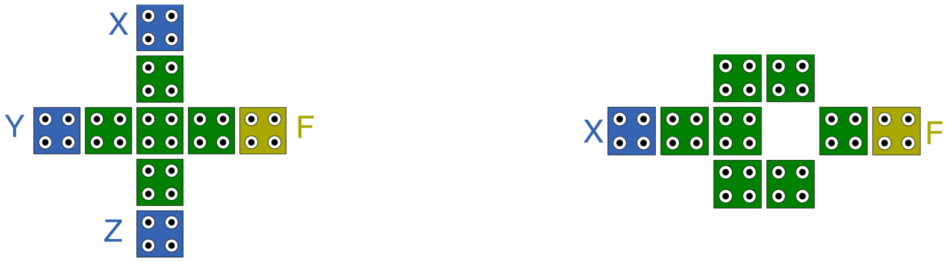

## 1. INTRODUCTION

Recently, a considerable importance has been given to System-on-Chip (SoC) designs in both research and industry, as they provide significant advantages for power-efficient, high-performance computing systems [1]-[3]. Typically, an SoC is constructed by integrating several system components, such as processors, programmable logic and on-chip memories on a single chip [4]. In such systems, the most popular clocking schemes used to coordinate data transfer between different processing elements are synchronous and asynchronous schemes [5]. In synchronous clocking scheme, the clock signal serves as a global timing reference for communicating data among different modules. It synchronizes the data processing elements to manage all latches and ensure correct timing in synchronous designs. However, the modern systems contain several communication systems (or cores) on a single chip on a larger scale. In such systems, driving multiple individual cores by a single clock becomes an increasingly complicated task, which has led to the creation of cores which are asynchronous to each other. In this regard, one possible solution is to adopt asynchronous clocking scheme, where the communication channel between different cores is considered as a shared resource between the mutually competing asynchronous subsystems. A dedicated arbitration circuit (or arbiter) is used to provide access to one or more shared resources, such as memory, difficult data processors or channels of communication. A fundamental illustration of arbitration process: suppose two transmitters that require to send data over a common network connection (channel). They cannot use the channel at the same time, so they are requesting a dedicated arbiter to approve access to the shared resource. The arbiter in turn ensures that the resource is available before access is granted to any of them.

Many research efforts have focused on the design and optimization of arbiters based on conventional CMOS technology [6]-[11]. However, with the rapid development of CMOS technology reaching the nanoscale limits, some issues, like power consumption and increased leakage current, are becoming a major impediment for further improvement in device scaling. According to the International Technology Roadmap for Semiconductors (ITRS) projections, the development of new device technologies is inevitable in future technology nodes [12]-[13]. In this regard, the nanoscale Quantum-dot Cellular Automata (QCA) technology is anticipated to offer higher density, lower power consumption and more flexible interconnection designs for future SoC. The binary values in the QCA-based elements depend on the positions of confined electrons in carefully designed quantum dots, allowing QCA-based designs to outperform CMOS-based designs in terms of switching speed, device density and power consumption

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

[14]-[15]. Typically, all computational logic gates and memory structures can be designed by assembling QCA cells in specific geometric patterns to achieve the intended functionality [16]-[17]. In QCA technology, the majority and inverter gates are the basic building blocks to synthesize different circuits. These gates are depicted in Figure 1. Other primitive logic gates, such as AND and OR logic gates, can be implemented based on the 3-input majority gate by fixing one of the inputs to either ”0” or ”1”, respectively.




(a) (b)

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

In QCA circuits, switching from one binary state to another is achieved by using external clock signals that control the inter-dot tunneling barriers, allowing the transfer of electrons between dots [18], as shown in Figure 2. In order to obtain stable logic states and information flow between adjacent cells, the inter-dot barrier of QCA cells is appropriately controlled through four different clock phases; namely, switch, hold, release and relax [18]-[19]. During the switch phase, a cell begins unpolarized and the inter-dot barrier is slowly raised and pushes the electrons into the corner dots, allowing it to attain a definitive polarity under the influence of its neighbors (which are in the hold phase). In the hold phase, barriers are held high and a cell maintains its polarity and acts as input to the neighboring cells. During the release phase, the potential barrier is slowly lowered until the cell loses its polarity. In the last phase; namely, relax, the barriers remain lowered and keep the cell in an unpolarized state, as shown in Figure 2. In addition, the QCA cells in a particular design are typically divided into sequential clocking zones whose clock signals are shifted by 90 degrees to synchronize polarization changes throughout QCA-based structures besides preventing back-propagation of information between adjacent cells.

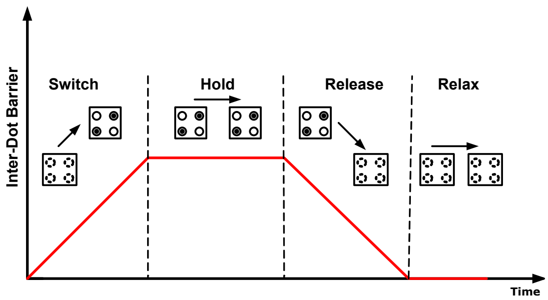



Figure 2. Inter-dot barrier in different clock phases.

In the last few years, significant research efforts have been made on the design of various SoC structures and their related interconnection networks, such as nano-router [20]-[23], serial communication network [24], error detection and correction systems [25] based on QCA technology. More recently, the authors of [26] have proposed various designs of QCA-based nano-arbiters, including the round-robin and the ping-pong arbiters, which are considered as synchronous arbiters. The proposed designs were implemented by translating the existing CMOS-based designs into equivalent majority-based QCA circuits, which ultimately could increase circuit complexity and power dissipation. However, to the best of our knowledge, no previous designs related to QCA-based asynchronous arbiters are reported in literature. This paper presents various implementations of 2- and 3-input QCA-based asynchronous arbiters. The proposed designs rely on cell interactions to achieve the intended functionally while reducing circuit complexity. Several designs were proposed based on a common baseline to overcome the metastability issue and introduce the notion of priority in the arbitration process. Finally, the proposed designs were simulated and analyzed to verify their functionality and assess their performance metrics. Ultimately, the proposed structures can serve as fundamental components in building large-scale, static-priority multi-way arbiters.

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

## 2. Proposed QCA-based Arbiter Structures

Figure 3 shows a majority-based QCA 2-input arbiter which implements the functionality characterized in Equation 1. The majority-based design consists of 36 QCA cells. As shown, the majority-based design requires the QCA cells to be divided into two different clock zones (clock zone 0 and clock zone 1) in order to achieve the proper functionality shown in Table 1.

$$G1 = MAJ(R1, NOT(R2), 0) = R1.\overline{R2}$$
$$G2 = MAJ(NOT(R1), R2,0) = \overline{R1}.R2 \quad (1)$$

where R1 and R2 represent the input requests while, G1 and G2 represent the output grants.

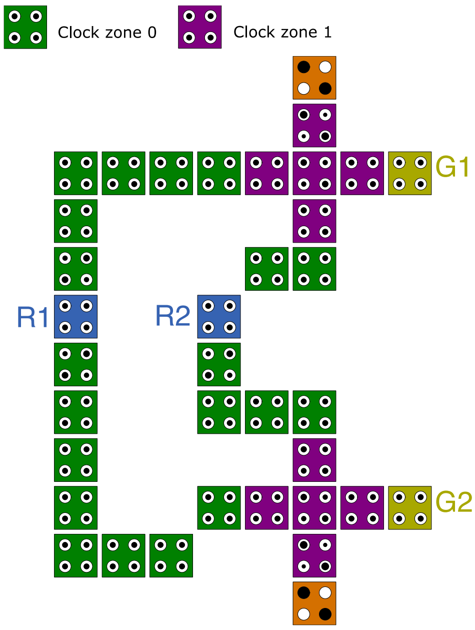



Figure 3. Majority-based 2-input arbiter.

Table 1. 2 - input majority-based arbiter truth table.

| R1 | R2 | G1 | G2 |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |

In fact, a key factor in characterizing an arbiter is its fairness in granting access to shared resources. Fairness is a measure of the ability of an arbiter to provide equal service to different requesters [9], [27]-[28]. As shown in Table 1, both requesters (R1 and R2) in the majority-based arbiter have equal chance of being granted access to a shared resource.

Figures 4(a) and 4(b) show the proposed baseline 2-input QCA-based arbiter structures. The first baseline structure (4(a)) is mainly composed of two L-shaped back-to-back QCA wires with two inputs (R1 and R2) and two outputs (G1 and G2). As shown, the design consists of 12 QCA cells. On the other hand, the other baseline design (4(b)) has less number of cells and is composed of 10 QCA cells.

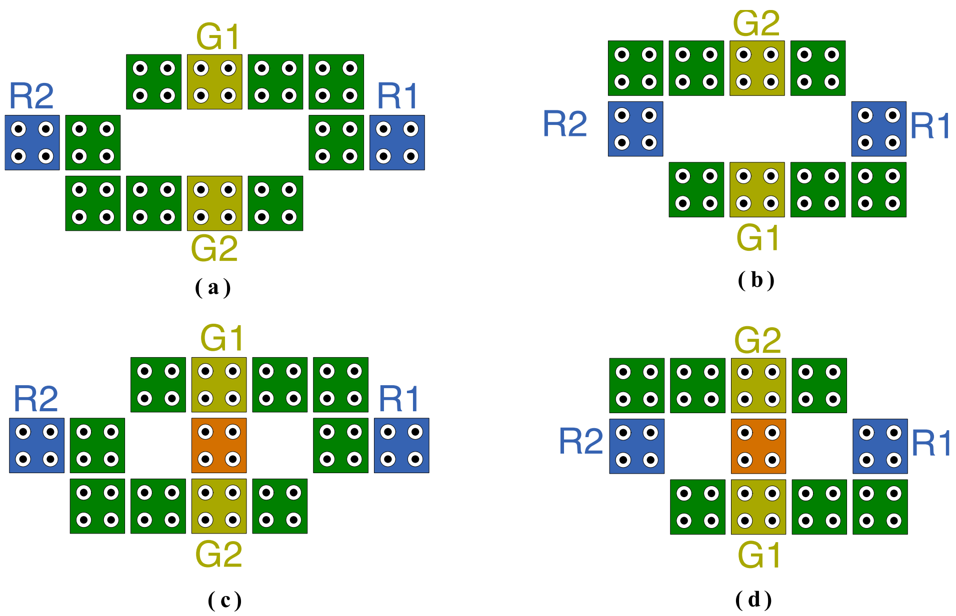



Figure 4. Proposed 2-input arbiter structures. (a) Baseline - design A, (b) Baseline-design B, (c) Modified (A) with fixed polarization (p=-0.01) and (d) Modified (B) with fixed polarization (p=-0.25).

Table 2 shows the truth table of the baseline designs. The drawback of such designs appears when the two input (request) lines are set to 0 or 1 simultaneously. As a result, a metastable state would arise, causing the output (grant) lines to potentially be activated simultaneously or grant resource access while the request line is not activated. Apparently, the occurrence of metastable states violates the concept of arbitration of resources, which dictates that only one grant line can be active at a time when a resource request is initiated. In general, the metastability phenomenon in a defect-free structure comes from unexpected polarization levels induced from the neighboring cells [29]. In the proposed baseline structures, the polarity of the output cells (G1 and G2) is determined by comparing the total electrostatic energies of two polarizations under the effect of information flow from the opposite input paths (i.e., R1 and R2). The polarization of the output cell having the smallest energy is considered to be its stable state. Due to the symmetrical cell configuration around the output cells (G1 and G2) in the proposed baseline structures, the electrostatic energies of the output cells induced from the two competing paths (from R1 and R2) are equal when both R1 and R2 have the same logic values. As a result, the output cells will be in metastable states, since the two possible polarizations, inside the output cells, will have the same energy levels. Hence, the output cells will settle in either logic 0 or 1.

Table 2. 2-input baseline arbiter truth table.

| R1 | R2 | G1 | G2 |
|---|---|---|---|
| 0 | 0 | 1/0 | 1/0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 1/0 | 1/0 |

To address this issue, an extra cell with fixed polarization is introduced in the design to resolve the metastable states, as shown in Figures 4(c) and 4(d) which present other alternative 2-input arbiter designs. The addition of the fixed polarization cell with (p = -0.01) in 4(c) and (p = -0.25) in 4(d) will mitigate the metastability issue by forcing the output cells to be in logic 0 state when both R1 and R2 have the same logic values. As shown, the number of QCA cells is reduced while resolving the metastable states. The truth table of the modified 2-input arbiters is equivalent to that of the majority-based arbiter shown in table 1. It can be observed that the fairness of the modified 2-input arbiters is similar to that of the majority-based design. However, when R1 and R2 have distinct logic levels, the effect of the fixed polarization cell on the output cells is negligible due to its low polarization level, as shown in Figure 5, which shows the simulation results of the modified 2-input arbiters.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 07, No. 02, June 2021.

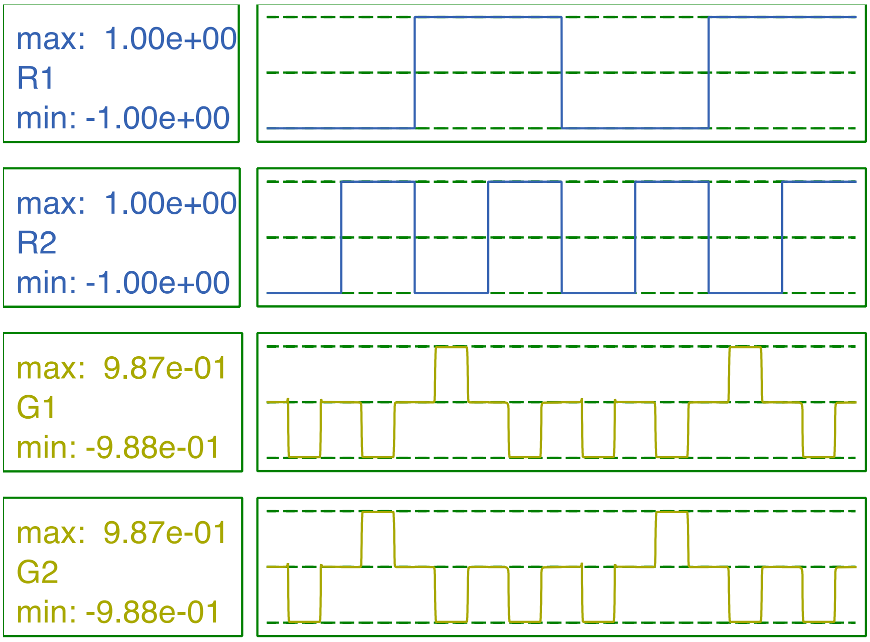
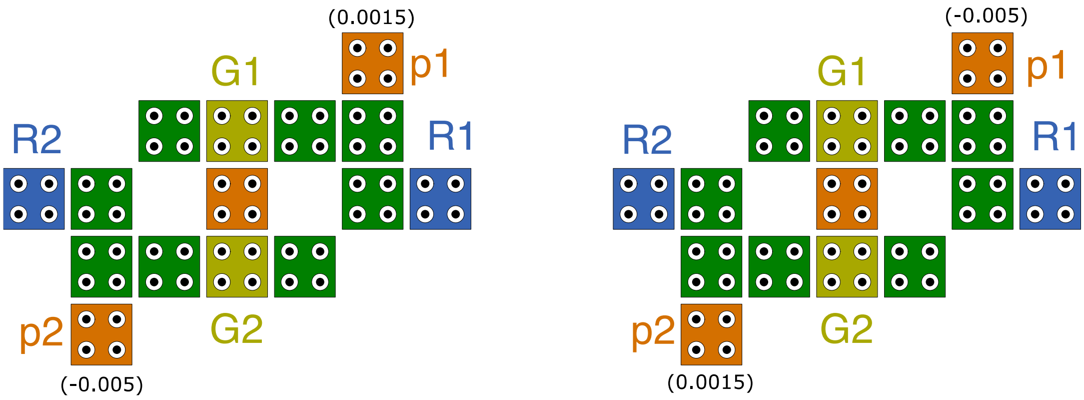



Figure 5. Simulation results of the modified 2-input arbiters.

A common theme among the modified 2-input arbiter designs is that no grants are given when both requests are issued at the same time (as shown in Figure 5) leading to a poor resource utilization. To remedy this situation, the notion of priority is introduced in order to favor one request line over the other when R1=R2=1. This can be achieved by adding 2 extra QCA cells with a fixed polarization, as shown in Figure 6. In Figure 6(a), the values of p1 and p2 were set to 0.0015 and -0.005, respectively in order to give R1 higher priority than R2. On the other hand, R2 is given higher priority than R1 by interchanging the fixed polarization values, as shown Figure 6(b). Although the introduction of priority in the arbiter design enhances resource utilization, it reduces the fairness of the proposed priority-based arbiters as compared to the majority-based and the modified 2-input arbiters. Figure 7 demonstrates the simulation results of the 2-input arbiter where input request R1 (7(a)) and input request R2 (7(b)) are given higher priority, respectively.




(a) (b)

Figure 6. Priority-based 2-input arbiter structures. (a) High priority for R1 (b) High priority for R2.

Figure 8 shows a majority-based 3-input arbiter which is composed of 6 majority gates to implement the intended arbitration functionality. This structure consists of 133 QCA cells. As shown, the QCA cells in the majority- based design are assigned to four different clocking zones (i.e., clock zones 0,1,2 and 3) in order to attain the proper functionality depicted in Equation 2 and Table 3. Figure 9 depicts the simulation results of the majority-based 3-input structure.

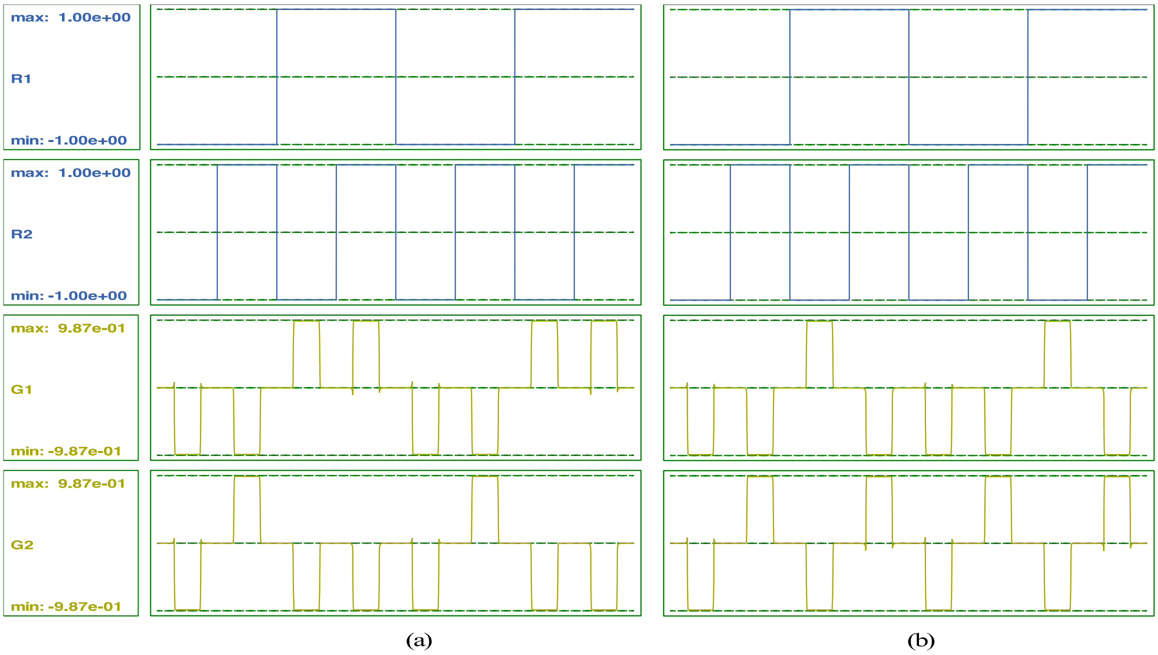



Figure 7. Simulation results of priority-based 2-input arbiter. (a) Higher priority for $R_1$ (b) Higher priority for $R_2$.

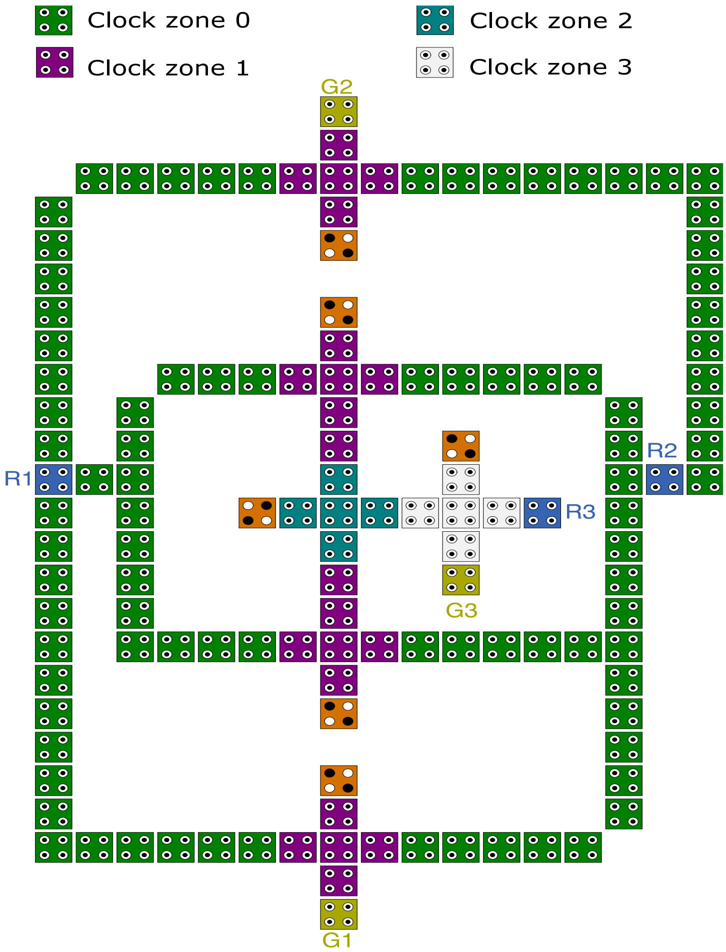



Figure 8. 3-input arbiter structure; majority-based.

$$
\begin{aligned}
G1 &= R1.\overline{R2}.\overline{R3} + R1.\overline{R2}.R3 \\
G2 &= \overline{R1}.R2.\overline{R3} + \overline{R1}.R2.R3 \\
G3 &= \overline{R1}.\overline{R2}.R3 + R1.R2.R3
\end{aligned}
\tag{2}
$$

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 07, No. 02, June 2021.

where R1, R2 and R3 represent the input requests, while G1, G2 and G3 represent the output grants.

Table 3. Truth table of the majority-based 3-input arbiter.

| R1 | R2 | R3 | G1 | G2 | G3 |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 |

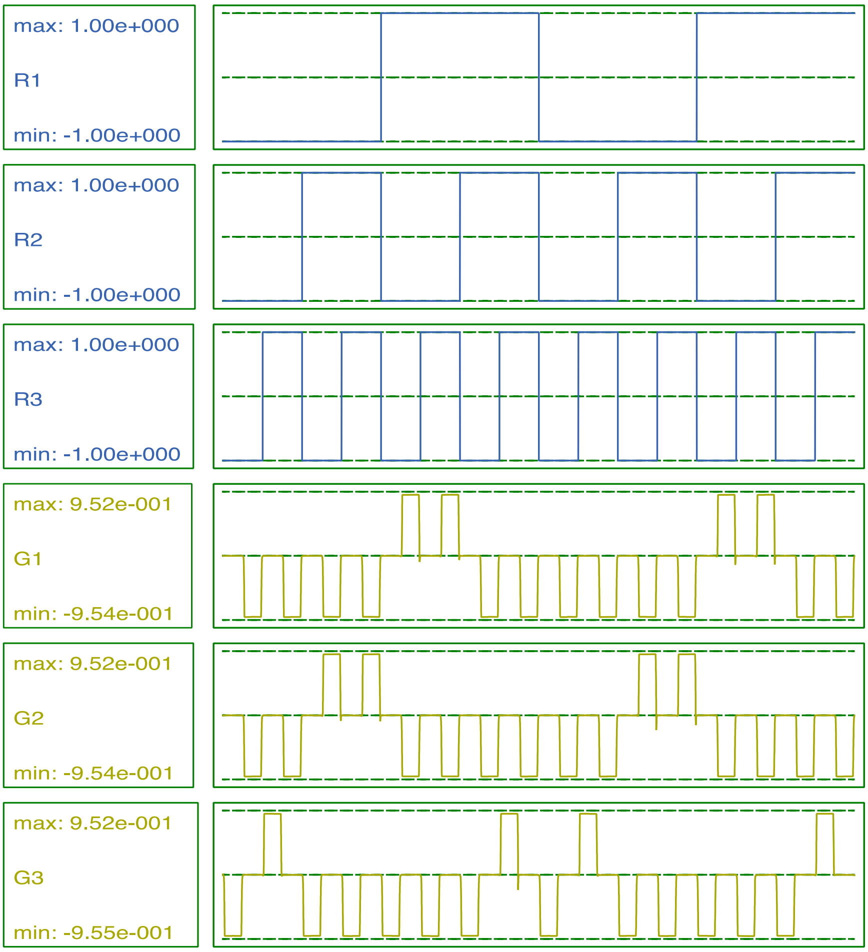



Figure 9. Simulation results of the majority-based 3-input arbiter.

Figure 10 shows an alternative design for a 3-input arbiter (Design A). This design relies on the proposed 2-input arbiters in which the arbitration functionality is achieved based on cell interactions without explicitly relying on majority gates. In Figure 10, R1 and R2 are first arbitrated using a 2-input arbiter and then combined and fed to another 2-input arbiter to which the third request line (R3) is connected.

Table 4 demonstrates the functionality of the proposed 3-input arbiter shown in Figure 10. It can be noticed that request lines R1 and R2 were given higher priority as compared to R3 except in the cases where R1 and R2 have the same logical value. For instance, when R1=R2=1, the grant line (G3) captures the value of R3 as evident in Equation 3. It can also be observed that the design shown in Figure 10 does not provide fair arbitration between requesters (i.e., R1, R2 and R3), since it assumes a static priority assignment among requesters. Figure 11 shows the simulation results of the 3-input arbiter which clearly demonstrates that the proposed design achieves mutual exclusion between request lines by ensuring that no grant lines are activated at the same time.

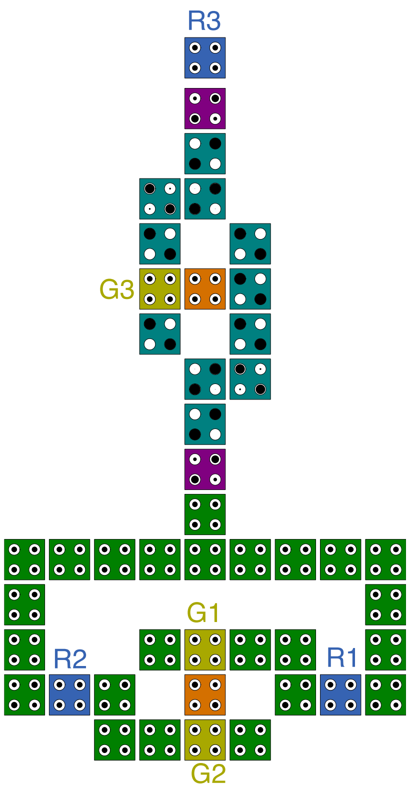



Figure 10. 3-input arbiter structures - Design A.

Table 4. Truth table of the 3-input arbiter (Design-A).

| R1 | R2 | R3 | G1 | G2 | G3 |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | (1-0) | (0-1) |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | (1-0) | 0 | (0-1) |
| 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 |

$$\begin{gathered} G1 = R1.\overline{R2}.\overline{R3} + R1.\overline{R2}.R3 \\ G2 = \overline{R1}.R2.\overline{R3} + \overline{R1}.R2.R3 \\ G3 = \overline{R1}.\overline{R2}.R3 + \overline{R1}.R2.R3 + R1.\overline{R2}.R3 + R1.R2.R3 \end{gathered} \tag{3}$$

where R1, R2 and R3 represent the input requests, while G1, G2 and G3 represent the output grants.

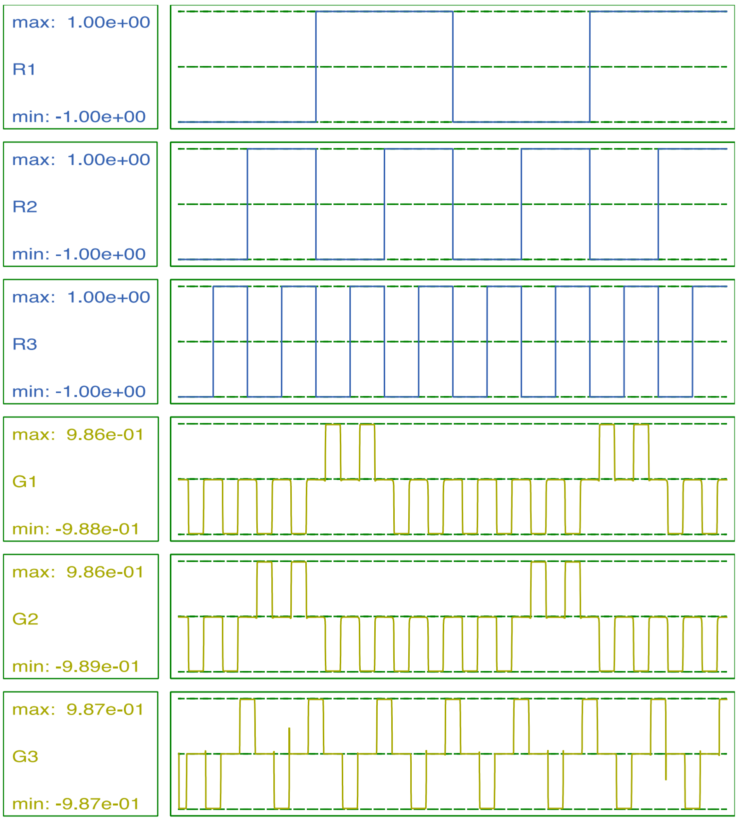



Figure 11. Simulation results of 3-input arbiters - Design A.

Figure 12 shows a different design for the 3-input arbiter (Design B) that maintains the 3-input arbitration functionality depicted in Table 3 and Equation 2.

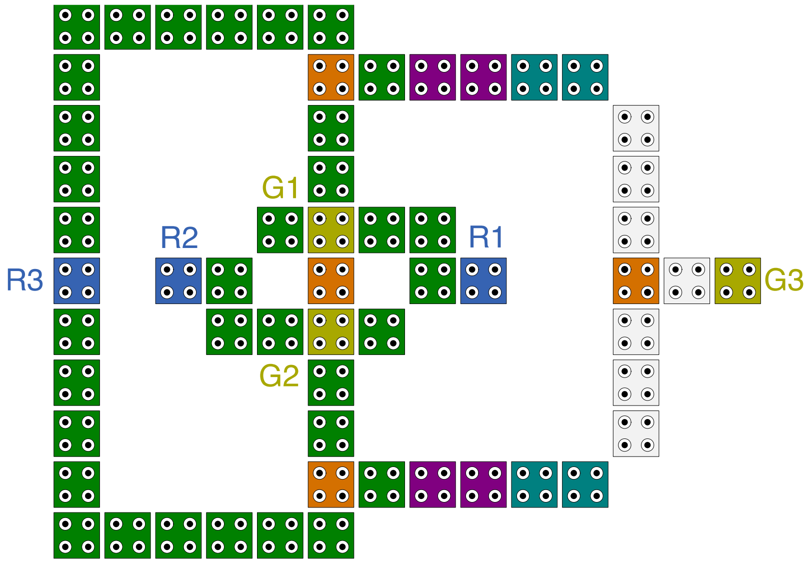



Figure 12. 3-input arbiter structures - Design B.

One of the main distinctions between the proposed 3-input arbiter designs is that the resource utilization in the design shown in Figure 10 (Design A) is better than those of the other proposed designs, as the number of resource grants is higher under the same input request combinations, as shown in Figure 13. However, the other two designs provide fair arbitration between requesters.

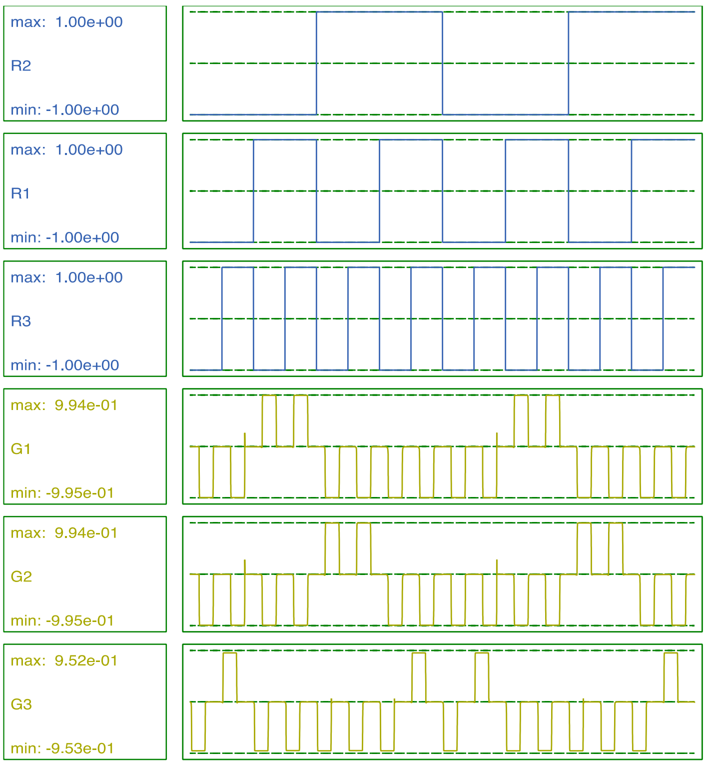



Figure 13. Simulation results of 3-input arbiters - Design B.

A more realistic implementation of the proposed 3-input structures can be obtained by adopting the two-dimensional wave (2-DW) clock distribution scheme that is based on parallel execution and processing in clocking zones [30]-[31]. The rationale behind this scheme is to overcome the limitations of the 1-D clocking scheme such as thermal fluctuation in QCA-based structures. Figures 14 and 15 show the 2-DW implementations of the proposed 3-input arbiters.

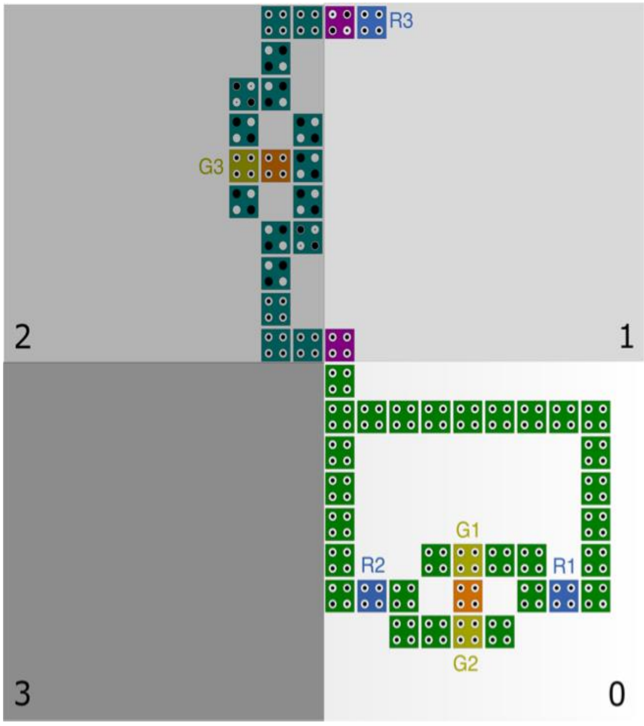



Figure 14. Realistic 2D implementation of 3-input arbiter - Design A.

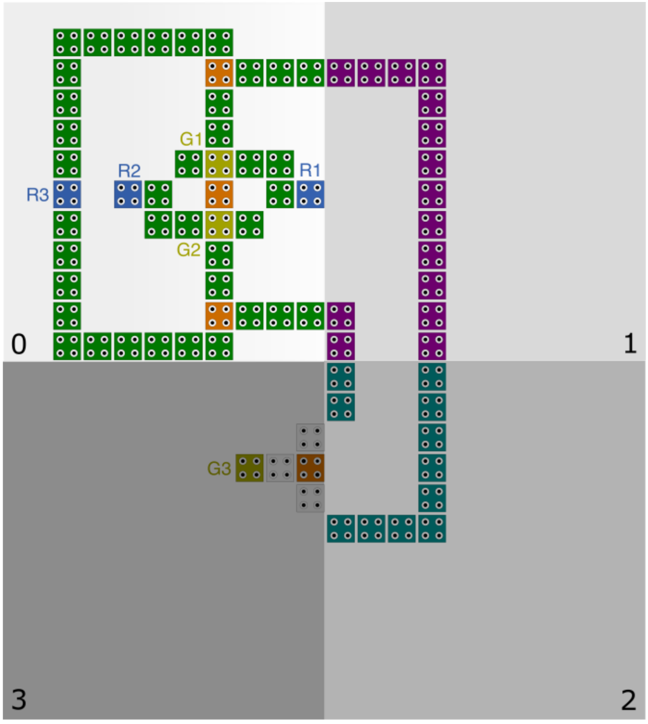



Figure 15. Realistic 2D implementation of 3-input arbiter - Design B.

## 3. Performance Analysis

The QCADesigner-2.0.3 simulation tool was used to analyze the proposed arbiter structures and assess their structural cost in terms of their number of cells and occupied area [32]. 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 5.

Table 5. Simulation parameters.

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

Figure 16 shows and compares the proposed arbiter designs in terms of the number of QCA cells required to achieve the proper functionality of the 2- and 3-input arbiters. As shown in Figure 16, all the proposed 2-input arbiters have fewer cells as compared to the majority-based 2-input structure. Moreover, the proposed 3-input arbiter structures (Design A and Design B) have lower number of cells when compared to the 3-input majority-based structure. The variation in the number of QCA cells has a noticeable consequence on the total occupied area, as shown in Figure 17. The total area is in fact the rectangular area which encapsulates all the QCA cells in a QCA-based structure including empty spaces.

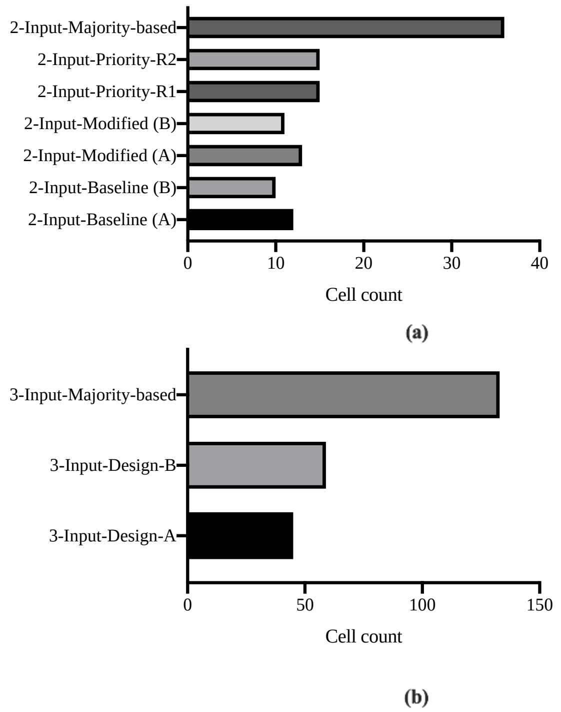



Figure 16. Number of cells comparison, (a) 2-input arbiters. (b) 3-input arbiters.

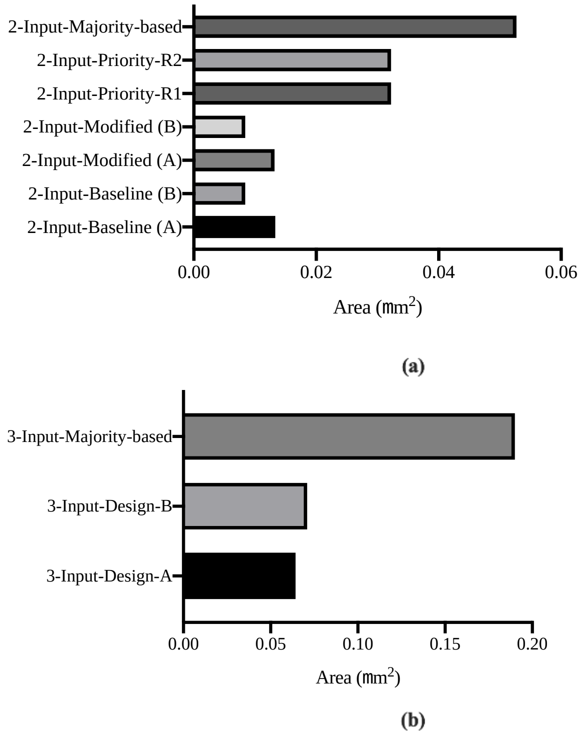



Figure 17. Occupied area comparison, (a) 2-input arbiters. (b) 3-input arbiters.

To estimate the energy dissipation of the various structures, the QCADesignerE tool has been used [33]. The QCADesignerE is a viable tool that models and estimates energy dissipation of QCA-based structures. Figures 18 and 19 illustrate the total energy dissipation and the average energy dissipation per clock cycle of the proposed arbiter structures.

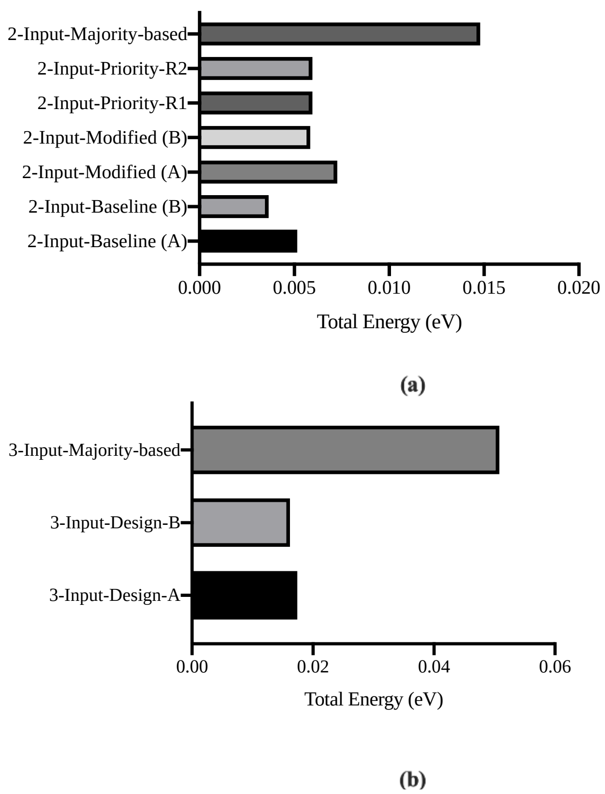



Figure 18. Total energy dissipation of the proposed structures. (a) 2-input arbiters. (b) 3-input arbiters.

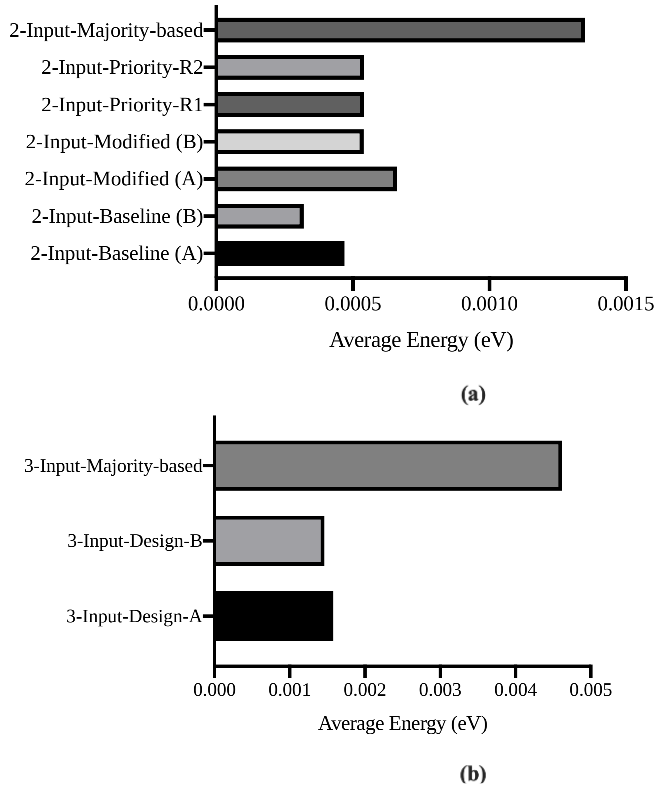



Figure 19. Average energy dissipation of the proposed structures. (a) 2-input arbiters. (b) 3-input arbiters.

For the 2-input arbiters, all of the proposed designs have less total energy dissipation and average energy per clock cycle as compared to the majority-based structure. For the 3-input arbiters, they have higher energy dissipation values as compared to the 2-input structures. In addition, the majority-based 3-input arbiter has higher energy dissipation values as compared to the other 3- input arbiters that have almost identical energy dissipation values, as shown in Figures 18 and 19. Table 6 compares the proposed 2-input arbiters against recently reported QCA-based arbiters. The comparison factors include arbiter type, cell count, area and latency. As shown, the proposed asynchronous arbiters outperform their synchronous counterparts in terms of their cell count, area and latency. The latency is defined as the number of cycles required to obtain the circuit's output after input's application. It is anticipated that the proposed 3-input arbiters would outperform synchronous designs as the 2-input structures can serve as building blocks for other large-scale arbiters.

Table 6. Comparison of the proposed structures with previous arbiter designs.

| Structure | Type | Cell count | Area ($\mu m^2$) | Latency |
|---|---|---|---|---|
| Basic round-robin arbiter [26] | synchronous | 636 | 1.14 | 5 |
| Improved round-robin [26] | | 189 | 0.27 | 2 |
| Ping pong arbiter [26] | | 147 | 0.2 | 2 |
| 2-input-modified | asynchronous | 13 | 0.0131 | 0.25 |
| 2-input-alternative | | 11 | 0.0137 | 0.25 |
| 2-input-priority-based | | 15 | 0.0322 | 0.25 |
| 2-input-majority-based | | 36 | 0.053 | 0.25 |

## 4. CONCLUSION

In this paper, different QCA-based arbiter structures were presented and thoroughly evaluated. For the 2-input arbiter structures, our proposed structures outperformed the majority-based structure in terms of the number of cells, area occupation and energy dissipation. These structures can be configured to improve resource utilization by assigning different priorities for the input requests. The proposed 2-input designs were further extended to accommodate 3-input request lines. Two 3-input arbiter designs were proposed. The proposed 3-input designs were found to have fewer cells, lower area and energy dissipation as compared to their majority-based counterpart. Ultimately, the proposed 2- and 3-input structures can serve as basic building blocks in asynchronous computational systems.

## REFERENCES

[1] A. P. Deb Nath, S. Boddupalli, S. Bhunia and S. Ray, "Resilient System-on-chip Designs with NoC Fabrics," IEEE Transactions on Information Forensics and Security, vol. 15, pp. 2808–2823, 2020.

[2] V. S. Chakravarthi, A Practical Approach to VLSI System on Chip (SoC) Design: A Comprehensive Guide, Cham: Springer International Publishing, 2020.

[3] J. Kaur and S. Reddy, "SchedCust: Design and Development of Scheduling Policy Customization Framework for ARM Based System on Chip," Procedia Computer Science, vol. 171, pp. 627 – 634, 2020.

[4] R. Ezz Eldin, M. A. El-Moursy and H. F. A. Hamed, Network on Chip Aspects, in Book: Analysis and Design of Networks-on-Chip under High Process Variation, pp. 11–44, Cham: Springer Int. Publishing, 2015.

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

[6] Y. Tamir and H.-C. Chi, "Symmetric Crossbar Arbiters for VLSI Communication Switches," IEEE Transactions on Parallel and Distributed Systems, vol. 4, no. 1, pp. 13–27, 1993.

[7] K. A. Helal, S. Attia, T. Ismail and H. Mostafa, "Priority-select Arbiter: An Efficient Round-robin Arbiter," Proc. of the 13th IEEE Int. New Circuits and Systems Conf. (NEWCAS), pp. 1–4, Grenoble, France, 2015.

[8] R. Bashizade and H. Sarbazi-Azad, "P2r2: Parallel Pseudo- Round-robin Arbiter for High Performance NoCs," Integration, vol. 50, pp. 173 – 182, 2015.

[9] L. Liu, Z. Zhu, D. Zhou and Y. Yang, "A Fair Arbitration for Network on Chip Routing with Odd Even

Turn Model," Microelectronics Journal, vol. 64, pp. 1 – 8, 2017.

[10] G. A. Subbarao and P. D. Hafliger, "Design and Comparison of Synthesizable Fair Asynchronous Arbiter," Proc. of the 18th IEEE International New Circuits and Systems Conference (NEWCAS), pp. 122–125, Montreal, QC, Canada, 2020.

[11] S. Golubcovs, A. Mokhov, A. Bystrov, D. Sokolov and A. Yakovlev, "Generalized Asynchronous Arbiter," Proc. of the 19th IEEE International Conference on Application of Concurrency to System Design (ACSD), pp. 3–12, Aachen, Germany, 2019.

[12] 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.

[13] 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.

[14] 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.

[15] H. M. H. Babu, Quantum Computing, Ser. 2053-2563, IOP Publishing, [Online], Available: http://dx.doi.org/10.1088/978-0-7503-2747-3, 2020.

[16] M. Al-Tarawneh and Z. Altarawneh, "C-element Design in Quantum Dot Cellular Automata," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 7, no. 1, pp. 51–63, March 2021.

[17] A. H. Majeed, "Quantum-dot Cellular Automata-based Superior Design of Conservative Reversible Parity Logic Circuits," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 7, no. 1, pp. 39– 50, March 2021.

[18] T. N. Sasamal, A. K. Singh and A. Mohan, Clocking Schemes for QCA, in Book: Quantum-dot Cellular Automata Based Digital Logic Circuits: A Design Perspective, pp. 139–145, Springer, Singapore, 2020.

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

[20] M. A. Tehrani, F. Safaei, M. H. Moaiyeri and K. Navi, "Design and Implementation of Multistage Interconnection Networks Using Quantum-dot Cellular Automata," Microelectronics Journal, vol. 42, no. 6, pp. 913 – 922, 2011.

[21] S. Das and D. De, "Nanocommunication Using QCA: A Data Path Selector CUM Router for Efficient Channel Utilization," Proc. of the IEEE International Conference on Radar, Communication and Computing (ICRCC), 2012, pp. 43–47, Tiruvannamalai, India, 2012.

[22] L. H. B. Sardinha, A. M. M. Costa, O. P. V. Neto, L. F. M. Vieira and M. A. M. Vieira, "Nanorouter: A Quantum-dot Cellular Automata Design," IEEE Journal on Selected Areas in Communications, vol. 31, no. 12, pp. 825–834, 2013.

[23] A. Kamaraj, Abinaya and S. Ramya, "Design of Router Using Reversible Logic in Quantum Cellular Automata," Proc. of the IEEE International Conference on Communication and Network Technologies, pp. 249–253, Sivakasi, India, 2014.

[24] D. Silva, L. Sardinha, M. Vieira, L. Vieira and O. Vilela Neto, "Robust Serial Nanocommunication with QCA," IEEE Transactions on Nanotechnology, vol. 14, no. 3, pp. 464–472, 2015.

[25] J. C. Das, D. De, S. P. Mondal, A. Ahmadian, F. Ghaemi and N. Senu, "QCA Based Error Detection Circuit for Nano Communication Network," IEEE Access, vol. 7, pp. 67 355–67 366, 2019.

[26] J. R. Monfared and A. Mousavi, "Design and Simulation of Nano-arbiters Using Quantum-dot Cellular Automata," Microprocessors and Microsystems, vol. 72, p. 102926, 2020.

[27] Y. Yang, R. Wu, L. Zhang and D. Zhou, "An Asynchronous Adaptive Priority Round-robin Arbiter Based on Four-phase Dual-rail Protocol," Chinese Journal of Electronics, vol. 24, no. 1, pp. 1–7, 2015.

[28] A. Monemi, C. Y. Ooi, M. Palesi and M. N. Marsono, "Ping-lock Round Robin Arbiter," Microelectronics Journal, vol. 63, pp. 81 – 93, 2017.

[29] M. M. Abutaleb, "QCAPUF: QCA-based Physically Unclonable Function as a Hardware Security Primitive," Semiconductor Science and Technology, vol. 33, no. 4, p. 045011, Mar. 2018.

[30] V. Vankamamidi, M. Ottavi and F. Lombardi, "Two-dimensional Schemes for Clocking/Timing of QCA Circuits," IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems, vol. 27, no.

1, pp. 34–44, 2008.

[31] A. Roohi, R. Zand, S. Angizi and R. F. DeMara, "A Parity-preserving Reversible QCA Gate with Self-checking Cascadable Resiliency," IEEE Transactions on Emerging Topics in Computing, vol. 6, no. 4, pp. 450– 459, 2018.

[32] 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.

[33] 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.

**ملخص البحث:**

تطورت أنظمة خلايا النّقط الكمّيّة القائمة على التّشغيل الذّاتي في مستوى النّانو بفعل الحاجة الشديدة الى أنظمة حوسبة عالية الأداء وفعّالة من حيث استهلاك الطاقة. في هذه الورقة، يتم تقديم واستقصاء أنظمة خلايا نّقط كمّيّة قائمة على التّشغيل الذّاتي، ثنائية المداخل وثلاثية المداخل. يتم اقتراح عدد من أنظمة التّحكيم ثنائية المداخل ومقارنتها بنظيرتها من الأنظمة القائمة على الأغلبية.

وتبيّن نتائج المحاكاة أنّ البِنى المقترحة تتفوّق على أنظمة التحكيم القائمة على الأغلبية من حيث عدد الخلايا، والمساحة، واستهلاك الطاقة في الوقت الذي تحقق فيه نتائج وظيفيّة مماثلة. بالإضافة الى ذلك، يتم تحقيق استغلال فعّال للمصادر عن طريق تشكيل البِنى المقترحة بحيث تأخذ الأولويات بعين الاعتبار عندما تتّخذ قرار التّحكيم.

علاوة على ذلك، تمّ تصميم اثنين من أنظمة التّحكيم ثلاثية المداخل استناداً الى البِنى ثنائية المداخل المقترحة، وقد أثبتت نجاعتها في تحقيق النّتائج الوظيفيّة المرجوة بفاعلية. وقد تفوّقت البِنى المقترحة ثلاثية المداخل على نظيرتها القائمة على الأغلبية.

وختاماً، يمكن لتصاميم أنظمة التّحكيم المقترحة أن تخدم كوحدات بناء أساسيّة للتّعامل مع تشارُك المصادر في الأنظمة المجمّعة بكاملها على دارة متكاملة.

# PERFORMANCE ENHANCEMENT OF MEDICAL IMAGE FUSION BASED ON DWT AND SHARPENING WIENER FILTER

Nada Jasim Habeeb

*(Received: 7-Jan.-2021, Revised: 28-Feb.-2021, Accepted: 16-Mar.-2021)*


## *ABSTRACT*

*The fusion of multimodal medical images plays an important role in data integration and improving image quality. It has a fundamental role in the accuracy of medical analysis and diagnosis. Despite the recent technological development, medical images may be exposed to blur and noise from various sources. This will affect the accuracy of the medical analysis. Therefore, de-blurring or noise removal from medical images is essential in this field. Discrete Wavelet Transform, DWT, is generally utilized in image fusion spatially in the fusion of the multimodal images. It produces a good image representation. The drawback of DWT-based image fusion is the blur presented in the fused image due to the limited directionality of wavelets. To solve this problem, Sharpening Wiener Filter and DWT-based image fusion for multimodal medical images are proposed. The proposed fusion method is evaluated using some of focus operators that were used to measure the amount of focus in the test images. The results showed that the proposed fusion gives good values of focus operators compared with the values of focus operators of image fusion techniques that are based on wavelet domain.*



## *KEYWORDS*

*Image fusion, Multimodal medical images, SWT, PCA, DT-CWT, DWT, Focus operators.*


## 1. INTRODUCTION

Medical imaging techniques are processes for obtaining images of the human body or parts of it for diagnostic, therapeutic or research purposes. The most popular types of medical imaging are Computed Tomography (CT) and Magnetic Resonance Imaging (MRI). The CT is characterized by very high clarity of the image and shows the details of the bones in an extremely accurate manner, unlike the MRI, which visualizes soft tissues with high accuracy. They are used in the diagnosis of many diseases, especially diseases of the nervous system and the brain [1], [30]. The process of image fusion of multimodal medical images can be applied to obtain a fused image which contains complementary information about these images [2]. The image resulting from the fusion of multimodal medical images plays an important role in the diagnosis and treatment process. Image quality enhancement aims to produce a better image than the original image with some standard. The image from which noise is removed is better than the image that contains noise [23]-[24]. Therefore, the fused image should contain more details, information and quality [3]. Image fusion can be performed in spatial domain (such as Intensity Hue Saturation and Principal Component Analysis) or/and in frequency domain (such as Discrete Wavelet Transform, pyramid and contour) [31]. The fused image in spatial domain has no spectral information and contains spatial distortions which may lead to wrong diagnosis. Therefore, the fusion operation in the spatial domain is not suitable for multimodal medical images. The spatial information in the medical images should be focused and clear. Multimodal medical image fusion in frequency domain achieves these requirements. Generally, the operation of image fusion in frequency domain can be summarized in the following steps: 1) transforming the input images into the wavelet coefficients. 2) Performing a fusion rule on the coefficients. 3) Inversing the transform to obtain the fused image. The fused image should not contain any distortion and has more useful information than the original images [4]. The nature of the medical images is blurry and low contrast due to the imaging system, motion of the patient or means of transmission. This will affect the image quality and limit the visibility of fine details. Blurring weakens the strength of the edges in the image, which prevents small details from appearing and causes a loss of sharpness (focus) in the density of edges. In addition, the resultant image using image fusion in wavelet domain suffers from blur due to the limited directionality of the wavelet.

N. J. Habeeb is with Middle Technical University, Iraq. Email: nadaj2013@mtu.edu.iq

To overcome the blur problem, the multimodal medical image fusion technique is proposed based on Discrete Wavelet Transform (DWT) and Sharpening Wiener Filter (SWF). The SWF contains two filters; one is the sharpening filter and the other is the Wiener filter. The multimodal medical images that contain some blur are enhanced using the SWF. The fusion process is achieved in wavelet domain using DWT. The DWT decomposes the filtered images producing low-frequency sub-bands and high-frequency sub-bands. The average rule is used to fuse the low-frequency sub-bands and the maximum rule is used to fuse the high-frequency sub-bands. The fused image is obtained by performing the inverse of the wavelet domain [21].

In the medical field, image clarity and quality play an important role in the diagnosis and treatment processes. Therefore, the importance of the proposed method is to clarify the quality of the image by improving the focus of the image, which facilitates other operations on medical images, such as segmentation and classification. The main goal of the proposed algorithm is to produce a fused image that involves high quality and more information with the advantage of focusing on the edges of the object.

The main contribution to this work is to propose a new filter called Sharpening Wiener Filter which consists of combining two filters; Wiener filter and sharpening filter. This filter works to highlight the edges and details of a blurry image. The combined image with clear details is obtained by using the sharpening Wiener filter and Discrete Wavelet Transform-based image fusion.

The rest of this paper is organized as follows: in Section 2, the related works to this work are discussed. Section 3 explains the methodology and proposed fusion method. In Section 4, the focus operators are explained. Sections 5 and 6 explain the results and conclusions, respectively.

## 2. RELATED WORKS

Generally, image fusion techniques are divided into three levels of fusion; decision-level fusion, feature-level fusion and pixel- or image- level fusion which is a low level of fusion. Although pixel-level fusion techniques have a high computational complexity compared to the other two-level techniques, it is characterized by high accuracy and no data loss occurs during it [5]. This makes pixel-level fusion methods more appropriate for application in multimodal medical image analysis. The DWT-based image fusion technique and other fusion techniques that work in the DWT domain are pixel-level fusion techniques.

The DWT is the most widely used in the image fusion process, because it produces spatial and spectral information in the fused image better than other fusion methods, such as the pyramid transform, Intensity Hue Saturation and Principal Component Analysis-based image fusion approaches. The most recent survey of medical image fusion can be found in [6]. Multimodal images are decomposed using DWT to four sub-band images (LL, LH, HL and HH, where L is Low sub-band and H is High sub-band) at each resolution level. Fusion rules are performed on these sub-bands. The most popular fusion rules are average and maximum. The final fused image is obtained by applying the inverse of DWT [7]-[8]. Figure 1 illustrates the general scheme of DWT-based image fusion.

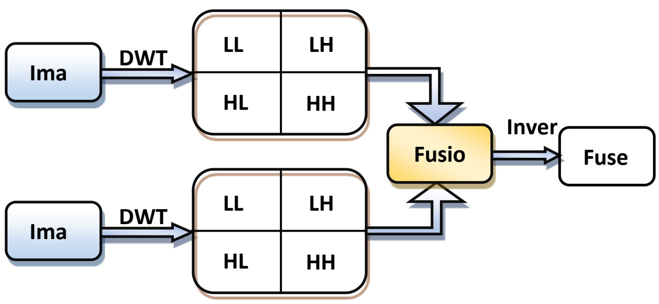



Figure 1. The general scheme of DWT-based image fusion.

The DWT-based image fusion has the main weaknesses which are the blur showing at the edges and surface areas as well as the absence of shift invariance [9]. To solve this problem, stationary wavelet transform (SWT) has been proposed which provides shift invariance. It cancels the down-sampling step which is presented in DWT and replaces up-sampling step by adding zeros between the wavelet coefficients [10]. The SWT- and DWT-based image fusion do not produce the directional information [11], [21], but Dual Tree Complex Wavelet Transform (DT-CWT) presents two characteristics; good directionality and shift invariance [12].

Principal Component Analysis (PCA) is used in many fields as a dimensionality reduction technique. Image fusion based on PCA produces an image with high spatial quality and low spectral quality. DWT-based image fusion produces an image with low spatial quality and high spectral quality. In order to take advantages of the two methods, PCA and DWT were combined to fuse the images. In PCA-DWT-based image fusion, the medical images are decomposed into sub-bands using DWT. The principal components (PC's) are evaluated and averaged. This average will constitute weights for the rule of fusion [13]. This fusion technique joins the upside of wavelet change into PCA combination as eigen estimations of multi-scale representations.

To capture an image with high visual quality and clear textures, the authors in [20] suggested using the convolution neural network for the purpose of creating a weight map. Meanwhile, the contrast pyramid is used for the purpose of analyzing the original image. The combined image is obtained based on the map of weights and on spatial frequency bands. In [22], the authors presented optimization fusion technique which contains three procedures. First, homomorphic filter is used to enhance the contrast quality of the original images. Second, DWT-based image fusion is applied to the result of the first step and third, to improve the efficiency of the fusion process, the optimization algorithm is applied using world cup and smell optimization algorithms. Finally, the inverse of the optimized DWT is applied to obtain the fused image.

The authors in [25] proposed a fusion method for multi-source medical images in the domain of the Non-subsampled Contourlet transform which converts the image into high- and low-pass images. To enhance the detailed features and preserve the original information of the medical images, the authors suggested using two fusion rules which are phase congruency and local Laplacian energy to fuse the images. The fused image is obtained by the inverse of the Non-subsampled Contourlet domain.

In [29], the authors proposed a fusion method to fuse three types of medical images which are MRI, CT and PET to obtain a single image which contains more information than the original images. The process of fusion is achieved using Simplified Pulse Coupled Neural Network in the Saturation-Hue-Value and Non-subsampled Shearlet Transformation domains using different fusion rules.

All these related works produced good results in the image fusion field for medical images; however, they didn't focus on sharpening (focusing) the blurry images.

## 3. Methodology and Proposed Fusion Method

### 3.1 Image Degradation

Figure 2 shows the model of the degradation and restoration system. Image restoration aims to reconstruct or estimate the original image which has been exposed to noise and blur. In order to obtain the enhanced image $I_{ij}$ , two conditions must be met: 1) the original image $Io_{ij}$ is uncorrelated with the noise $\eta_{ij}$ . 2) The degradation function is known. The following equations represent the degradation and restoration system [14].

$$g_{ij} = Io_{ij} \otimes h_{ij} + \eta_{ij} \tag{1}$$

$$I_{ij} = f_{ij} \cdot g_{ij} \tag{2}$$

where $h_{ij}$ and $f_{ij}$ are the degradation function and restoration filter, respectively. $g_{ij}$ is the degraded (blurred or noisy) image. The symbol '⊗' is the convolution operation and '.' is the de-convolution operation.

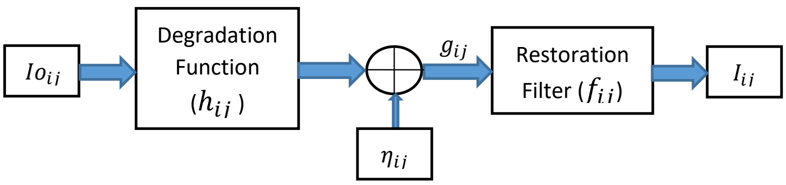



Figure 2. Degradation and restoration system.

Generally speaking, the convolution process is a multiplication operation between the pixels of the original image and the pixels of degradation kernel or filter and the noise signals are added to the results of the convolution process producing a degraded image. To estimate the original image, the de-convolution process is applied, which is a multiplication operation between the degraded image and the restoration filter such as Wiener filter [28].

### 3.2 Sharpening Wiener Filter

The Sharpening Wiener Filter (SWF) is proposed in this paper to obtain a robust sharpening (de-blurring/focusing) filter, assuming that the degradation function is known in this work. The traditional Wiener filter (WF) is a restoration filter for blurred and noisy images. By applying Fourier transformation (FT), Equation (1) can be represented in the frequency domain by:

$$G_{uv}=IO_{uv} \cdot H_{uv}+ N_{uv} \tag{3}$$

where $G_{uv}$,$IO_{uv}$, $H_{uv}$ and $N_{uv}$ are the 2D of FT for $g_{ij}$, $Io_{ij}$, $h_{ij}$ and $\eta_{ij}$ respectively and '.' is the multiplication operation. To estimate the original image in frequency domain, the following equation of the enhanced (de-blurred) image can be defined by:

$$IF_{uv} = \frac{H_{uv}^{*}}{H_{uv}^{*} \cdot H_{uv}+(\frac{1}{SNR})} \cdot G_{uv} \tag{4}$$

where $H_{uv}^{*}$ is the conjugate complex of $H_{uv}$ , SNR is the Signal to Noise Ratio. $1/SNR$ is a constant value and can be chosen between [0.0001- 0.01]. The WF is widely used and simple in implementation. The disadvantages of the traditional WF are blurring and loss of sharpness in the image edges and Spatial invariance [15]. Finally, the enhanced image is obtained in spatial domain by applying 2D inverse of Fourier Transformation (IFT) of $IF_{uv}$. To solve the problem of the WF, the sharpening filter is embedded with WF to obtain an image with good quality in terms of de-blurring and focusing. Generally, the sharpening filter is a high-pass filter which is used to preserve the edges and line details in the image. The matrix center of the sharpening filter is positive, while the surrounding values are negative [26].

Figure 3 illustrates the steps of the proposed Sharpening Wiener Filter. First, read the blurred image (original image), $Io_{ij}$ and the $3 \times 3$ sharpening filter matrix, k. Second, convert the image and the filter into the frequency domain by applying Fourier Transform (FT). The results of FT are Fourier transform coefficients of the image ($G_{uv}$) and Fourier transform coefficients of the filter ($K_{nn}$), where $u \times v$ pixels are the size of the image coefficients and $n \times n$ pixels is the size of sharpening filter coefficients. Third, apply the Wiener Filter equation on $G_{uv}$ and $K_{nn}$. The results are $IFW_{uv}$ and $\mathcal{R}_{nn}$. Fourth, the de-convolution operation (multiplication operator) is applied between $IFW_{uv}$ and $\mathcal{R}_{nn}$ and the result is $IF_{uv}$ . Finally, the inverse of the Fourier transform is achieved on $IF_{uv}$ to obtain the focused image in time domain.

The proposed Sharpening Wiener Filter is performed in order to handle the blur degradation of the image before the fusion process of multimodal medical images. This will increase the quality of the fused image through increasing the focus of the image. Figure 4 shows the block diagram of the multimodal image fusion based on DWT and SWF.

The proposed fusion algorithm was performed by a computer with the following properties: processor Intel(R), Core ™, @ 2.00 GHz. The RAM is 8.00 GB. The system type is a 64-bit OS, Windows 10. The program language is MATLAB R2017b.

The following steps explain the DWT- SWF-based image fusion:

1. Preprocessing step: first, the registration process of the two images in each pair was performed. The purpose of this step is matching the two images by transforming them into one coordinate system. In this work, it is assumed that the images in each pair in the dataset are previously registered. Second, the image resize function is applied to all images to convert their sizes into 256×256 pixels. Third, each image in the dataset is converted into a 2D image (grayscale image).
2. Applying SWF on the images.
3. The resultant images of SWF are transformed into the DWT domain. The DWT works to decompose the images into their sub-bands (Low-frequency (LF) coefficients and High-frequency (HF) coefficients).

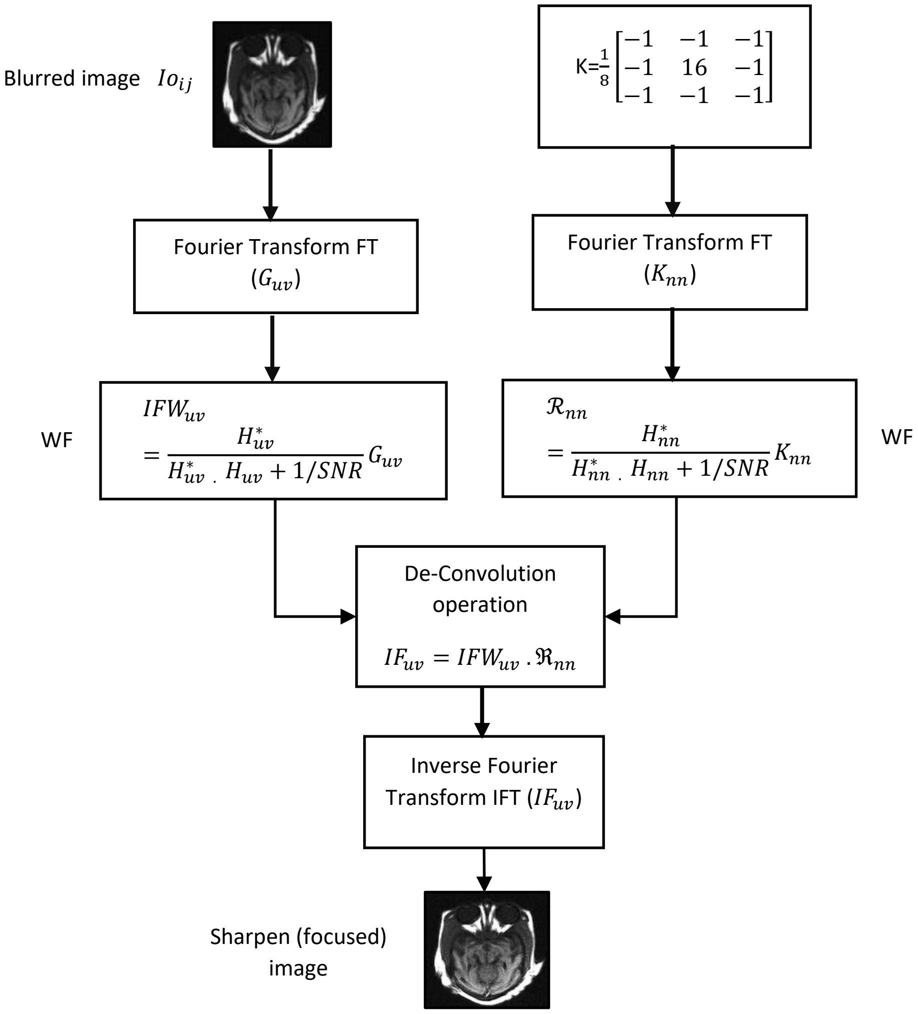



Figure 3. The proposed sharpening Wiener filter.

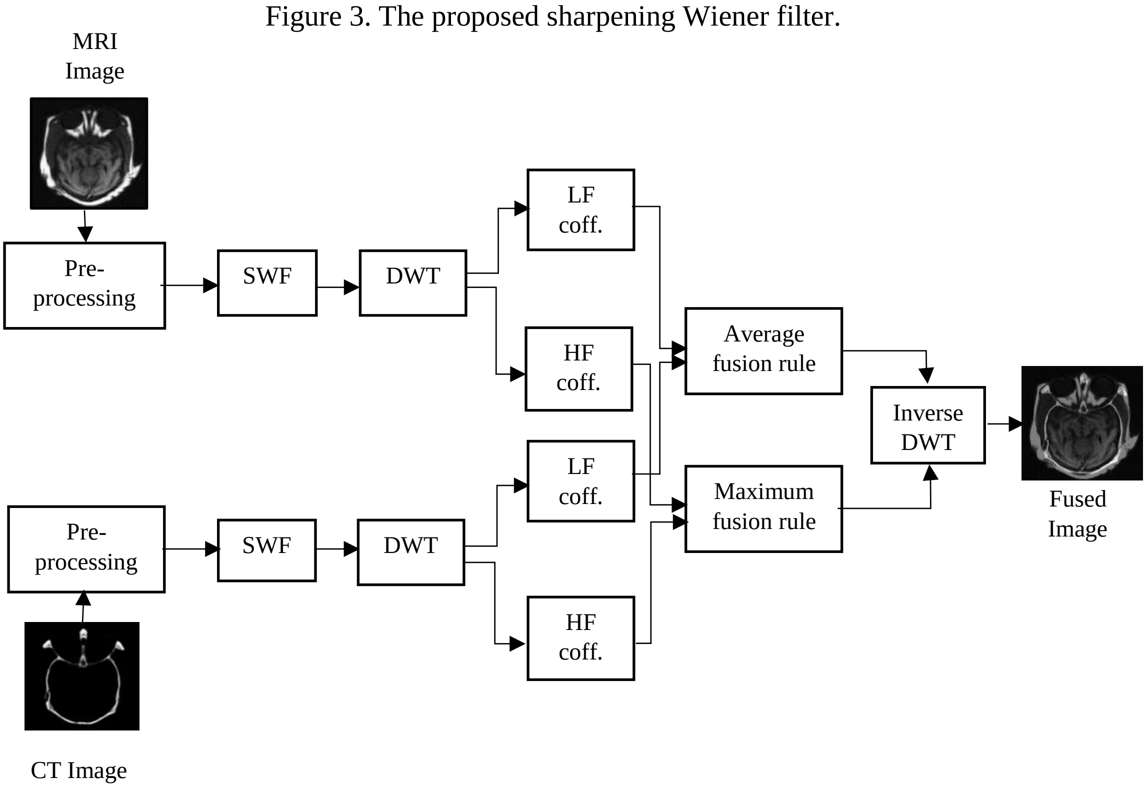



Figure 4. Block diagram of the multimodal image fusion based on DWT and SWF.

4. Average fusion rule is applied on the low bands of the coefficients of the two images. In the average fusion rule, the averages of the corresponding pixels of the LF coefficients are the pixels of the fusion result. Maximum rule is applied on the high bands of the coefficients of the two images. In the maximum fusion rule, the pixels of the fusion result are represented as the maximum values of the pixel intensity of the corresponding pixels of HF coefficients of the multimodal images.
5. Fused image is obtained by performing the inverse of the wavelet transform.

Depending on our computer system and the amount of data (two input images and one output image at each time), the proposed algorithm has an order of small-time complexity. Therefore, the algorithm is considered efficient in terms of runtime and storage space in memory.

## 4. PERFORMANCE EVALUATION METRICS

In this section, image quality metrics with no reference image, which are named as focus operators, are presented. The performance of the proposed fusion method is evaluated using the focus operators. The spatial quality and spectral quality have been measured. Table 1 shows the equations of these metrics with their descriptions. The higher the value of each one of these operators, the higher the focus in the enhanced image.

Table 1. The focus metrics with their equations and descriptions.

| Metric | Formula | Description |
|---|---|---|
| Energy of Laplacian (EoL) | $EoL_{x,y} = \sum_{(i,j)\in\Omega(x,y)} \Delta If(i,j)^2$ | EoL measures the amount of edges in the enhanced image by using Laplacian of image [16]. |
| Modified Laplacian (ML) | $ML(x,y) = \sum_{(i,j)\in\Omega(x,y)} \Delta_m If(i,j)$ <br> $\Delta_m If = abs(If * L_x) - abs(If * L_y)$ <br> $L_x = [-1\ 2 - 1] and\ L_y = L_x^T$ | ML measures the amount of edges in the enhanced image by using modified Laplacian of image [16]. |
| Variance of Laplacian (VL) | $VL_{i,j=} \sum_{(i,j)\in\Omega(x,y)} (\Delta f(i,j) - \overline{\Delta f}^2)$ <br> where $\overline{\Delta f}$ is the mean value of the image Laplacian within its neighborhood $\Omega(x,y)$. | VL measures the amount of edges in the enhanced image by using Variance of Laplacian of image [19], [16]. |
| Sum of Wavelet Coefficients (SWAV) | $\text{SWAV} = \sum_{(i,j)\in\Omega_D} abs(W_{LH1}(i,j)) + abs(W_{HL1}(i,j)) + abs(W_{HH1}(i,j))$ <br> where $\Omega_D$ is the corresponding of the neighbourhood $(\Omega(i.j))$ of the enhanced image pixel $If(i,j)$. | SWAV computes the focus score of the enhanced image using Sum of Wavelet Coefficients [17], [16]. |
| Variance of Wavelet Coefficients (VWAV) | $\text{VWAV} = \sum_{(i,j)\in\Omega_D} (W_{LH1}(i,j) - \mu_{LH1})^2 + \sum_{(i,j)\in\Omega_D} (W_{HL1}(i,j) - \mu_{HL1})^2 + \sum_{(i,j)\in\Omega_D} (W_{HH1}(i,j) - \mu_{LL1})^2$ <br> where $\Omega_D$ is the corresponding window of $\Omega$ in the DWT sub-bands and $\mu_{LH1}$, $\mu_{HL1}, \mu_{HH1}$ denote the mean value of the respective DWT sub-bands within $\Omega_D$. | VWAV computes the focus score of the enhanced image using Variance of wavelet coefficients [17], [16]. |
| Ratio of Wavelet Coefficients (RWAV) | $\text{RWAV} = \frac{C_H^2}{C_L^2}$ <br> where $C_H$ and $C_L$ are defined as follows: <br> $C_H^2 = \sum_k \sum_{(i,j)\in\Omega_D} W_{HLk}(i,j)^2 + W_{HLk}(i,j)^2 + W_{HHk}(i,j)^2$ | RWAV computes the ratio between the high-frequency coefficients and the low-frequency coefficients [16]. |

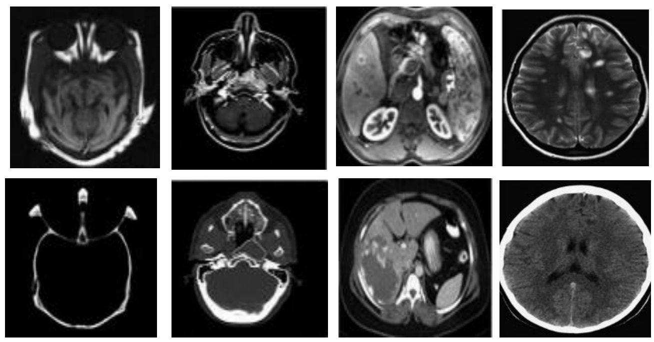
| | | |
|---|---|---|
| | $C_L^2 = \sum_k \sum_{(i,j)\in\Omega_D} W_{LLk}(i,j)^2$ | |
| Entropy (En) | $En = -\sum_{i=0}^{255} P_i log_2 P_i$ where $P_i$ is the probability of the gray level $i$ in the enhanced image. | En computes the information content in the enhanced image [18]. |
| Spatial Frequency (SF) | $SF(x,y) = \sqrt{\sum_{(i,j)\in\Omega(x,y)} (If_x(i,j)^2 + \sum_{(i,j)\in\Omega(x,y)} (If_x(i,j)^2}$ where $SF(x,y)$ is the first derivative of the two directions ($If_x(i,j)$ and $If_y(i,j)$) of the image $If$. | SF computes the focus of the image in the spatial frequency [32]. |

## 5. RESULTS AND DISCUSSION

In the experiments, three pairs of original medical images of size $256 \times 256$ pixels were used in order to verify the effectiveness of the proposed fusion algorithm. These images were collected from the site specialized in medical images of the brain (http://www.med.harvard.edu/AANLIB/home.html) [27]. Medical images generally are defocus images (Figure 5). This means that they contain some blur. Increasing blur score means more blur in the image. At first, we want to know the amount of blurring in the medical images using a blur score. The score of blurs to an image is determined by blurring the image with a low-pass filter and comparing the intensity variations between the adjacent pixels before and after adding the filter. The range of the blur score is from 0 (the best quality) to 1 (the worst quality). More details about the blur score can be found in [19]. Table 2 illustrates the amount of blur in each image from each pair. Then, the amount of blurring is measured in the fused images that resulted from the proposed fusion method and other fusion techniques (DWT, SWT, DWT-PCA and DT-DWT) as in Table 3.

MRI images

CT images



Pair-1 Pair-2 Pair-3 Pair-4

Figure 5. The four pairs of original medical images.

Table 2. The numerical values of the blur score in the original images.

| Multimodal image | Original image | Blur score |
|---|---|---|
| Pair-1 | Mri-1 | 0.5236 |
| | CT-1 | 0.3875 |
| Pair-2 | Mri-2 | 0.2800 |
| | CT-2 | 0.3282 |
| Pair-3 | Mri-3 | 0.4446 |
| | CT-3 | 0.4778 |
| Pair-4 | Mri-4 | 0.3908 |
| | CT-4 | 0.3244 |

Table 3. The numerical values of the blurring in the fused images of the proposed fusion technique compared with those of existing fusion techniques.

| Fused Image Name | DWT | SWT | DWT-PCA | DT-DWT | Proposed SWF-DWT |
|---|---|---|---|---|---|
| Pair-1 | 0.4314 | 0.4060 | 0.4235 | 0.4457 | **0.3490** |
| Pair-2 | 0.2814 | 0.2657 | 0.2241 | 0.2709 | **0.2108** |
| Pair-3 | 0.4264 | 0.4202 | 0.4483 | 0.4214 | **0.3234** |
| Pair-4 | 0.3507 | 0.3102 | 0.3669 | 0.3299 | **0.2562** |

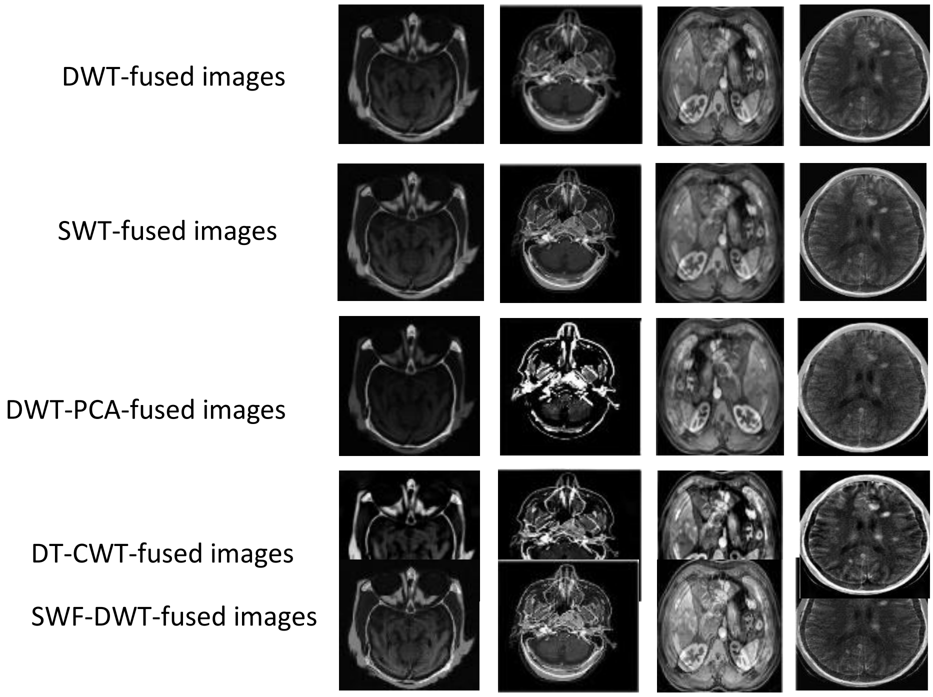



Figure 6. The resultant fused images using the proposed fusion and other fusion techniques.

Table 4. Focus metric values for the comparison of the proposed fusion technique and the existing fusion techniques for pair-1 of medical images.

| Fusion techniques / Metrics | DWT | SWT | DWT-PCA | DT-DWT | Proposed fusion SWF-DWT |
|---|---|---|---|---|---|
| EoL | 27.1301 | 26.2892 | 16.6045 | 31.0789 | **43.3688** |
| ML | 4.0694 | 3.7980 | 2.5409 | 4.6629 | **8.0715** |
| VL | 68.4137 | 55.0769 | 41.1258 | 106.0811 | **293.5792** |
| Entropy | 6.0399 | 6.0855 | 5.3133 | 5.6464 | **6.4007** |
| SWAV | 2.2862 | 2.1525 | 1.3204 | 2.3567 | **4.7265** |
| VWAV | 5.6402 | 4.3132 | 3.1391 | 6.2300 | **19.6959** |
| RWAV | 0.1018 | 0.0761 | 0.0625 | 0.0988 | **0.3637** |
| SF | 2.8216 | 2.9073 | 1.9821 | 3.3825 | **3.1702** |

It is observed from Table 2 that the original medical images naturally have some blur. From Table 3, the image fusion process using the proposed fusion method based on SWF-DWT and the other fusion techniques leads to reducing the blur level in the merged images. However, the proposed fusion method gives lower values of blur score compared with the other fusion techniques. The proposed fusion method gives visually more focus in the resultant images compared with other fused images. Tables 4, 5, 6 and 7 show comparisons of the performance of the proposed fusion method and the existing fusion techniques using the focus metrics on the medical images.

Table 5. Focus metric values for the comparison of the proposed fusion and the existing fusion techniques for pair-2 of medical images.

| Fusion techniques / Metrics | DWT | SWT | DWT-PCA | DT-DWT | Proposed fusion SWF-DWT |
|---|---|---|---|---|---|
| EoL | 45.0893 | 44.4590 | 43.8890 | 46.9290 | **61.9868** |
| ML | 8.4693 | 7.3517 | 13.0498 | 8.9971 | **18.6924** |
| VL | 273.2873 | 222.2005 | 1.1182e+03 | 359.1289 | **1.2083e+03** |
| Entropy | 5.1524 | 5.1759 | 3.8058 | 4.8093 | **5.7311** |
| SWAV | 4.7489 | 4.1874 | 8.4955 | 4.9091 | **11.4291** |
| VWAV | 23.1164 | 18.2136 | 74.4336 | 24.3379 | **80.1349** |
| RWAV | 0.3869 | 0.3088 | 1.4604 | 0.3830 | **1.3634** |
| SF | 3.7094 | 3.6936 | 3.3926 | 3.9191 | **4.3940** |

Table 6. Focus metric values for the comparison of the proposed fusion and the existing fusion techniques for pair-3 of medical images.

| Fusion techniques / Metrics | DWT | SWT | DWT-PCA | DT-DWT | Proposed fusion SWF-DWT |
|---|---|---|---|---|---|
| EoL | 50.2176 | 45.4960 | 39.3105 | 60.1983 | **86.6782** |
| ML | 6.7471 | 5.5930 | 4.7656 | 7.9568 | **16.5056** |
| VL | 101.4639 | 66.5640 | 48.1794 | 146.8988 | **565.8526** |
| Entropy | 7.2673 | 7.2797 | 7.3042 | 7.1876 | **7.6446** |
| SWAV | 3.5331 | 2.7849 | 2.0746 | 3.7073 | **9.2858** |
| VWAV | 5.8358 | 3.7028 | 2.4055 | 5.8451 | **21.9226** |
| RWAV | 0.0620 | 0.0363 | 0.0173 | 0.0626 | **0.3196** |
| SF | 5.0327 | 5.1485 | 5.0032 | 6.1892 | **6.0506** |

Table 7. Focus metric values for the comparison of the proposed fusion and the existing fusion techniques for pair-4 of medical images.

| Fusion techniques / Metrics | DWT | SWT | DWT-PCA | DT-DWT | Proposed fusion SWF-DWT |
|---|---|---|---|---|---|
| EoL | 54.8010 | 65.2433 | 45.9720 | 68.3392 | **80.3525** |
| ML | 9.9298 | 11.2523 | 7.0232 | 12.0920 | **16.9428** |
| VL | 410.4178 | 428.8384 | 188.9500 | 517.8509 | **764.2706** |
| Entropy | 6.7890 | 6.8568 | 6.7521 | 6.7358 | **6.9162** |
| SWAV | 5.9777 | 7.1859 | 4.2600 | 7.6850 | **10.0184** |
| VWAV | 26.1873 | 31.7202 | 16.0987 | 36.4523 | **44.3586** |
| RWAV | 0.3991 | 0.4445 | 0.1479 | 0.4772 | **0.7993** |
| SF | 4.5343 | 5.0945 | 4.0933 | 5.5460 | **6.1219** |

From Tables 4, 5, 6 and 7, the proposed fusion method based on Sharpening Wiener Filter and DWT gives good values of the focus operators (EoL, MF, VL, Entropy, SWAV, VWAV, RWAV and SF) on three pairs of medical images compared with the existing fusion techniques (DWT, SWT, DWT-PCA and DT-DWT) that are based on wavelet domain.

## 6. CONCLUSIONS

The quality of medical images has a great influence on the processes of analysis, diagnosis and medical treatment. Fusion of multimodal medical images takes the benefits of the characteristics of the images and gives a more complete image that helps in reaching a more accurate medical diagnosis. The image fusion based on DWT is widely used for medical images. It produces a good image representation in time and frequency domains. Due to the limited directionality of wavelets, the fused image using DWT suffers from blur. Therefore, a multimodal medical image fusion method is proposed to solve this problem using sharpening Wiener filter and DWT. The performance of the proposed fusion method was evaluated by using different focus operators. The experimental results showed that the proposed fusion method gives good values of focus operators compared with the existing DWT, SWT, DWT-PCA and DT-DWT image fusion techniques. Sharpening Wiener filter handles the problem of blurring in medical images. This is useful for clinical diagnosis. Also, the proposed fusion method can be used to increase the focus of any type of images. In future work, sharpening Wiener filter can be used with the other fusion techniques to reach more robust fusion operations.

## REFERENCES

[1] M. K. Chung, Statistical and Computational Methods in Brain Image Analysis, CRC Press, 2013.

[2] R. Singh and A. Khare, "Multiscale Medical Image Fusion in Wavelet Domain," The Scientific World Journal, vol. 5, Article ID 521034, DOI: 10.1155/2013/521034, 2013.

[3] K. Padmavathi, C. S. Asha and V. Karki Maya, "A Novel Medical Image Fusion by Combining TV-L1 Decomposed Textures Based on Adaptive Weighting Scheme," Engineering Science and Technology: An International Journal, vol. 23, no. 1, pp. 225-239, 2020.

[4] H. Ullah, B. Ullah, L. Wu, F. Y. Abdalla, G. Ren and Y. Zhao, "Multimodality Medical Images Fusion Based on Local-features Fuzzy Sets and Novel Sum-modified-Laplacian in Non-subsampled Shearlet Transform Domain," Biomedical Signal Processing and Control, vol. 57, DOI: 10.1016/j.bspc.2019.101724, 2020.

[5] S. C. Kulkarni and P. P. Rege, "Pixel Level Fusion Techniques for SAR and Optical Images: A Review," Information Fusion, vol. 59, pp. 13-29, DOI: 10.1016/j.inffus.2020.01.003, 2020.

[6] B. Huang, F. Yang, M. Yin, X. Mo and C. Zhong, "A Review of Multimodal Medical Image Fusion Techniques," Computational and Mathematical Methods in Medicine, DOI: 10.1155/2020/8279342, 2020.

[7] A. Vijan, P. Dubey and S. Jain, "Comparative Analysis of Various Image Fusion Techniques for Brain Magnetic Resonance Images," Procedia Computer Science, vol. 167, pp. 413-422, DOI: 10.1016/j.procs.2020.03.250, 2020.

[8] N. Habeeb, S. Hasson and P. D. Picton, "Multi-sensor Fusion Based on DWT, Fuzzy Histogram Equalization for Video Sequence," Int. Arab J. Inf. Technol., vol. 15, no. 5, pp. 825-830, 2018.

[9] S. Polinati and R. Dhuli, "Multimodal Medical Image Fusion Using Empirical Wavelet Decomposition and Local Energy Maxima," Optik, vol. 205, DOI: 10.1016/j.ijleo.2019.163947, 2020.

[10] P. Dwivedy, A. Potnis, S. Soofi and M. Mishra, "Comparative Study of MSVD, PCA, DCT, DTCWT, SWT and Laplacian Pyramid Based Image Fusion," Proc. of the IEEE International Conference on Recent Innovations in Signal Processing and Embedded Systems (RISE), pp. 269-273, Bhopal, India, 2017.

[11] S. Patnaik, X. S. Yang and I. K. Sethi, "Advances in Machine Learning and Computational Intelligence," Proceedings of ICMLCI 2019, Springer Nature, 2021.

[12] M. Singh, P. K. Gupta, V. Tyagi and J. Flusser, "Advances in Computing and Data Sciences," Proc. of the 3rd International Conference, ICACDS 2019, Dehradun, India, Springer Singapore, 2018.

[13] R. Vijayarajan and S. Muttan, "Discrete Wavelet Transform Based Principal Component Averaging Fusion for Medical Images," AEU-International Journal of Electronics and Communications, vol. 69, no. 6, pp. 896-902, 2015.

[14] F. Gao, C. Sun, H. Liu, J. An and S. Xu, "An Image Restoration Method Using Matrix Transform and Gaussian Mixture Model for Radio Tomographic Imaging," Wireless Communications and Mobile Computing, vol. 2017, DOI: 10.1155/2017/5703518, 2017.

[15] M. Wang, S. Zhou and W. Yan, "Blurred Image Restoration Using Knife-edge Function and Optimal Window Wiener Filtering," PLOS One, vol. 13, no. 1, DOI: 10.1371/journal.pone.0191833, 2018.

[16] Pertuz, Said, Domenec Puig and Miguel Angel Garcia, "Analysis of Focus Measure Operators for Shape-from Focus," Pattern Recognition, vol. 46, no. 5, pp. 1415-1432, 2013.

[17] G. Yang and B. Nelson, "Wavelet-based Autofocusing and Unsupervized Segmentation of Microscopic Images," Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 3, pp. 2143–2148, Las Vegas, USA 2003.

[18] S. P. Yadav and S. Yadav, "Image Fusion Using Hybrid Methods in Multimodality Medical Images," Medical & Biological Engineering & Computing, vol. 58, pp. 669-687, 2020.

[19] F. Crete, T. Dolmiere, P. Ladret and M. Nicolas, "The Blur Effect: Perception and Estimation with a New No-reference Perceptual Blur Metric," Proc. of Human Vision and Electronic Imaging XII, vol. 6492, p. 64920I, International Society for Optics and Photonics, San Jose, USA, 2007.

[20] K. Wang, M. Zheng, H. Wei, G. Qi and Y. Li, "Multi-modality Medical Image Fusion Using Convolutional Neural Network and Contrast Pyramid," Sensors, vol. 20, no. 8, Article ID: 2169, 2020.

[21] M. Diwakar, A. Tripathi, K. Joshi, A. Sharma, P. Singh and M. Memoria, "A Comparative Review: Medical Image Fusion Using SWT and DWT," Materials Today Proceedings, vol. 37, no. 2, pp. 3411-3416, 2020.

[22] L. Xu, Y. Si, S. Jiang, Y. Sun and H. Ebrahimian, "Medical Image Fusion Using a Modified Shark Smell Optimization Algorithm and Hybrid Wavelet-homomorphic Filter," Biomedical Signal Processing and Control, vol. 59, DOI: 10.1016/j.bspc.2020.101885, 2020.

[23] S. F. Abuowaida and H. Y. Chan, "Improved Deep Learning Architecture for Depth Estimation from Single Image", Jordanian Journal of Computers and Information Technology (JJCIT), vol. 06, no. 04, pp. 434-445, December 2020.

[24] A. Al-Shannaq and L. Elrefaei, "Age Estimation Using Specific Domain Transfer Learning," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 6, no. 2, pp. 122-139, June 2020.

[25] Z. Zhu, M. Zheng, G. Qi, D. Wang and Y. Xiang, "A Phase Congruency and Local Laplacian Energy Based Multi-modality Medical Image Fusion Method in NSCT Domain," IEEE Access, vol. 7, pp. 20811-20824,

2019.

[26] G. N. Yannakakis and J. Togelius, Artificial Intelligence and Games, vol. 2, ISBN 978-3-319-63519-4, New York, Springer, 2018.

[27] K. A. Johnson and J. A. Becker, "The Hole Brain," Atlas, [Online], Available: http://www.med.harvard.edu/AANLIB/home.html.

[28] Z. Wang, Z. Wang, Q. Li and H. Bilen, "Image Deconvolution with Deep Image and Kernel Priors," Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), DOI: 10.1109/ICCVW.2019.00127, Seoul, S. Korea, 2019.

[29] X. Jin, G. Chen et al. "Multimodal Sensor Medical Image Fusion Based on Non-subsampled Shearlet Transform and S-PCNNs in HSV Space," Signal Processing, vol. 153, pp. 379-395, 2018.

[30] J. Du, W. Li and H. Tan, "Three-layer Medical Image Fusion with Tensor-based Features," Information Sciences, vol. 525, pp. 93-108, 2020.

[31] X. Jin, Q. Jiang, X. Chu, X. Lang, S. Yao, K. Li and W. Zhou, "Brain Medical Image Fusion Using L2-Norm-based Features and Fuzzy-weighted Measurements in 2-D Littlewood–Paley EWT Domain," IEEE Transactions on Instrumentation and Measurement, vol. 69, no. 8, pp. 5900-5913, 2019.

[32] Z. Jing, H. Pan, Y. Li and P. Dong, "Evaluation of Focus Measures in Multi-focus Image Fusion," Non-cooperative Target Tracking, Fusion and Control, pp. 269-281, Springer, Cham, 2018.

## ملخص البحث:

يلعب دمج الصّور الطّبيّة متعددة الأنماط دوراً مهماً في تكامل البيانات وتحسين جودة الصّور، وله دور أساسي في دقّة التحليل والتشخيص الطّبّيّيْن. وعلم الرغم من التطورات التكنولوجية الحديثة، قد تتعرض الصّور الطّبيّة الى قِلّة الوضوح بفعل الزَّيْغ والتشويش من مصادر متنوعة. وهذا من شأنه أن يؤثر سلباً في دقّة التحليل الطّبي. لذا، فإنّ توضيح الصّور الطّبيّة وإزالة التشويش منها أمرٌ أساسي في هذا المجال.

يُستخدم النّقل المجرّد للمويجات (DWT) بشكلٍ عامّ في دمج الصّور مكانيّاً بالنسبة للصّور متعددة الأنماط، ويُنتج تمثيلاً جيداً للصّور. إلّا أنّ من مساوئ هذه التقنية قلّة الوضوح بسبب الزّيْغ الحاصل في الصّور المدمجَة نتيجة الاتّجاهية المحدودة للمويجات. ولحل هذه المشكلة، نقترح في هذه الورقة طريقةً لدمج الصّور الطّبيّة متعددة الأنماط ترتكز على استخدام مرشِّح فينر (Wiener) لزيادة الحِدّة جنباً الى جنبٍ مع النّقل المجرّد للمويجات.

كذلك تم تقييم الطريقة المقترحة عبر عددٍ من عوامل التّركيز المستخدمة لقياس مقدار التّركيز في الصّور المفحوصة. وقد كشفت النتائج أنّ الطريقة المقترحة تُعطي قيماً جيدة لعوامل التّركيز مقارنةً بقيم تلك العوامل التي يتم الحصول عليها من طرقٍ أخرى لدمج الصّور قائمة على حقْل المويْجات.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 07, No. 02, June 2021.

# HYBRID FEATURE SELECTION FRAMEWORK FOR SENTIMENT ANALYSIS ON LARGE CORPORA

*Kayode S. Adewole[1], Abdullateef O. Balogun[1], Muiz O. Raheem[1], Muhammed K. Jimoh[2], Rasheed G. Jimoh[1], Modinat A. Mabayoje[1], Fatima E. Usman-Hamza[1], Abimbola G. Akintola[1] and Ayisat W. Asaju-Gbolagade[1]

*(Received: 5-Jan.-2021, Revised: 22-Feb.-2021, Accepted: 17-Mar.-2021)*

## *ABSTRACT*

*Sentiment analysis has recently drawn considerable research attention in recent years owing to its applicability in determining users' opinions, sentiments and emotions from large collections of textual data. The goal of sentiment analysis centred on improving users' experience by deploying robust techniques that mine opinions and emotions from large corpora. There are several studies on sentiment analysis and opinion mining from textual information; however, the existence of domain-specific words, such as slang, abbreviations and grammatical mistakes further posed serious challenges to existing sentiment analysis methods. In this paper, we focus on the identification of an effective discriminative subset of features that can aid classification of users' opinions from large corpora. This study proposes a hybrid feature-selection framework that is based on the hybridization of filter- and wrapper-based feature selection methods. Correlation feature selection (CFS) is hybridized with Boruta and Recursive Feature Elimination (RFE) to identify the most discriminative feature subsets for sentiment analysis. Four publicly available datasets for sentiment analysis: Amazon, Yelp, IMDB and Kaggle are considered to evaluate the performance of the proposed hybrid feature selection framework. This study evaluates the performance of three classification algorithms: Support Vector Machine (SVM), Naïve Bayes and Random Forest to ascertain the superiority of the proposed approach. Experimental results across different contexts as depicted by the datasets considered in this study clearly show that CFS combined with Boruta produced promising results, especially when the features selected are passed to Random Forest classifier. Indeed, the proposed hybrid framework provides an effective way of predicting users' opinions and emotions while giving substantial consideration to predictive accuracy. The computing time of the resulting model is shorter as a result of the proposed hybrid feature selection framework.*

## *KEYWORDS*

*Sentiment analysis, Opinion mining, Hybrid feature selection, Boruta, Recursive feature elimination.*

## 1. INTRODUCTION

Nowadays, the content on the World Wide Web (WWW) has witnessed exponential growth with the advent of e-commerce, blogs, microblogs and social media websites. Availability of large textual data, usually referred to as corpora, has created a massive opportunity to mine users' opinions from such data for business analytics and for decision-making to expand businesses, products and brands and improve customers'/users' experience [1]-[2]. Although large corpora are now available for sentiment analysis to extract opinions and sentiments, processing this text data to extract such information has sparked recent attention from researchers in the last few years. Sentiment analysis is a significant stakeholder in the decision-making process and it enables individuals and groups to make sense of other people's opinions, which can be in textual form [3]. Natural Language Processing (NLP) and text classification are used in Sentiment Analysis (SA), which is a fast-growing area of computing that deals with the challenges of interpreting the text (usually human feelings or opinions) using lexicon-based approach or machine learning approach or hybrid approaches [1], [3]-[5]. Other approaches can be through knowledge-based analysis and statistical analysis or a hybrid of the two [5]. Written text can contain lots of expressions and feelings that may not be easily interpreted by the system. Unique forms of expression -called Emojis- have also been introduced into communication at various user-owned contents at large, be it personal blogs, e-commerce websites or social networks [4], [6].

A lexicon-based method is an approach that utilizes collections of sentiment words and phrases which

1. K. S. Adewole*, A. O. Balogun, M. O. Raheem, R. G. Jimoh, M. A. Mabayoje, F. E. Usman-Hamza, A. G. Akintola and A. W. Asaju-Gbolagade are with Department of Computer Science, University of Ilorin, Ilorin, Nigeria. Email: adewole.ks@unilorin.edu.ng
2. M. K. Jimoh is with Department of Education Technology, University of Ilorin, Ilorin, Nigeria. Email: jmklarularu@gmail.com

are predefined for classifying documents in the corpora to positive, negative or neutral. Lexicon-based approaches can be visualized as methods for clustering documents into clusters of positive, negative or neutral based on sentiment words and phrases already predefined. Recent development and research advancement in the domain of lexicon-based sentiment analysis have been published [7]-[8]. Conversely, machine learning approaches can be broadly classified as supervised and unsupervised learning. This method has been employed to classify or group documents based on some extracted features from documents for sentiment analysis. Supervised machine learning relies on already classified documents as training and a testing dataset which can be used to develop a predictive model for classifying emotions of new unseen documents. On the other hand, unsupervised learning groups documents into some clusters based on the similarities that exist between the documents in the corpora. The literature is vast in the application of machine learning methods for sentiment analysis [1], [6], [9]-[10]. The use of machine learning approaches for sentiment analysis relied majorly on the extraction and selection of highly discriminative features to build effective predictive models.

Feature selection involves the process of selecting a subset of features from the originally extracted features, which are considered as the best features for predicting the class of the documents in the corpora [11]. The purpose of searching for the best subset of features is to reduce the model training time and maintain the accuracy of the predictive model, if not higher than the performance with the original features. Feature selection helps in reducing the dimensionality of the feature subsets by removing irrelevant and noisy features that may hamper the performance of the predictive model for sentiment analysis. In most cases, these irrelevant and noisy features negatively affect the model generalization ability and predictive performance when used for sentiment analysis. Feature selection methods can be classified as filter-, wrapper- and embedded-based techniques [1], [11]-[14]. These three categories of feature selection pertain to the realm of classification algorithms.

At present, the size of user-owned information on the Internet is large and, on the increase, daily. The complex nature of information increase has raised the need for sentiment analysis as a tool used to understand or extract human emotions [6]. Furthermore, researchers are using machine learning algorithms to extract features from the available extensive collection of high-dimensional feature space that identifies and picks relevant features, leaving the noisy and irrelevant features behind [1]. This has made machine learning-based sentiment analysis popular in the field [15]. Some of the popularly used feature selection techniques in the literature are: Recursive feature elimination (RFE), Fast Rank-based method, Relief-F, Gain Ratio, Information Gain, Chi-Square and Boruta [1], [9]. Several supervised algorithms are also available in the literature for classification using the extracted relevant features for sentiment analysis, such as Support Vector Machine (SVM), Decision Trees, Naïve Bayes, Logistic Regression, K-Nearest Neighbour (KNN) and Multilayer Perceptron [4], [6], [9], [16].

There exist several machine learning approaches for extracting sentiment from corpora [1], [15]; however, this paper focuses specifically on corpora from different domains, which are publicly available for sentiment analysis and require a robust technique to extract discriminative features that are of high relevance and can reduce model prediction time. In this paper, a hybrid feature selection framework is proposed through a combination of filter and wrapper feature-selection methods. The resultant discriminative features were subjected to three classification algorithms to ascertain the superiority of the proposed hybrid feature selection framework. More specifically, the main contributions of this paper are briefly highlighted as follows:

- Proposing a new hybrid filter- and wrapper-based feature selection framework for sentiment analysis on large corpora.
- Considering different contexts to ascertain the applicability of the proposed hybrid feature selection framework across various domains, including movie review, public opinion and product review.
- Analyzing the performance of the extracted features by conducting extensive comparative evaluation based on the three machine learning classifiers considered in this study.

The remaining sections of this paper are structured as follows. Section 2 gives related studies on sentiment analysis and machine learning techniques that have been deployed for sentiment analysis. Section 3 explains the proposed hybrid feature selection framework for sentiment analysis, the description of the corpora, the evaluation metrics and the feature selection techniques as well as the classification algorithms used in this study. Section 4 discusses the results of the various experiments conducted and comparatively analyzes the two proposed hybrid feature selection approaches. Finally, Section 5 concludes the paper and gives future research directions.

## 2. RELATED WORKS

Sentiment analysis and opinion mining are two inter-related concepts that deal with the computational study of people's reaction, attitudes, opinions, sentiments and emotions towards a topic, entity, aspect …etc. expressed in texts [17]-[18]. A vital step of sentiment analysis is selecting an appropriate approach in classifying the opinions. The classification methods of opinion mining can be categorized into two groups; namely: machine, learning approach and lexicon-based approach [19]. The machine learning techniques for opinion mining can be broadly categorized into three aspects, which are; supervised, semi-supervised and unsupervised learning. Several studies have employed unsupervised learning using probabilistic classification, stochastic classification or a combination of both [20]. Probabilistic classification is a famous classification approach in opinion mining; it involves using mathematical expressions to classify the sentiments about a given text. Since the techniques are obtained from probabilistic models, they provide a logical way for classification in a complex domain, such as the field of NLP [20]. Thus, it also has an effective application in opinion mining. Some of the prominent methods in the field of opinion mining belonging to this classification include Naïve Bayes, Bayesian Network and Maximum Entropy. In other situations, which might be the nature of the problem at hand, probabilistic classifiers might be ineffective. Thus, the other option for solving the classification problem is by using stochastic classifiers (also called non-probabilistic classifiers). Some widely used non-probabilistic classifiers in sentiment analysis include Neural Network, Support Vector Machine, K-Nearest Neighbour (KNN), Rule-based methods as well as Decision Tree.

The Bag of Words (BoW) is popularly used to depict sentiment analysis in recent research due to its ability to make word objectivity independent and important as well as giving less importance to subjectivity and text arrangement [15], [21]. In their work, a novel framework was proposed which minimized the size of the feature vectors through semantic clustering and data sparseness for sentiment analysis. Due to the challenges of feature extraction in sentiment analysis, Ansari et al. [9] utilized a hybrid filter and wrapper technique for feature vector selection in order to minimize the vector size and boost the classification accuracy. The researchers used the fast rank-based method on the initial feature set and the result is passed through recursive feature elimination RFE and the evolutionary method of binary particle swarm optimization to obtain the ultimate feature subset.

Hassonah et al. [1] proposed the combination of ReliefF and Multi-Verse Optimizer (MVO) feature extraction algorithms to enhance sentiment analysis. Support Vector Machine (SVM) was used for the classification of the model based on positive, neutral and negative emotions. Comparing the result of the feature extraction techniques and SVM in terms of accuracy against other works showed an improved result with a decrease in the feature numbers by approximately 97% from the original set and datasets yielded an improved result as well [1]. Do et al. [6] carried out a review on deep learning approaches for sentiment analysis with a focus on aspect extraction and sentiment classification. The researchers compared major deep learning methods for aspect level of sentiment analysis. It was concluded that both aspect level sentiment analysis and deep learning need more research work focus as most researchers work on extraction or classification alone, whereas combining both will give a better result. Arulmurugan, et al. [4] proposed a cloud-based technique to integrate emotions like calmness, excitement, stress, confusion and frustration in the construction of an intelligent system for sentiment analysis. The system increased the sentence level sentiment through the use of support vector machine, Naïve Bayes and Neural Network algorithms for classification of specific features of the dataset and a modified K-means clustering method for dataset outliers.

A Japanese large word corpus was developed with five billion words derived from Japanese blogs due to the lack of such in existence anywhere. A two-dimensional model of annotation was used to get information on sentence valence used in sentiment analysis. The evaluation was done on the annotations in more than one way and the large corpus can be used for object ontology and significance of action [22]. Word embedding method was used to improve the accuracy of sentiment analysis for pre-trained words. The technique made use of Part-of-Speech (POS) tagging techniques, lexicon-based approaches, word position algorithm and Word2Vec/GloVe methods. Several deep learning algorithms were used to check the accuracy of the proposed method [3].

Hasan et al. [5] developed a framework for sentiment analysis and classification of hashtag (#) messages representing the opinions of political interest on Twitter. The research compared three sentiment lexicons (W-WSD, SentiWordNet and TextBlob). The polarity and subjectivity were derived using several libraries while Naïve Bayes and support vector machine algorithms were applied to the training set in WEKA to derive the classification model. The best result was obtained from the analysis of tweets with W-WSD. In a research conducted by Arulmurugan et al. [4], Binary Cuckoo Search (based on the characteristics of Cuckoo bird) was used for feature selection of online text content for sentiment analysis and supervised algorithm (support vector machine, decision tree, Naïve Bayes, k-nearest neighbour and multilayer perception) for its classification. The result showed an enhanced accuracy in the sentiment classification due to application on BCS on the dataset for optimized feature selection. Cambria et al. [23] proposed ensemble application of symbolic and subsymbolic Artificial Intelligence (AI) for sentiment analysis. The study integrates both top-down and bottom-up learning using an ensemble of symbolic and subsymbolic AI tools. This was then applied to the problem of polarity identification from text data. A common-sense based Application Programming Interface (API) for concept-level sentiment analysis has been proposed in the literature [24].

Jeyapriya and Selvi [25] employed Naïve Bayes in phrase-level opinion mining in customer product review. The datasets were obtained from Amazon, Eponions and Chet. The model was evaluated based on aspect extraction and sentiment orientation. Also, Tripathy et al. [26] compared the performances of SVM and Naïve Bayes for movie review datasets that were obtained from IMDB. The study was able to show that SVM outperforms Naïve Bayes classifier in predicting the sentiment of a movie review. Alfaro et al. [27] compared the results of SVM and kNN based on content classification and opinion mining on weblog comments. Based on the experiments conducted in the study, it was shown that SVM outperforms kNN in terms of accuracy. Hussain and Cambria [28] employed a semi-supervised learning approach for big social data analytics. The study proposed an affective common-sense reasoning architecture based on random projections and SVM which showed a noteworthy improvement in emotion recognition accuracy as well as in polarity detection. Also, Claypo and Jaiyen [29] utilized an unsupervised machine learning approach for a restaurant review dataset which was obtained from TripAdvisor. The study applied an MRF feature selection technique and KMeans for clustering the reviews into positive and negative. In addition, Al-Agha and Abu-Dahrooj [30] conducted a study by taking data from Twitter to analyze world public sentiment about the Palestinian- Israeli crisis. Their study proposed a multi-level research model that utilizes several variables at the group and individual levels, using statistical methods to carry out a systematic public sentiment. Similarly, Kumar et al. [31] used sale tweets to analyze consumers' thoughts about electronic goods. The researchers discovered that the logistic regression technique has a promising result for all datasets employed. Nahar et al. [32] presented a lexicon-based approach to identify the sentiments of posts and comments on Jordanian telecommunication companies on Facebook. The researchers were able to formulate an Arabic Sentiment Lexicon on which they applied three classification algorithms (SVM, kNN and Naïve Bayes).

From the literature, it is evident that many machine learning models for sentiment analysis have been proposed. However, the utilization of a hybrid machine learning algorithm for feature selection on large corpora is still an open research issue. Thus, the goal of this study is to fill this research gap by proposing a robust multilayer hybrid feature selection framework for sentiment analysis across different domains, such as product reviews, movie reviews, public opinions … and so on.

## 3. METHODOLOGY

This section details the structure and flow of the techniques used in hybridizing the algorithms for sentiment analysis modeling. First, we begin with the process of data collection and then explain the preparation processes as well as the mechanisms applied to the data gathered to make it accessible and useful for machine learning modeling. Next, we offer a comprehensive overview of the datasets with the number of their attributes, instances and classes. The subsequent subsections explain the selected feature selection algorithms and the classification techniques employed.

## 3.1 Proposed Framework

Figure 1 shows the proposed hybrid feature selection framework, which comprises different stages to achieve the overall aim of this study. Four publicly available datasets were collected to evaluate the performance of the proposed hybrid framework. The subsequent sections provide detailed explanations of the various stages involved in the proposed hybrid framework.

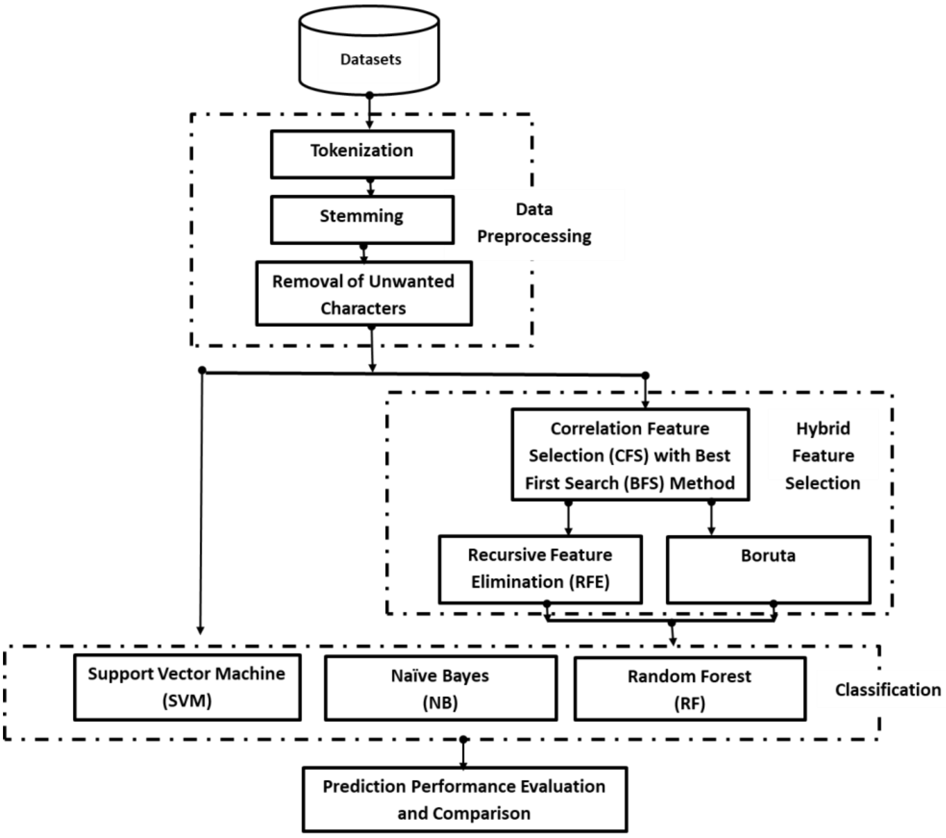



Figure 1. The proposed hybrid framework for sentiment analysis.

## 3.2 Data Collection and Description

In evaluating the proposed hybrid feature selection framework for sentiment analysis on textual data, several open-source datasets were utilized. These include Amazon, Yelp, IMDB and Kaggle datasets which are publicly available for research purposes. The brief descriptions of these datasets are as explained in Table 1. The corpora are written in English language.

Table 1. Corpora description.

| Dataset | No of instances | No of attributes |
|---|---|---|
| **Amazon** | 1000 | 620 |
| **Yelp** | 1000 | 691 |
| **IMDB** | 1000 | 961 |
| **Kaggle** | 13871 | 1218 |

The corpora used in this study comprise three datasets of customer reviews (Amazon, Yelp and IMDB datasets) on several products and services and the United States 2016 Presidential debate, which is a Kaggle dataset. The first dataset, Amazon, is an open-source corpus that is publicly available at “https://registry.opendata.aws/amazon-reviews/”. The corpus has 1000 instances of customer reviews on products purchased on Amazon store. Similarly, Yelp is also a corpus consisting of 1000 instances of customers’ reviews on products. This corpus is readily available at https://www.yelp.com/dataset. IMDB is a corpus of customers’ reviews on movies which is also publicly accessible at https://www.kaggle.com/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews. The Kaggle dataset on the other hand is the First US GOP debate which is openly accessible at https://www.kaggle.com/crowdflower/first-gop-debate-twitter-sentiment. This corpus entails tweets about the presidential debate in determining whether the contributor's opinion is positive, negative or neutral. The corpus has 13,871 tweets which were analyzed based on relevancy, the candidate that was mentioned, the subject as well as the sentiment that was given to the tweet.

## 3.3 Data Pre-processing and Preparation

The corpora obtained undergo a series of pre-processing stages to prepare and transform them into a more consumable form that can be used by the algorithms. These phases include:

i. Tokenization: This is the process of producing several representations of information-enriched texts which can lead to a better classification outcome. It operates by transforming the extracted documents and texts into more practical and machine-consumable forms of texts, such as words, phrases, sentences, …etc. This is the first process of feature extraction where texts are converted into tokens before transforming into vectors.

ii. Stemming: Next, with the "tm_map" feature provided *via* the "tm" package in R, all the derivative words were transformed back to their root form. Stemming is highly valuable as it assists in the recognition of related terms as well as their reduction in data dimensionality.

iii. Removal of unwanted characters: Numbers, unwanted spaces, special characters, …etc. are all eliminated from the word list, as they are unnecessary and meaningless. These do not contribute to the sentiment, as they degrade the performance of the machine learning models.

iv. Feature extraction: Finally, the conversion to the document term frequency matrix was done. Document term frequency is a statistical matrix that displays the frequency of words contained in a record set. In this matrix, each row denotes a document, each column represents one term (word) and each entry value has the number of appearances of that term in that document.

## 3.4 Hybrid Feature Selection

The major target of attribute selection in sentiment analysis is to discover the best set of features that allows building useful models. The major goal of most applications is to develop a well-performing prediction model. Another, sometimes more important, is to identify those variables that enhance this good prediction; i.e., reducing the large set of measured variables to the ones that contain more information rather than noise [14], [33]. Thus, several feature selection techniques have been proposed using different principles and approaches to report the set of truly relevant variables. In this study, a hybrid feature selection is employed. First is the Correlative Feature Selection (CFS), a filter-based attribute selection technique that is based on Best First Search (BFS) technique. The result of this feature selection process is then passed to either Boruta or RFE which are wrapper-based attribute reduction methods. The final feature subsets are passed to the classification algorithms to evaluate the proposed hybrid feature selection framework.

### 3.4.1 Correlative Feature Selection

Correlation Feature Selection (CFS) is a typical type of filter feature subset selection methods. Filter feature subset selection evaluates, ranks and selects features based on some properties. CFS generates a feature subset based on the search method that is employed to select features that possess good prediction capacities. The search method traverses the feature space to generate a subset of the features with high predictive potentials. According to [34], CFS considers the existence of better predictive performance when combining features. The performance of CFS varies considerably based on the search method employed. Therefore, we carefully selected one of the best search methods as reported in the literature for the CFS stage, which is based on BFS approach [35]. The best first search strategy works by first emptying a set of attributes, beginning with the complete set of attributes or beginning a quest in any given direction and going backwards (by considering all possible single attribute additions and deletions at a given point).

### 3.4.2 Boruta Algorithm

The Boruta algorithm, named after a Slav god of the forest, was created to identify all important attributes in a classification framework [36]. It is a wrapper technique that is built around the Random

Forest classifier, with the main idea of comparing the importance of the real predictor variables (known as real-data/original data) with those of random (also called shadow) variables using statistical testing and some runs of Random Forest. In each run, the set of predictor variables is folded by adding a copy of each variable. The values of those shadow variables are generated by permuting the original values across observations and therefore destroying the relationship with the outcome. A random forest is trained on the extended set of data (called extended data) and the variables' importance values are collected. For each real variable, a statistical test is conducted to compare its importance with the maximum value of all the shadow variables. Variables with significantly larger or smaller importance values are declared as important or unimportant, respectively. All irrelevant features and shadow attributes are eliminated and then the previous steps are repeated until all the features are classified or a pre-specified number of runs have been performed.

The Boruta algorithm is described as follows [36]:

```
Algorithm 1. Boruta Algorithm
Input: realDate – The dataset; RFruns – specified number of random forests runs
Output: finalSet: It has set of important and unimportant features
confirmedSet = NULL
rejectedSet = NULL
for each RFruns do
    originalPredictors = realData(predictors)
    shadowAttributes = permute(originalPredictors)
    extendedPredictors = cbind(originalPredictors, realData(decisions))
    zScore = randomForest(extendedData)
    MZSA = max(zScoreSet(shadowAttributes))
    for each a which belongs to originalPredictors do
        if zScoreSet(a) >  MZSA then
        hit(a)
        endif
      endfor
endfor
for each a an element of originalPredictors
    significance(a) = twoSizedEqualityTest(a)
    if(significance(a)) >> MZSA) then
      confirmedSet = finalSet U a
endfor
return finalSet = rejectedSet U confirmedSet
```

```
Algorithm 2. Recursive Feature Elimination
Tune/Train the model on the training set using all predictors
Calculate model performance
Calculate variable importance or rankings
for each subset size S_i, i =1 … S do
     Keep the S_i most important variables
     Tune/Train the model on the training set using S_i predictors
     Calculate model performance
endfor
Calculate the performance profile over the S_i
Determine the appropriate number of predictors
Use the model corresponding to the optimal S_i
```

### 3.4.3 Recursive Feature Elimination (RFE)

Recursive Feature Elimination (RFE) is a wrapper-based feature selection technique. It is a brute-force approach to attribute selection that operates by searching for a subset of features. It begins with all attributes in the training dataset and successfully eliminates the weakest features until the desired number of features remains. In RFE, features are classified according to the model's attributes. RFE aims to remove co-linearity and dependencies that may exist in a model by recursively deleting a limited number of features per cycle.

## 3.5 Classification Algorithms

This study investigates the performance of three machine learning algorithms to ascertain the applicability of the proposed hybrid feature selection framework for sentiment analysis across different domains. The classification algorithms were selected due to their wide range of acceptability for similar classification tasks in the literature [1].

### 3.5.1 Support Vector Machine

Support Vector Machine (SVM) is a well-known supervised machine learning technique used for classification. It is an efficient machine learning technique based on the principle of structural risk minimization; it is capable of solving the small-sample and nonlinear classification problems. The basic principle of SVM is that it searches for optimal separating hyperplane so that the classification problem becomes linearly separable. Given a set of labelled data where there are two possible label classes, SVM builds a model that maps the data as points in a space so that the two separate classes of labelled data are divided by a clear gap as wide as possible. Thereafter, the model is used in mapping unknown data into the previously mentioned space and predicting the label class of the unknown data based on which side of the gap it is mapped.

### 3.5.2 Random Forest

Random Forest is a prominent machine learning classification algorithm that has a collection of tree predictors. Each of these predictors is used for classifying an unknown instance. The resulting classification for the unknown instance is selected based on the majority result of the trees' predictions. Random Forest is a class of decision tree algorithms based on an ensemble approach [37]. It creates an ensemble of classifiers by creating several decision trees using a random feature selection and bagging approach at the training stage. This decision tree yields two types of nodes: the leaf node labelled as a class and the interior node associated with an attribute. A different subset of training data is selected with a replacement in training each tree. Entropy is applied to compute the information gain contributed by each feature. Let $D$ represent the corpus with the labelled instances and $C$ the class such that $C = \{C_1, C_2, C_3, \ldots, C_j\}$, where $j$ is the number of classes considered. In this paper, the value of $j$ is set to 2 or 3 depending on the specific corpus used, as earlier discussed. Formally, the information needed to identify the class of an instance in the corpus $D$ is denoted as $Info(D) = Entropy(P)$, where $P$ is the class probability distribution, such that:

$$P = \left\{ \frac{|C_1|}{|D|}, \frac{|C_2|}{|D|}, \frac{|C_3|}{|D|}, \ldots, \frac{|C_j|}{|D|} \right\} \tag{1}$$

By partitioning $D$ based on the value of a feature $F$ according to subsets $\{D_1, D_2, D_3, \ldots, D_n\}$, $Info(F,D)$ with respect to $F$ can be computed as:

$$Info(F,D) = \sum_{i=1}^{n} \frac{|D_i|}{|D|} Info(D_i) \tag{2}$$

The corresponding information gain after obtaining the value of $F$ is computed as:

$$Gain(F,D) = Info(D) - Info(F,D) \tag{3}$$

Thus, the *GainRatio* is defined as:

$$GainRatio(F,D) = \frac{Gain(F,D)}{SplitInfo(F,D)} \tag{4}$$

where, *SplitInfo(F, D)* shows the information due to the splitting of $D$ according to the feature $F$. Random Forest uses the majority voting of all the individual decisions to obtain the final decision of the classifier.

### 3.5.3 Naïve Bayes

A Naïve Bayes is a supervised probabilistic machine learning classifier that is based on Bayes' theorem with strong independence (naïve) assumption among the features. Naïve Bayesian classification assumes that the variables are independent given the classes. That is, Naïve Bayes assumes that the presence of a specific attribute in a class is unrelated to the presence of any other attributes.

Formally, let $C$ be the random variable denoting the class of an instance; Let $X$ be a vector of a random variable denoting the observed attribute values. Let $c$ be a particular class label and $x$ represent a particular observed attribute value. According to the independence assumption, attributes $x_1, x_2, \ldots, x_n$ are all conditionally independent of one another, given C. The value of this assumption is that it simplifies the representation of conditional probability *P(x|c)*. Naïve Bayes gives a way of finding the conditional probability *P(x|c)* from *P(c)*, *P(x)* and *P(c|x)*. This relationship is as described in Equation (5) below.

$$P(x \mid c) = \frac{P(c \mid x)P(x)}{P(c)} \tag{5}$$

where, *P(x|c)* is the posterior probability of class *x*, given *c*, *P(x)* is the prior probability of the class, *P(c|x)* represents the likelihood which is the probability of predictor, given class and *P(c)* represents the prior probability of predictor.

## 3.6 Evaluation Metrics

The details of the evaluation metrics employed in this study are discussed in this section. The metrics provide globally acceptable techniques to check the performance of the proposed method. In machine learning, model classification performance can be obtained *via* a confusion matrix to ascertain the model ability in classifying the instances under consideration. The confusion matrix, shown in Table 2, is a matrix that gives the classification performance on how well a classifier can separate one class from another. The table presents the confusion matrix general structure for the binary class classification problem. In this table, True Positive (TP) and True Negative (TN) refer to the number of correctly classified positive and negative sentiments, respectively. False Positive (FP) represents the number of negative sentiment documents classified as positive, while False Negative (FN) represents the number of positive sentiment documents classified as negative.

Table 2. Confusion matrix for a binary class problem (positive and negative sentiments).

| | | Predicted Class | |
|---|---|---|---|
| | | Class = Positive sentiment | Class = Negative sentiment |
| Actual Class | Class = Positive sentiment | TP | FN |
| | Class = Negative sentiment | FP | TN |

The parameters TP, TN, FP and FN, as shown in Table 2, can be used to derive some standard metrics, such as Accuracy, Precision, Recall, F-Measure and Receiver Operating Characteristics (ROCs). Each of these metrics is discussed as follows:

Accuracy: Accuracy is the most intuitive indicator of performance, as it a ratio of appropriately predicted observations to the overall observations. Formally, it is represented as:

$$Accuracy = \frac{Tp + Tn}{Tp + Tn + Fp + Fn} \tag{6}$$

Precision: Precision is a measure that evaluates the correct number of positive predictions. It is calculated as:

$$\mathrm{Pr}ecision = \frac{Tp}{Tp + Fp} \tag{7}$$

Recall: It is otherwise referred to as sensitivity which measures the model’s ability to correctly identify the true positives. Mathematically, it is expressed as:

$$\mathrm{Re}call = \frac{Tp}{Tp + Fn} \tag{8}$$

F-Measure: F-measure, otherwise called F-score, is a common evaluation metric for machine learning models. It is described as the harmonic mean of the precision and recall of a model. The relationship is as described in Equation 9.

$$F - Measure = 2 \times \frac{preccision \times recall}{precision + recall} \tag{9}$$

Receiver Operating Characteristics (ROCs): ROC term illustrates how well a classification model performs at all classification levels. A ROC curve is a graph that reveals the relationship between sensitivity and specificity for every individual possible cut-off. ROC curve is a graph in which the x-axis represents 1 – specificity, while the y-axis is the value of sensitivity.

## 4. EXPERIMENTAL RESULTS AND DISCUSSION

Attribute reduction is a vital phase of machine learning modeling, as it assists greatly in reducing the number of features needed for classification and subsequently reducing the classifier's processing time. This section discusses the results obtained when the hybrid feature selection framework was used on different corpora. It first presents the environment and detail specifications for the various experiments conducted. We then discuss the results obtained for each of the classification algorithms employed in this study without the proposed feature selection techniques. Finally, we present the results of the proposed hybrid feature selection framework based on filter and wrapper feature selection techniques. In this research, three machine learning classifiers (SVM, Naïve Bayes and Random Forest) were considered, with or without the combination of three feature selection optimizers (CFS, Boruta and RFE). The performance of the selected classifiers was evaluated using Accuracy, Precision, Recall, F-measure and ROCs.

### 4.1 Experimental Setup

Several experiments were conducted in this research based on two different machine learning tools; R and Weka. RStudio version 1.2.5001 was used as the Integrated Development Environment (IDE) for coding the R scripts. R language was used for preprocessing, feature extraction and feature selection including implementation of CFS, Boruta and RFE techniques. Weka was used to implement the selected classification algorithms. Four (4) datasets; namely, Amazon, Yelp, IMDB and Kaggle were considered. The descriptions of the datasets have been discussed in Section 3. The classification experiments were conducted on WEKA version 3.8 running on a 32GB-RAM personal computer with Core i9 2.90GHz processor speed. The computer is running on a 64-bit Windows 10 operating system. For training and testing of the classification algorithms, 10-fold cross-validation has been employed, which allows to obtain models that can be generalized when deployed in real-world for sentiment analysis on large corpora.

### 4.2 Discussion of Results

Table 3 summarizes the different experiments conducted, which are discussed in the subsequent sections. The first three experiments help investigate the performance of the selected classifiers using the original features extracted from each of the datasets. Experiments 4, 5 and 6 discussed the results of the three classification algorithms when considering the proposed hybrid feature selection approach with CFS combined with Boruta. The last three experiments discussed the results of the classification algorithms on the proposed hybrid features by combining CFS with RFE.

Table 3. Summary of experiments.

| S/No. | Description of experiments |
|---|---|
| 1. | Experiment based on SVM with original features |
| 2. | Experiment based on Naïve Bayes with original features |
| 3. | Experiment based on Random Forest with original features |
| 4. | Experiment based on SVM with hybrid features using CFS and Boruta |
| 5. | Experiment based on Naïve Bayes with hybrid features using CFS and Boruta |
| 6. | Experiment based on Random Forest with hybrid features using CFS and Boruta |
| 7. | Experiment based on SVM with hybrid features using CFS and RFE |
| 8. | Experiment based on Naïve Bayes with hybrid features using CFS and RFE |
| 9. | Experiment based on Random Forest with hybrid features using CFS and RFE |

#### 4.2.1 Results of SVM with Original Features

To statistically understand the contribution of the proposed hybrid framework, each of the classifiers selected was evaluated with original features as extracted from the four datasets that were considered in this study. Table 4 shows the results of SVM with original feature subsets. SVM algorithm was evaluated on the four datasets without any feature selection technique. The results show that SVM achieved an accuracy of 80.1%, 75.3%, 74.2% and 66.8% on Amazon, Yelp, IMDB and Kaggle datasets, respectively. We observed that the performance of SVM classifier drops slightly when the number of features is increasing. For instance, SVM yielded an accuracy of 66.8% on the Kaggle dataset which has the highest number of features. The results obtained based on the other performance metrics using SVM classifier have been highlighted in Table 4.

Table 4. Results of SVM with original features.

| | | Datasets | | | |
|---|---|---|---|---|---|
| | | **Amazon** | **Yelp** | **IMDB** | **Kaggle** |
| **Algorithm** | **# of Features** | **620** | **691** | **961** | **1218** |
| SVM | Accuracy | 0.801 | 0.753 | 0.742 | 0.668 |
| | Precision | **0.801** | **0.754** | **0.743** | **0.649** |
| | Recall | 0.801 | 0.753 | 0.742 | 0.668 |
| | F-Measure | 0.801 | 0.753 | 0.742 | 0.654 |
| | ROC | 0.801 | 0.753 | 0.741 | **0.687** |

### 4.2.2 Results of Naïve Bayes with Original Features

Similarly, we evaluated the performance of Naïve Bayes classifier with original feature subsets. The results based on the four datasets are presented in Table 5. This results show a slight drop in the performance of Naïve Bayes when compared with SVM results in Table 4 based on the accuracy metric. However, the ROC results across the four datasets produced better results when compared with SVM in Table 4. ROC values of 80.9%, 78.8%, 78.5% and 71.1% were obtained on Amazon, Yelp, IMDB and Kaggle datasets, respectively. Similarly, we observed a similar pattern in the results produced by the classifier when the number of features is reduced.

Table 5. Results of Naïve Bayes with original features.

| | | Datasets | | | |
|---|---|---|---|---|---|
| | | **Amazon** | **Yelp** | **IMDB** | **Kaggle** |
| **Algorithm** | **# of Features** | **620** | **691** | **961** | **1218** |
| Naïve Bayes | Accuracy | 0.717 | 0.726 | 0.716 | 0.590 |
| | Precision | 0.717 | 0.735 | 0.716 | 0.61 |
| | Recall | 0.717 | 0.726 | 0.716 | 0.59 |
| | F-Measure | 0.717 | 0.723 | 0.716 | 0.598 |
| | ROC | **0.809** | **0.788** | **0.785** | **0.711** |

### 4.2.3 Results of Random Forest with Original Features

Random Forest classifier achieved the best result with original feature subsets based on the ROC metric. This result produced 86%, 85.2%, 81% and 77.8% for Amazon, Yelp, IMDB and Kaggle datasets, respectively. The result based on the Kaggle dataset also drops slightly when compared with other datasets as observed in the previous results. This result is shown in Table 6.

Table 6. Results of Random Forest with original features.

| | | Datasets | | | |
|---|---|---|---|---|---|
| | | **Amazon** | **Yelp** | **IMDB** | **Kaggle** |
| | **# of Features** | **620** | **691** | **961** | **1218** |
| Random Forest | Accuracy | 0.777 | 0.763 | 0.744 | 0.675 |
| | Precision | 0.777 | 0.769 | 0.746 | 0.657 |
| | Recall | 0.777 | 0.765 | 0.744 | 0.675 |
| | F-Measure | 0.777 | 0.700 | 0.743 | 0.658 |
| | ROC | **0.86** | **0.852** | **0.81** | **0.778** |

### 4.2.4 Results' Comparison Based on Original Features

To understand the variation in the results obtained with original feature subsets, this sub-section provides a detailed discussion of the results of the classification algorithms based on the four datasets considered.

(a) **Results of Classifiers with the Original Amazon Dataset:** Table 7 shows the results of three classification algorithms without using the proposed feature selection approach. SVM had the highest Accuracy of 80.1% using the original dataset, as against Naïve Bayes and Random Forest having 71.7% and 77.7%, respectively. It also had the highest F-Measure point of 0.801, using the original dataset, against Naïve Bayes and Random Forest that had 0.717 and 0.777, respectively. Using ROC metrics on the three classifiers, it was observed that Random Forest had a better ROC of 0.86 than SVM (0.801) and Naïve Bayes (0.809). Therefore, it was shown that SVM had better performance than Naïve Bayes and Random Forest in sentiment classification on Amazon dataset in terms of Accuracy, Precision, Recall and F-measure, while Random Forest slightly outperformed SVM and Naïve Bayes based on ROC metric.

Table 7. Results' comparison with original features based on Amazon dataset.

| | # of Features | Accuracy (%) | Precision | Recall | F-Measure | ROC |
|---|---|---|---|---|---|---|
| **SVM** | 620 | **80.1** | **0.801** | **0.801** | **0.801** | 0.801 |
| **Naïve Bayes** | 620 | 71.7 | 0.717 | 0.717 | 0.717 | 0.809 |
| **Random Forest** | 620 | 77.7 | 0.777 | 0.777 | 0.777 | **0.860** |

(b) **Results of Classifiers with the Original Yelp Dataset:** From Table 8, the performance of the three classifiers without using the proposed feature selection approach shows that Random Forest with an Accuracy of 76.25% outperforms both SVM (75.3%) and Naïve Bayes (72.6%). SVM has 0.753 value of F-Measure which is higher than the corresponding values for both Naïve Bayes and Random Forest classifiers with F-Measure of 0.723 and 0.700, respectively. Equally, Random Forest classifier has a higher ROC point (0.852) over both SVM (0.753) and Naïve Bayes (0.788) classifiers. Therefore, Random Forest classifier outperforms both SVM and Naïve Bayes classifiers on Yelp dataset by considering Accuracy, Precision, Recall and ROC. However, SVM has the highest F-Measure.

Table 8. Results' comparison with original features based on Yelp dataset.

| | # of Features | Accuracy (%) | Precision | Recall | F-Measure | ROC |
|---|---|---|---|---|---|---|
| **SVM** | 691 | 75.30 | 0.754 | 0.753 | **0.753** | 0.753 |
| **Naïve Bayes** | 691 | 72.60 | 0.735 | 0.726 | 0.723 | 0.788 |
| **Random Forest** | 691 | **76.25** | **0.769** | **0.765** | 0.700 | **0.852** |

(c) **Results of Classifiers with the Original IMDB Dataset:** Table 9 shows the results of the performances of the three classifiers on IMDB dataset without the application of the proposed feature selection approach. It was observed that Random Forest classifier has the records of 74.37% Accuracy, 0.743 F-Measure and 0.81 ROC, which outperforms the results obtained with SVM and Naïve Bayes. This results show that Random Forest classifier is a promising algorithm for sentiment analysis on IMDB dataset.

Table 9. Results' comparison with original features based on IMDB dataset.

| | # of Features | Accuracy (%) | Precision | Recall | F-Measure | ROC |
|---|---|---|---|---|---|---|
| **SVM** | 961 | 74.17 | 0.743 | 0.742 | 0.742 | 0.741 |
| **Naïve Bayes** | 961 | 71.57 | 0.716 | 0.716 | 0.716 | 0.785 |
| **Random Forest** | 961 | **74.37** | **0.746** | **0.744** | **0.743** | **0.810** |

(d) **Results of Classifiers with the Original Kaggle Dataset:** Table 10 shows the results of the three classifiers without the proposed feature selection approach. The best performance is recorded by Random Forest classifier which achieved an Accuracy of (67.53%), an F-Measure of (0.658) and a ROC metric of (0.778). This performance is followed by SVM classifier with an accuracy of 66.75%, an F-Measure of (0.654) and a ROC metric of (0.687). It is noteworthy that despite the least performance record credited to the Naïve Bayes classifier, it is observed that it outperforms SVM classifier based on ROC evaluation metric. This result implies that Random Forest classifier is still considered as the promising classification algorithm for sentiment analysis based on the Kaggle dataset.

Table 10. Results' comparison with original features based on Kaggle dataset.

| | # of Features | Accuracy (%) | Precision | Recall | F-Measure | ROC |
|---|---|---|---|---|---|---|
| **SVM** | 1218 | 66.75 | 0.649 | 0.668 | 0.654 | 0.687 |
| **Naïve Bayes** | 1218 | 59.02 | 0.61 | 0.59 | 0.598 | 0.711 |
| **Random Forest** | 1218 | **67.53** | **0.657** | **0.675** | **0.658** | **0.778** |

#### 4.2.5 Results of the Hybrid Framework Using CFS+Boruta and SVM Classifier

Table 11 shows the results obtained using the proposed hybrid framework, which comprises the hybridization of CFS and Boruta algorithms. These results show that across the four datasets used in this study, SVM can provide an accuracy of 78.5%, 76.9%, 71.67% and 65.16% on Amazon, Yelp, IMDB and Kaggle datasets, respectively despite a significant reduction in the number of features for the classification task. This shows that the proposed hybrid framework produced promising results despite the huge reduction in the number of features considered. The features selected by the proposed CFS + Boruta feature selection method for each dataset using SVM classifier are shown in the Appendix.

Table 11. Results for SVM Classifier based on hybrid (CFS + Boruta) feature selection.

| | | | **Amazon** | **Yelp** | **IMDB** | **Kaggle** |
|---|---|---|---|---|---|---|
| **Hybrid Feature Selection Algorithm** | **Classifier** | **# of Features** | 620 | 691 | 961 | 1218 |
| CFS + Boruta | SVM | Selected Features | **23** | **25** | **25** | **44** |
| | | Accuracy (%) | **78.5** | **76.9** | **71.67** | **65.16** |
| | | Precision | 0.812 | 0.801 | 0.754 | 0.611 |
| | | Recall | 0.785 | 0.769 | 0.717 | 0.652 |
| | | F-Measure | 0.78 | 0.763 | 0.706 | 0.59 |
| | | ROC | **0.785** | **0.769** | **0.716** | **0.62** |

#### 4.2.6 Results of the Hybrid Framework Using CFS+Boruta and Naïve Bayes Classifier

Table 12 shows the results obtained using the proposed hybrid framework; that is, hybridization of CFS and Boruta algorithms. These results show that across the four datasets used in this study, Naïve Bayes can provide an accuracy of 74.1%, 71.7%, 71.37% and 63.38% on Amazon, Yelp, IMDB and Kaggle datasets, respectively. This result shows a slight drop in performance of Naïve Bayes when compared with SVM algorithm with the proposed hybrid framework that is based on CFS and Boruta algorithms. However, a significant improvement in the ROC metric was observed when Naïve Bayes was used as the classification algorithm, as shown in Table 12.

Table 12. Results for Naïve Bayes Classifier based on hybrid (CFS + Boruta) feature selection.

| | | | **Amazon** | **Yelp** | **IMDB** | **Kaggle** |
|---|---|---|---|---|---|---|
| **Hybrid Feature Selection Algorithm** | **Classifier** | **# of Features** | 620 | 691 | 961 | 1218 |
| CFS + Boruta | Naïve Bayes | Selected Features | **23** | **25** | **25** | **44** |
| | | Accuracy (%) | **74.1** | **71.7** | **71.37** | **63.38** |
| | | Precision | 0.783 | 0.765 | 0.727 | 0.594 |
| | | Recall | 0.741 | 0.717 | 0.714 | 0.634 |
| | | F-Measure | 0.731 | 0.704 | 0.709 | 0.582 |
| | | ROC | **0.791** | **0.794** | **0.76** | **0.691** |

#### 4.2.7 Results of the Hybrid Framework Using CFS+Boruta and Random Forest Cassifier

Random Forest algorithm when used with the proposed hybrid framework that is comprised of CFS and Boruta algorithm was able to produce the best results in terms of accuracy and ROC metrics, as shown in Table 13. According to this table, accuracies of 78.9%, 76.9%, 71.87% and 65.76% were obtained on Amazon, Yelp, IMDB and Kaggle datasets, respectively. The ROC results also show a considerable increase in the performance of Random Forest when compared with SVM and Naïve Bayes classifiers based on the four datasets considered in this study. This result further testifies to the applicability of the proposed hybrid framework for sentient analysis considering different problem domains. The result obtained is promising despite a significant reduction in the number of features as compared with the original datasets.

Table 13. Results for Random Forest Classifier based on hybrid (CFS + Boruta) feature selection.

| | | | **Amazon** | **Yelp** | **IMDB** | **Kaggle** |
|---|---|---|---|---|---|---|
| **Hybrid Feature Selection Algorithm** | **Classifier** | **# of Features** | 620 | 691 | 961 | 1218 |
| CFS + Boruta | Random Forest | Selected Features | **23** | **25** | **25** | **44** |
| | | Accuracy (%) | **78.9** | **76.9** | **71.87** | **65.76** |
| | | Precision | 0.816 | 0.801 | 0.756 | 0.621 |
| | | Recall | 0.789 | 0.769 | 0.719 | 0.658 |
| | | F-Measure | 0.784 | 0.763 | 0.708 | 0.594 |
| | | ROC | **0.808** | **0.801** | **0.776** | **0.711** |

#### 4.2.8 Results of the Hybrid Framework Using CFS+RFE and SVM Classifier

Next is to discuss the results obtained when using the proposed hybrid framework for feature selection for sentiment analysis that considered the hybridization of CFS and RFE algorithms to effectively select the most discriminative features. Table 14 shows the results obtained using this approach. Accuracies of 64.2%, 68.1%, 72.27% and 63.56% were obtained based on Amazon, Yelp, IMDB and Kaggle datasets, respectively. Hybridization using CFS and RFE algorithms selected the least number of features in most cases when compared with the number of features selected using CFS and Boruta algorithms. It can easily be seen that the results obtained with the CFS + Boruta outperformed those obtained when using the CFS + RFE method. This was also manifested in the results obtained when considering ROC metric. The features selected by the proposed CFS + RFE feature selection method for each dataset using SVM classifier are shown in the Appendix.

Table 14. Results for SVM Classifier based on hybrid (CFS + RFE) feature selection.

| | | | **Amazon** | **Yelp** | **IMDB** | **Kaggle** |
|---|---|---|---|---|---|---|
| **Hybrid Feature Selection Algorithm** | **Classifier** | **# of Features** | 620 | 691 | 961 | 1218 |
| CFS + RFE | SVM | Selected Features | **3** | **7** | **70** | **22** |
| | | Accuracy (%) | **64.2** | **68.1** | **72.27** | **63.56** |
| | | Precision | 0.738 | 0.742 | 0.725 | 0.536 |
| | | Recall | 0.642 | 0.681 | 0.723 | 0.636 |
| | | F-Measure | 0.602 | 0.66 | 0.722 | 0.535 |
| | | ROC | **0.642** | **0.681** | **0.722** | **0.577** |

#### 4.2.9 Results of the Hybrid Framework Using CFS+RFE and Naïve Bayes Classifier

Table 15 shows the results of Naïve Bayes classifier based on hybridization of CFS and RFE algorithms. These results show accuracies of 64.2%, 68.2%, 73.87% and 62.71% on Amazon, Yelp, IMDB and Kaggle datasets, respectively. This result still shows a reduction in performance when compared with the results obtained using CFS and Boruta algorithms with Naïve Bayes classifier.

#### 4.2.10 Results of the Hybrid Framework Using CFS+RFE and Random Forest Classifier

Table 16 shows the results of the Random Forest algorithm based on CFS and RFE hybridization. These results show accuracies of 64.2%, 68.2%, 76.38% and 63.94% on Amazon, Yelp, IMDB and

Table 15. Results for Naïve Bayes Classifier based on hybrid (CFS + RFE) feature selection.

| | | | Amazon | Yelp | IMDB | Kaggle |
|---|---|---|---|---|---|---|
| Hybrid Feature Selection Algorithm | Classifier | # of Features | 620 | 691 | 961 | 1218 |
| CFS + RFE | Naïve Bayes | Selected Features | **3** | **7** | **70** | **22** |
| | | Accuracy (%) | **64.2** | **68.2** | **73.87** | **62.71** |
| | | Precision | 0.738 | 0.744 | 0.75 | 0.703 |
| | | Recall | 0.642 | 0.682 | 0.739 | 0.627 |
| | | F-Measure | 0.602 | 0.66 | 0.736 | 0.54 |
| | | ROC | **0.631** | **0.7** | **0.818** | **0.668** |

Kaggle datasets, respectively. When compared with the results of the Random Forest classifier based on hybridization of CFS and Boruta algorithms, there is a significant drop in performance based on CFS and RFE approach proposed in this study. Similar results were obtained using the ROC metric as an evaluation metric.

Table 16. Results for Random Forest classifier based on hybrid (CFS + RFE) feature selection.

| | | | Amazon | Yelp | IMDB | Kaggle |
|---|---|---|---|---|---|---|
| Hybrid Feature Selection Algorithm | Classifier | # of Features | 620 | 691 | 961 | 1218 |
| CFS + RFE | Random Forest | Selected Features | **3** | **7** | **70** | **22** |
| | | Accuracy (%) | **64.2** | **68.2** | **76.38** | **63.94** |
| | | Precision | 0.738 | 0.744 | 0.779 | 0.565 |
| | | Recall | 0.642 | 0.682 | 0.764 | 0.639 |
| | | F-Measure | 0.602 | 0.66 | 0.761 | 0.543 |
| | | ROC | **0.631** | **0.697** | **0.852** | **0.679** |

#### 4.2.11 Results' Comparison of the Classifiers Based on the Hybrid Framework

In this sub-section, we now compare the results of the two proposed hybrid methods (CFS + Boruta and CFS + RFE) based on the individual datasets.

**(a) Amazon Dataset**

The results in Table 17 show that the proposed hybridization method using CFS + Boruta selected 23 features on Amazon dataset as compared with 620 features available in the original dataset, while CFS + RFE approach selected 3 features. The classification results revealed that Random Forest classifier outperformed the other two classification algorithms based on all the metrics used for evaluation in this study. Using 23 reduced features based on CFS + Boruta, Random Forest was able to produce the following metrics: Accuracy (78.9%), Precision (81.60%), Recall (78.9%), F-measure (78.4%) and ROC (80.80%). This result is promising when considering the number of features used for the classification task.

Table 17. Performance evaluation of the hybrid feature selection framework based on Amazon dataset.

| | # of Features | Accuracy (%) | Precision | Recall | F-Measure | ROC |
|---|---|---|---|---|---|---|
| CFS+Boruta+SVM | 23 | 78.5 | 0.812 | 0.785 | 0.78 | 0.785 |
| CFS+Boruta+NB | 23 | 74.1 | 0.783 | 0.741 | 0.731 | 0.791 |
| **CFS+Boruta+RF** | **23** | **78.9** | **0.816** | **0.789** | **0.784** | **0.808** |
| CFS+RFE+SVM | 3 | 64.2 | 0.738 | 0.642 | 0.602 | 0.642 |
| CFS+RFE+NB | 3 | 64.2 | 0.738 | 0.642 | 0.602 | 0.631 |
| CFS+RFE+RF | 3 | 64.2 | 0.738 | 0.642 | 0.602 | 0.631 |

**(b) Yelp Dataset**

The results in Table 18 show that the proposed hybridization method using CFS + Boruta selected 25 features on Yelp dataset as compared with 691 features available in the original dataset, while CFS + RFE approach selected 7 features. The classification results revealed that Random Forest classifier outperformed the other two classification algorithms based on all the metrics used for evaluation. With the 25 reduced features based on CFS + Boruta, Random Forest was able to produce the following metrics: Accuracy (76.9%), Precision (80.10%), Recall (76.9%), F-measure (76.3%) and ROC (80.10%). SVM classifier also produced similar results as compared with Random Forest classifier except for the ROC of Random Forest classifier that is slightly higher than the one obtained with SVM. Furthermore, this result is promising when considering the number of features used for the classification task.

Table 18. Performance evaluation of the hybrid feature selection framework based on Yelp dataset.

| | **# of Features** | **Accuracy (%)** | **Precision** | **Recall** | **F-Measure** | **ROC** |
|---|---|---|---|---|---|---|
| CFS+Boruta+SVM | 25 | 76.9 | 0.801 | 0.769 | 0.763 | 0.769 |
| CFS+Boruta+NB | 25 | 71.7 | 0.765 | 0.717 | 0.704 | 0.794 |
| **CFS+Boruta+RF** | **25** | **76.9** | **0.801** | **0.769** | **0.763** | **0.801** |
| CFS+RFE+SVM | 7 | 68.1 | 0.742 | 0.681 | 0.66 | 0.681 |
| CFS+RFE+NB | 7 | 68.2 | 0.744 | 0.682 | 0.66 | 0.7 |
| CFS+RFE+RF | 7 | 68.2 | 0.744 | 0.682 | 0.66 | 0.697 |

**(c) IMDB Dataset**

The results in Table 19 show that the proposed hybridization method using CFS + Boruta selected 25 features on IMDB dataset as compared with 961 features available in the original dataset, while CFS + RFE approach selected 70 features. This result shows an increase in the number of features selected by RFE when compared with the previous results. The classification results revealed a different scenario in which CFS + RFE outperformed CFS + Boruta according to the results of the RFE classifier based on the evaluation metrics. With the 70 reduced features based on CFS + RFE, Random Forest was able to produce the following metrics: Accuracy (76.38%), Precision (77.9%), Recall (76.4%), F-measure (76.1%) and ROC (85.20%).

Table 19. Performance evaluation of the hybrid feature selection framework based on IMDB dataset.

| | **# of Features** | **Accuracy (%)** | **Precision** | **Recall** | **F-Measure** | **ROC** |
|---|---|---|---|---|---|---|
| CFS+Boruta+SVM | 25 | 71.67 | 0.754 | 0.717 | 0.706 | 0.716 |
| CFS+Boruta+NB | 25 | 71.37 | 0.727 | 0.714 | 0.709 | 0.76 |
| CFS+Boruta+RF | 25 | 71.87 | 0.756 | 0.719 | 0.708 | 0.776 |
| CFS+RFE+SVM | 70 | 72.27 | 0.725 | 0.723 | 0.722 | 0.722 |
| CFS+RFE+NB | 70 | 73.87 | 0.75 | 0.739 | 0.736 | 0.818 |
| **CFS+RFE+RF** | **70** | **76.38** | **0.779** | **0.764** | **0.761** | **0.852** |

**(d) Kaggle Dataset**

The results in Table 20 show that the proposed hybridization method using CFS + Boruta selected 44 features on the Kaggle dataset as compared with 1218 features available in the original dataset, while CFS + RFE approach selected 22 features. The classification results revealed that Random Forest classifier still outperformed the other two classification algorithms as revealed by the results of the different evaluation metrics. Using the reduced 44 features based on CFS + Boruta showed that Random Forest classifier was able to produce the following metrics: Accuracy (65.76%), Precision (62.10%), Recall (65.80%), F-measure (59.40%) and ROC (71.10%). Taking a closer look at the results in this table, it can be seen that SVM classifier also achieved very close results with Random

Forest algorithm. Besides, it was noticed that the Precision (70.30%), accounting for the highest value of precision for all the classification cases, was obtained with CFS + RFE + NB classifier. This is the only scenario where Naïve Bayes outperformed the other classification algorithms from the various results obtained in this study based on the four datasets considered for analysis. The results in Table 20 further strengthen the superiority of the proposed CFS + Boruta hybrid algorithm based on Random Forest classifier.

Table 20. Performance evaluation of the hybrid feature selection framework based on Kaggle dataset.

| | **# of Features** | **Accuracy (%)** | **Precision** | **Recall** | **F-Measure** | **ROC** |
|---|---|---|---|---|---|---|
| CFS+Boruta+SVM | 44 | 65.16 | 0.611 | 0.652 | 0.59 | 0.62 |
| CFS+Boruta+NB | 44 | 63.38 | 0.594 | 0.634 | 0.582 | 0.691 |
| **CFS+Boruta+RF** | **44** | **65.76** | **0.621** | **0.658** | **0.594** | **0.711** |
| CFS+RFE+SVM | 22 | 63.56 | 0.536 | 0.636 | 0.535 | 0.577 |
| CFS+RFE+NB | 22 | 62.71 | 0.703 | 0.627 | 0.54 | 0.668 |
| CFS+RFE+RF | 22 | 63.94 | 0.565 | 0.639 | 0.543 | 0.679 |

## 4.3 Results' Comparison with and without the Proposed Hybrid Algorithm

In this sub-section, we compare the results of the classification algorithms with and without the application of the proposed hybrid feature selection algorithm. More specifically, we selected the best results obtained in each case for analysis. This involves comparing the best results of the classifiers on the original features and also on the feature subsets selected by the proposed hybrid feature selection algorithm. The results' analyses have been grouped under the dataset used for the experiment in each scenario.

### (a) Results from Comparison Based on Amazon Dataset with and without Hybrid Feature Selection

Figure 2 shows the results obtained with and without the proposed hybrid feature selection method. According to this figure, Accuracy (80.1%), Precision (80.1%), Recall (80.1%), F-measure (80.1%) and ROC (80.1%) were obtained based on SVM algorithm with original 620 features available on

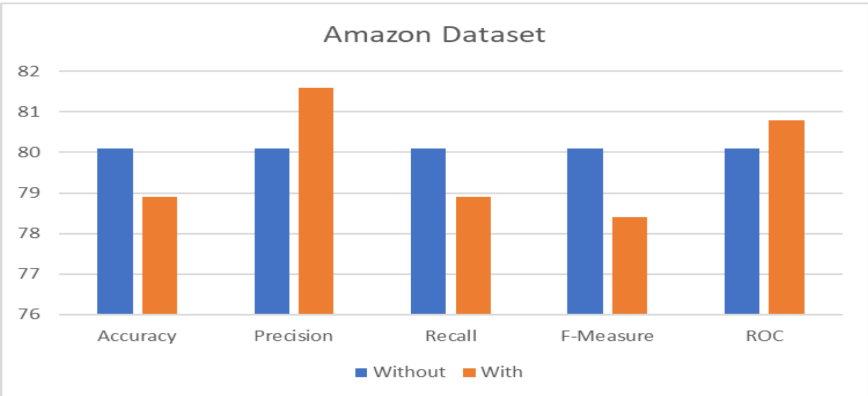



Figure 2. Results of comparison based on Amazon dataset with and without hybrid feature selection.

Amazon dataset, while Accuracy (78.9%), Precision (81.6%), Recall (78.9%), F-measure (78.4%) and ROC (80.8%) were obtained when the proposed hybrid feature selection algorithm was used based on 23 features. This result shows a significant improvement when considering the number of irrelevant features that have been removed from the original Amazon dataset. Specifically, the noticeable achievement was observed in the ROC result when the proposed hybrid feature selection algorithm was used. A similar thing was noticed for the Precision result. By considering the percentage of reduction in the number of features (96.29%), the results obtained with the proposed hybrid feature selection are promising and further confirmed the superiority and applicability of the proposed hybrid feature selection method.

**(b) Results' Comparison Based on Yelp Dataset with and without Hybrid Feature Selection**

Figure 3 shows the results obtained with and without the proposed hybrid feature selection method using Yelp dataset. According to this figure, Accuracy (76.25%), Precision (76.9%), Recall (76.5%), F-measure (70.00%) and ROC (85.2%) were obtained based on Random Forest algorithm with original 691 features available on Yelp dataset, while Accuracy (76.9%), Precision (80.1%), Recall (76.9%), F-measure (76.3%) and ROC (80.1%) were obtained when the proposed hybrid feature selection algorithm (CFS + Boruta) was used based on 25 features. This result shows a significant improvement when considering the number of irrelevant features that have been removed from the original Yelp dataset. More importantly is the improvement cut across the different evaluation metrics used in this study as shown in the figure. However, the ROC result of the hybrid framework drops slightly when compared with the ROC result obtained based on original features. By considering the percentage of reduction in the number of features (96.38%), the results obtained with the proposed hybrid feature selection are promising and superior. Also, this further confirmed the superiority and applicability of the proposed hybrid feature selection method.

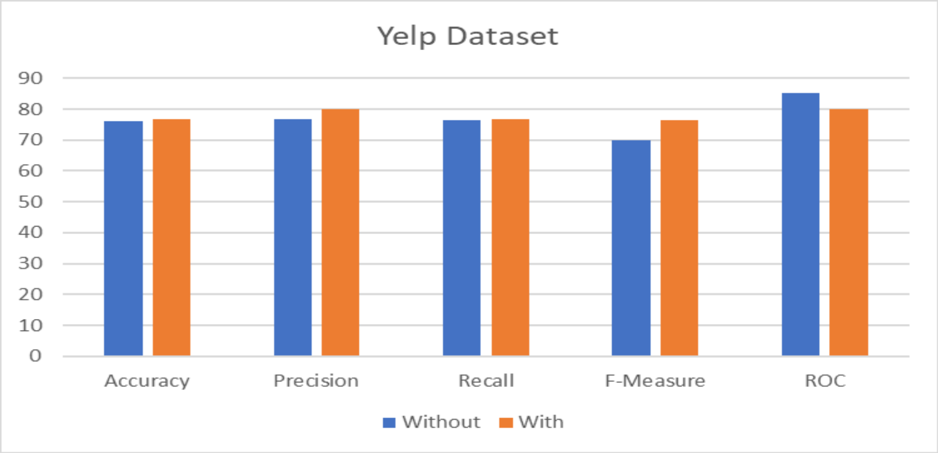



Figure 3. Results of comparison based on Yelp dataset with and without hybrid feature selection.

**(c) Results' Comparison Based on IMDB Dataset with and without Hybrid Feature Selection**

As shown in Figure 4, the results obtained with the proposed hybrid feature selection method outperformed the results achieved when feature selection was not used on IMDB dataset. According to this figure, Accuracy (74.37%), Precision (74.46%), Recall (74.4%), F-measure (74.30%) and ROC (81.0%) were obtained based on Random Forest algorithm with original 961 features available on IMDB dataset, while Accuracy (76.38%), Precision (77.9%), Recall (76.40%), F-measure (76.10%) and ROC (85.2%) were obtained when the proposed hybrid feature selection algorithm was used. This result shows a significant improvement when considering the number of irrelevant features that have been removed from the original IMDB dataset. More importantly is the improvement cut across the

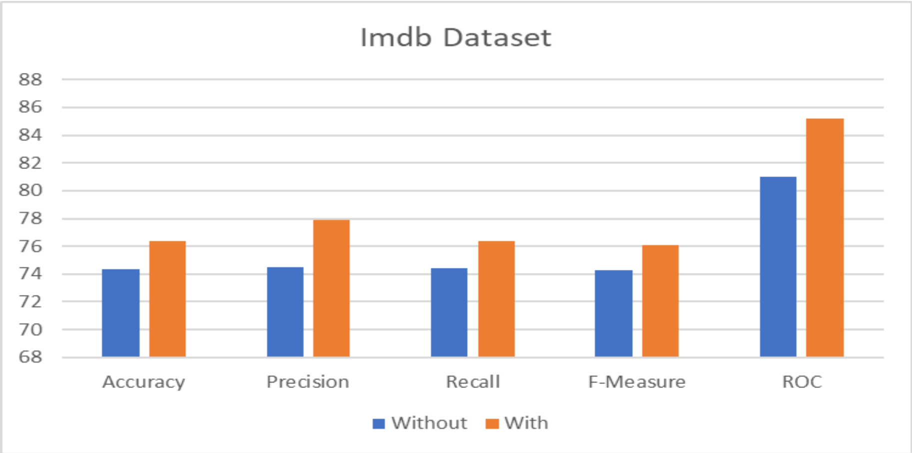



Figure 4. Results of comparison based on IMDB dataset with and without hybrid feature selection.

different evaluation metrics used in this study as shown in the figure. Based on the percentage of reduction in the number of features (92.72%), the results obtained with the proposed hybrid feature selection are promising and superior. Also, this further confirmed the superiority and applicability of the proposed hybrid feature selection method.

**(d) Results' Comparison Based on Kaggle Dataset with and without Hybrid Feature Selection**

Figure 5 shows the results obtained with and without the proposed hybrid feature selection method using the Kaggle dataset. About 96.39% reduction in the original features available in the dataset was achieved. According to this figure, Accuracy (67.53%), Precision (65.7%), Recall (67.5%), F-measure (65.8%) and ROC (77.8%) were obtained based on Random Forest algorithm with original 1218 features available on the Kaggle dataset, while Accuracy (65.76%), Precision (62.1%), Recall (65.8%), F-measure (59.4%) and ROC (71.1%) were obtained when the proposed hybrid feature selection algorithm (CFS + Boruta) was used based on 44 features. The results obtained without the use of the proposed hybrid feature selection algorithm dropped on the Kaggle dataset. However, by considering the number of features used for the classification, the results obtained across the different evaluation metrics still show the applicability of the proposed hybrid feature selection algorithm to reduce model complexity while still achieving comparable performance with the original features.

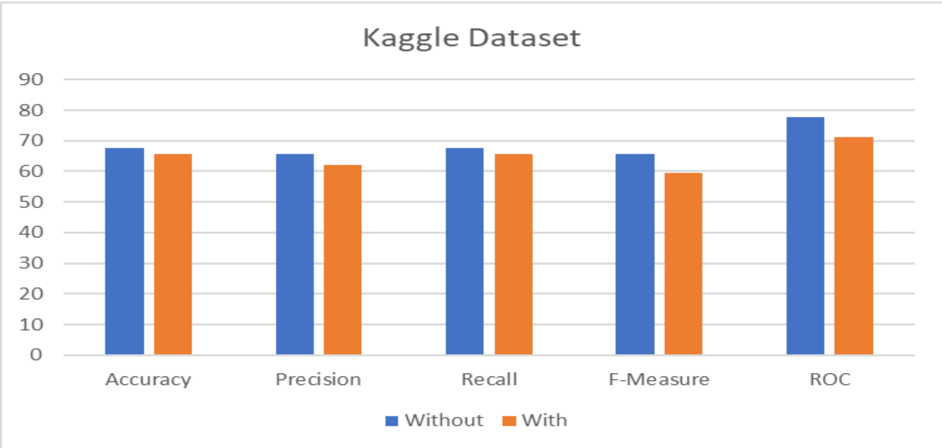



Figure 5. Results of comparison based on Kaggle dataset with and without hybrid feature selection.

## 5. Conclusions

Sentiment analysis studies have been receiving wide attention in recent years, mostly due to their significant role in opinion mining and prediction. Sentiment analysis has helped organizations understand customers' opinions as well as their attitudes towards particular products or brands. In this study, a hybrid feature selection framework was proposed to address the research issue of identifying discriminating attributes that can be used to model customers' opinions across different domains. This study employed four public datasets (Amazon, Yelp, IMDB and Kaggle) to examine the applicability of the proposed hybrid feature selection framework for sentiment analysis. The proposed hybrid feature selection framework has two levels of feature selection strategies by employing filter- and wrapper-based feature selection. Correlation-based feature selection (CFS) with Best First Search (BFS) method has been used at the top layer of the proposed framework and the bottom layer is based on the combination of CFS with either Boruta or Recursive Feature Elimination (RFE) wrapper-based feature selection method. The goal is to examine the specific combination of the feature selection approach that will provide improved performance for sentiment analysis across different domains. Therefore, based on several experiments conducted using three classification algorithms: SVM, Naïve Bayes and Random Forest, the study was able to establish the superiority and applicability of the proposed hybrid feature selection framework using well-known evaluation metrics.

CFS combined with Boruta produced the most promising results. Therefore, it is recommended to build a prototype for sentiment analysis tasks across different domains. The proposed hybrid feature selection approach reduced the number of features from the original datasets by 95% on average while still maintaining promising classification results. It was observed that the Random Forest algorithm demonstrated superiority over the two other classifiers by producing interesting results across the four datasets used in this study. Despite a considerable reduction in the number of features used for classification, performance figures are somewhat lower, but relatively on par with evaluation using the full feature set, but the computing time of the resulting model is shorter as a result of the proposed hybrid feature selection framework. As observed in the results produced during this study, there is still the need to focus on improving the classification accuracy of the proposed framework while still ensuring that model complexity is reduced to save prediction time. Also, future work should further investigate other feature selection strategies, which may likely produce better results when still considering domain-specific sentiment analysis.

## REFERENCES

[1] M. A. Hassonah, R. Al-Sayyed, A. Rodan, A.-Z. Ala'm, I. Aljarah and H. Faris, "An Efficient Hybrid Filter and Evolutionary Wrapper Approach for Sentiment Analysis of Various Topics on Twitter," Knowledge-based Systems, vol. 192, p. 105353, 2020.

[2] 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.

[3] S. M. Rezaeinia, R. Rahmani, A. Ghodsi and H. Veisi, "Sentiment Analysis Based on Improved Pre-trained Word Embeddings," Expert Systems with Applications, vol. 117, pp. 139-147, 2019.

[4] R. Arulmurugan, K. Sabarmathi and H. Anandakumar, "Classification of Sentence Level Sentiment Analysis Using Cloud Machine Learning Techniques," Cluster Comp., vol. 22, pp. 1199-1209, 2019.

[5] A. Hasan, S. Moin, A. Karim and S. Shamshirband, "Machine Learning-based Sentiment Analysis for Twitter Accounts," Mathematical and Computational Applications, vol. 23, p. 11, 2018.

[6] H. H. Do, P. Prasad, A. Maag and A. Alsadoon, "Deep Learning for Aspect-based Sentiment Analysis: A Comparative Review," Expert Systems with Applications, vol. 118, pp. 272-299, 2019.

[7] Y. Wang, M. Wang and H. Fujita, "Word Sense Disambiguation: A Comprehensive Knowledge Exploitation Framework," Knowledge-based Systems, vol. 190, p. 105030, 2020.

[8] M. Taboada, J. Brooke, M. Tofiloski, K. Voll and M. Stede, "Lexicon-based methods for sentiment analysis," Computational Linguistics, vol. 37, pp. 267-307, 2011.

[9] G. Ansari, T. Ahmad and M. N. Doja, "Hybrid Filter–Wrapper Feature Selection Method for Sentiment Classification," Arabian Journal for Science and Engineering, vol. 44, pp. 9191-9208, 2019.

[10] Y. A. Alsariera, V. E. Adeyemo, A. O. Balogun and A. K. Alazzawi, "AI Meta-learners and Extra-trees Algorithm for the Detection of Phishing Websites," IEEE Access, vol. 8, pp. 142532-142542, 2020.

[11] M. S. Akhtar, D. Gupta, A. Ekbal and P. Bhattacharyya, "Feature Selection and Ensemble Construction: A Two-step Method for Aspect Based Sentiment Analysis," Knowledge-Based Systems, vol. 125, pp. 116-135, 2017.

[12] A. O. Balogun, S. Basri, S. Mahamad, S. J. Abdulkadir, M. A. Almomani, V. E. Adeyemo et al., "Impact of Feature Selection Methods on the Predictive Performance of Software Defect Prediction Models: An Extensive Empirical Study," Symmetry, vol. 12, p. 1147, 2020.

[13] B. A. Oluwagbemiga, B. Shuib, S. J. Abdulkadir and A. Sobri, "A Hybrid Multi-filter Wrapper Feature Selection Method for Software Defect Predictors," International Journal of Supply Chain Management, vol. 8, pp. 916-922, 2019.

[14] A. O. Balogun, S. Basri, S. J. Abdulkadir and A. S. Hashim, "Performance Analysis of Feature Selection Methods in Software Defect Prediction: A Search Method Approach," Applied Sciences, vol. 9, p. 2764, 2019.

[15] B. Agarwal and N. Mittal, "Machine Learning Approach for Sentiment Analysis," Proc. of Prominent Feature Extraction for Sentiment Analysis, pp. 21-45, Springer, 2016.

[16] K. S. Adewole, T. Han, W. Wu, H. Song and A. K. Sangaiah, "Twitter Spam Account Detection Based on Clustering and Classification Methods," The Jour. of Supercomputing, vol. 76, pp. 4802-4837, 2020.

[17] L. Zhang and B. Liu, "Sentiment analysis and opinion mining," Encyclopedia of Machine Learning and Data Mining, pp. 1152-1161, 2017.

[18] B. Liu, Sentiment Analysis: Mining Opinions, Sentiments and Emotions, Cambridge Uni. Press, 2020.

[19] S. Ahmed and A. Danti, "Effective Sentimental Analysis and Opinion Mining of Web Reviews Using Rule Based Classifiers," Proc. of Computational Intelligence in Data Mining, vol. 1, pp. 171-179, Springer, 2016.

[20] F. Hemmatian and M. K. Sohrabi, "A Survey on Classification Techniques for Opinion Mining and Sentiment Analysis," Artificial Intelligence Review, vol. 52, pp. 1495-1545, 2019.

[21] E. Cambria, D. Das, S. Bandyopadhyay and A. Feraco, "Affective Computing and Sentiment Analysis," Proc. of a Practical Guide to Sentiment Analysis, pp. 1-10, Springer, 2017.

[22] M. Ptaszynski, R. Rzepka, K. Araki and Y. Momouchi, "Automatically Annotating a Five-billion-word Corpus of Japanese Blogs for Sentiment and Affect Analysis," Computer Speech & Language, vol. 28, pp. 38-55, 2014.

[23] E. Cambria, Y. Li, F. Z. Xing, S. Poria and K. Kwok, "SenticNet 6: Ensemble Application of Symbolic and Subsymbolic AI for Sentiment Analysis," Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pp. 105-114, DOI: 10.1145/3340531.3412003, 2020.

[24] E. Cambria, S. Poria, A. Gelbukh and K. Kwok, "Sentic API: A Common-sense Based API for Concept-level Sentiment Analysis," Proc. of Making Sense of Microposts (# Microposts2014), p. 2, [Online], Available: https://hdl.handle.net/10356/84835, 2014.

[25] A. Jeyapriya and C. K. Selvi, "Extracting Aspects and Mining Opinions in Product Reviews Using Supervised Learning Algorithm," Proc. of the 2nd IEEE International Conference on Electronics and Communication Systems (ICECS), pp. 548-552, Coimbatore, India, 2015.

[26] A. Tripathy, A. Agrawal and S. K. Rath, "Classification of Sentimental Reviews Using Machine Learning Techniques," Procedia-Computer Science, vol. 57, pp. 821-829, 2015.

[27] C. Alfaro, J. Cano-Montero, J. Gómez, J. M. Moguerza and F. Ortega, "A Multi-stage Method for Content Classification and Opinion Mining on Weblog Comments," Annals of Operations Research, vol. 236, pp. 197-213, 2016.

[28] A. Hussain and E. Cambria, "Semi-supervised Learning for Big Social Data Analysis," Neurocomputing, vol. 275, pp. 1662-1673, 2018.

[29] N. Claypo and S. Jaiyen, "Opinion Mining for Thai Restaurant Reviews Using K-Means Clustering and MRF Feature Selection," Proc. of the 7th IEEE International Conference on Knowledge and Smart Technology (KST), pp. 105-108, Chonburi, Thailand, 2015.

[30] I. Al-Agha and O. Abu-Dahrooj, "Multi-level Analysis of Political Sentiments Using Twitter Data: A Case Study of the Palestinian-Israeli Conflict," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 5, no.3, pp. 195-215, 2019.

[31] S. Kumar, V. Koolwal and K. K. Mohbey, "Sentiment Analysis of Electronic Product Tweets Using Big Data Framework," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 5, no. 1, pp. 43-59, 2019.

[32] K. M. Nahar, A. Jaradat, M. S. Atoum and F. Ibrahim, "Sentiment Analysis and Classification of Arab Jordanian Facebook Comments for Jordanian Telecom Companies Using Lexicon-based Approach and Machine Learning," Jordanian Jour. of Comp. and Inf. Tech. (JJCIT), vol. 6, no.3, pp. 247-262, 2020.

[33] F. Degenhardt, S. Seifert and S. Szymczak, "Evaluation of Variable Selection Methods for Random Forests and Omics Datasets," Briefings in Bioinformatics, vol. 20, pp. 492-503, 2019.

[34] S. S. Rathore and A. Gupta, "A Comparative Study of Feature-ranking and Feature-subset Selection Techniques for Improved Fault Prediction," Proceedings of the 7th India Software Engineering Conference, pp. 1-10, Chennai, India, 2014.

[35] Z. Xu, J. Liu, Z. Yang, G. An and X. Jia, "The Impact of Feature Selection on Defect Prediction Performance: An Empirical Comparison," Proc. of the IEEE 27th International Symposium on Software Reliability Engineering (ISSRE), pp. 309-320, Ottawa, Canada, 2016.

[36] M. B. Kursa and W. R. Rudnicki, "Feature Selection with the Boruta Package," J. Stat. Softw., vol. 36, pp. 1-13, 2010.

[37] A. O. Balogun, S. Basri, S. J. Abdulkadir, V. E. Adeyemo, A. A. Imam and A. O. Bajeh, "Software Defect Prediction: Analysis of Class Imbalance and Performance Stability," Journal of Engineering Science and Technology, vol. 14, pp. 3294-3308, 2019.

**APPENDIX:** (a) Selected features using CFS + Boruta and (b) Selected features using CFS + RFE.

| (a) Dataset/No. of features | Features |
|---|---|
| Amazon (23) | awesom beauti best charm comfort disappoint excel fine flawless good great happier love nice perfect poor price rock seller setup sturdy wast well |
| Yelp (25) | amaz awesom bad bread delici delight don't excel fantast friend fun good great happi incred love minut mouth nice outstand perfect town wasn't white wonder |
| IMDB (25) | actual amaz bad beauti best brilliant cast cinema enjoy entertain excel film funni great love nice perform play portray right stupid terribl wast will wonder |
| Kaggle (44) | attent best carlyfiorina carson democraticdeb don't character enjoy favorit fox goldietaylor gopdeb great httptcospzaa imwithhuck jeb job johnkasich just kasich look love lrihendri marcorubio mostretweet nail rate reaction realbencarson realdonaldtrump realli rubio rwsurfer girl superman hotmal tedcruz thank transcript truth until via vine winner women won |

| (b) Dataset/No. of features | Features |
|---|---|
| Amazon (3) | great good price |
| Yelp (7) | great good love delici don't friend amaz |
| IMDB (70) | bad film love great even play wonder wast best enjoy cast didn't excel stupid beauti bore funni terribl perform will worst portray wors hour job amaz suck start actual disappoint cinema role right mess avoid lack subtl poor brilliant cool horribl histori fail cheap annoy ridicul hole nice crap fine lame |
| Kaggle (22) | rwsurfergirl character realdonaldtrump fox tedcruz rubio gopdeb thank look jeb just job don't rate great carlyfiorina best love lrihendri carson enjoy women |

## ملخص البحث:

اجتذب تحليل العواطف حديثاً اهتماماً ملحوظاً في السّنوات الأخيرة؛ بسبب ما له من قابليّة للتّطبيق في تحديد آراء المستخدمين وعواطفهم وانفعالاتهم من مجموعات بياناتٍ ضخمة تحتوي على كمّ هائل من النّصوص. ويتركّز الهدف من التّحليل العاطفي على تحسين خبرة المستخدمين عن طريق توظيف تقنيات متينة يمكنها التّنقيب عن الآراء والعواطف من مجموعات البيانات الضّخمة. وهناك دراسات متعددة تناولت التّحليل العاطفي والتّنقيب عن الآراء من المعلومات النصية. ومع ذلك، فإنّ وجود كلماتٍ خاصةٍ بمجالٍ ما دون غيره؛ إلى جانب اللّهجات العاميّة والاختصارات والاخطاء القواعدية، فرض تحدّيات جدّية إضافية على طرق التّحليل العاطفي القائمة. في هذه الورقة، نركّز على تحديد مجموعة فرعيّة مميزة فعّالة من المميزات التي تساعد في تصنيف آراء المستخدمين من مجموعات البيانات الضّخمة. تقترح هذه الدراسة إطاراً هجيناً لانتقاء المميزات مبنياً على تهجين طريقة اختيار المميزات القائمة على الفلترة وطريقة اختيار المميزات القائمة على الحَجْب. ويستخدم انتقاء المميزات المستند الى الارتباط (CFS)، المهجّن مع بوروتا (Boruta) وإزالة المميزات الثانوية (RFE) من أجل تحديد المجموعات الفرعيّة من المميزات الأكثر تمييزاً للتّحليل العاطفي. وقد تمّ اعتبار أربعٍ من مجموعات البيانات المتاحة للعموم هي: (Amazon، Yelp، و IMDB، وKaggle) من أجل تقييم أداء إطار انتقاء المميزات الهجين المقترح. وتعمل هذه الدراسة على تقييم أداء ثلاث خوارزميات تصنيف هي: (SVM، و NB، وRF) للتحقق من تفوُّق النّظام المقترح. وبينت نتائج التجارب العملية المجراة على مجموعات البيانات والخوارزميات المذكورة آنفاً، أن انتقاء المميزات المستند الى الارتباط (CFS) المستخدم مع بوروتا (Boruta) نتجت عنه نتائج واعدة وخصوصاً عندما يتم تمرير المميزات المختارة الى خوارزمية (RF). وفي الحقيقة، فإن الإطار الهجين المقترح يقدم طريقةً فعالةً لتوقُّع آراء المستخدمين وعواطفهم مع اعتبارٍ أساسيٍ لدقة التوقُّع. وتجدر الإشارة الى أنّ زمن الحساب للنموذج الناتج أقصر نتيجةً لإطار انتقاء المميزات للهجين المقترح.

# ENHANCING COLLABORATIVE FILTERING RECOMMENDATION USING REVIEW TEXT CLUSTERING

Ayman S. Ghabayen[1] and Basem H. Ahmed[2]

*(Received: 6-Jan.-2021, Revised: 5-Mar.-2021, Accepted: 27-Mar.-2021)*

## ABSTRACT

*The enormous rapid growth of the online world and universal computing brought a wide range of choices for Internet users to obtain information of interest. However, the huge amount of new information released every day in "big data" is greater than the human information processing capacity. As a result, it becomes harder and harder for users to obtain the required information quickly and they are also facing the problem of information overload. Collaborative Filtering (CF) systems play an important role in overcoming the information overload phenomenon by providing users with relevant information based on their preferences. CF is one of the best recommendation approaches that automate the process of the "word-of-mouth" paradigm. The most critical tasks in CF are finding similar users with similar preferences and then predicting user ratings to provide a personalized list of ranked items to the users. Previous studies have almost exclusively focused on these tasks separately to enhance the quality of recommendation. Nevertheless, we argue that these two tasks are not completely independent, but are part of an incorporated process. The purpose of this study is to propose a recommendation method that bridge the gap between the tasks of rating prediction and ranking to better grasp the best similar users to the target user by combing the advantage potential information of users review text clustering and user numerical ratings to enhance the CF recommendation methods proposed in the literature. The experimental results on three different datasets from Amazon show a considerable improvement over the baseline CF approaches in terms of recall, precision and F1-measure.*

## KEYWORDS

*Recommender systems, Collaborative filtering, Review text, Clustering, Top-N recommendation.*

## 1. INTRODUCTION

There has been significant growth in the digital world across the Internet in recent years. With a tremendous amount of information, it has become extremely difficult to decide with the wide-ranging of alternatives and suggestions provided to us every day. Recommender systems (RSs) can address the information overload issue, by suggesting users with recommendations of items, such as websites, movies, books, songs and music based on their individual historical preferences [1]-[2]. In e-commerce, many commercial websites, such as Amazon.com, eBay, Netflix, Yelp, last.FM, YouTube, etc. provide recommendation services. Also, in social media sites, recommender systems can help users annotate items with tags using tag recommendation, thus impeding more effective retrieval and classification in tagging systems [3].

This recommendation service could be a strategy to improve the relationship between commercial websites and their customers. Therefore, it is intended that a high-quality personalized recommendation service can ensure customers' satisfaction and loyalty [4]. CF approach is considered probably one of the most commonly applied and successful technology in RS [2], [5]. CF assumes that users who chose item A will be interested in item B if other users who chose item A were also interested in item B. CF matches the target user choices against other users to identify a group of 'like-minded', also known as 'nearest neighbour', users. This is typically done using metrics, such as cosine similarity or Pearson's correlation coefficient. Once the group of 'like-minded' users is identified, those items, which gain a high rate or are selected by the group top-N preferable items that a target user has not accessed, are then recommended [6].

Ordinary CF approaches depend on the commonality among users. Similar users or items are realized

1. A. S. Ghabayen is with the Department of Computer Science, University College of Science and Technology, KhanYounis, Palestine. E-mail: a.ghabayen@cst.ps
2. B. H. Ahmed is with the Department of Computer Science, Al-Aqsa University, Gaza, Palestine. Email: basem@alaqsa.edu.ps

by computing the similarities of common users with the active user in the rating items [5]. Generally, the CF recommender system performs well once there are sufficient user preferences. However, it suffers from certain limitations related to data sparsity and cold-start problems [7].

Data sparsity is considered one of the critical issues in collaborative filtering approaches [7]. In practice, many commercial recommender systems are based on a large dataset, where the number of items is always bigger compared to the number of users. Furthermore, most active users usually provide a rating for a rather restricted number of items. As a result, the user-item matrix used for collaborative filtering approaches could be extremely sparse, which makes it a challenge to make a useful recommendation. Moreover, data sparsity issue occurs in several situations and is specifically evident in a situation where a new user has just entered the system or a new item has just been added to the system, which is commonly known as 'cold start' problem [5], [7].

Typically, considering only the user rating data on items fails to completely indicate users' similarity for two reasons 1) ratings alone does not demonstrate the reason overdue to a user's rating and 2) users may rate items equally in the same way; however, their ratings may be based on different perspectives or item features.

To deal with the aforementioned problems, numerous approaches have been proposed by representing users and items with external knowledge resources, including user tags [5], [8], item contextual data [9] and use social data [10].

Nowadays, plenty of users often tend to provide their opinions on the Internet utilizing text. These reviews have the potential to provide a system with more details and efficient user preferences [11]. Putting it simply, user text reviews could be exploited, in combination with numerical ratings, to improve the word-of-mouth recommendation process.

This study aims to propose a recommendation method to better grasp the best alike users to the target user by combing the advantage potential information of review text clustering and user numerical ratings to enhance the CF recommendation methods proposed in literature works.

The remainder of this paper is structured as follows: Section 2 presents previous work related to CF. Section 3 then presents the proposed methods, details about the different steps which are performed, data pre-processing, standard collaborative filtering, similarity weighting, rating prediction, review clustering, item ranking and recommendation. Section 4 presents the dataset used, methodology and metrics used to examine the proposed approach and a variety of existing most common related CF recommendation algorithms to compare with. Section 5 presents the results and discussion. Lastly, Section 6 gives the overall conclusions of the work presented in this paper, as well as suggestions for future research to be performed in this field.

## 2. RELATED WORK

This section presents prior work related to recommender systems. The first part presents related work utilizing clustering algorithms in recommender systems and the second part presents related work exploiting user reviews.

### 2.1 Clustering-based Recommender Systems

Clustering is considered as an unsupervised method of grouping content based on some obvious features, such as words or word phrases in a set of documents. In simple words, clustering is the process of grouping patterns or entities into restricted classes of similar objects. In this case, a large volume of data is classified into similar groups (related instances into clusters) [12]. Clustering has been widely utilized in a wide range of disciplines, such as image segmentation [13], information retrieval and filtering [13], text mining [14] and many other real-world applications [15].

In CF, the review clustering process works either by identifying users into groups with similar item reviews or items into groups that have the same users' preferences. Thereby, when a target user is recognized as similar to a given cluster, then items associated with these users within the related cluster are recommended to the target user. There have been various methods proposed to enhance RS accuracy by utilizing clustering methods [16]-[19]. Wang et al. [17] proposed a clustering-based CF for dealing with data sparsity problems. They initially clustered users according to their rating preferences into k clustering through the K-means clustering algorithm. Then, they introduced a formula to determine the missing rating in the user-item rating matrix to obtain a high-density matrix. The new calculated rating is used to determine the similarity of items and estimate the rating of the active user on items that have not been rated. Sarwar et al. [18] proposed a clustering approach that groups users into clusters upon their numerical rating behaviour. They confirmed that using rating clustering shows promising improvement in the recommendation accuracy on the traditional CF. Z. Cui et al. [19] introduced a recommendation model based on a time correlation coefficient called TCCF. The proposed model clusters similar users together based on the user's interest over time. Their model provides a higher-quality recommendation. In [16], CF and content-based filtering approaches were performed through clustering to identify similar users and items, respectively; after that, a personalized recommendation to the active user was made. The clustering procedure has been realized as a successful way to enhance the recommendation accuracy compared to the basic CF approach [16], [18]-[19]. However, literature reviews show that the majority of previous studies clustered users or items individually and identified the similarity between users and items based on numerical rating data. This inspires us to propose a new CF approach by performing review clustering to group reviews into clusters and locate users into groups of clusters based on their reviewing behaviour on items, in order to improve the recommendation quality of current traditional CF. K-means clustering algorithm is one of the common algorithms utilized with model-based CF system [20]. K-means clustering does a very good job when the clusters have a kind of spherical shape. Nevertheless, this algorithm is highly dependent on the user-defined variants; i.e., the number of clusters from the data and the selection on the initial centroid need to be initialized. Subsequently, different variants lead to inaccurate recommendation quality. Arthur and Vassilvitskii [21] proposed an enhanced k-means clustering algorithm called K-means++. K-means++ randomly chooses the initial centroids, then determines the subsequent centroid using the proportional probability to the squared distance from its closet existing centroid. According to [21], this algorithm shows an improvement in the speed and accuracy of the k-means, in addition to its ability to automatically identify the optimal number of clusters. We use the K-means++ clustering algorithm in this study.

## 2.2 Review-based Recommender Systems

Through the last decade, there has been intensive research in RS and various approaches have been proposed to improve the RS reliability and accuracy through exploring knowledge from other sources. Examples are: Knowledge-based Systems [22]-[23], Internet of Things (IoT) [24], Information Retrieval Systems [25], User Tags [5] and Neural Networks [26]. Besides, information from customers' reviews can be exploited to provide accurate recommendations. Nilashi et al. [27] proposed a recommender system for e-tourism platforms. By utilizing the online reviews on social network sites, they applied supervised and unsupervised machine learning methods to analyze the customers' online reviews besides using multi-criteria ratings in building their recommender system. The evaluation results confirmed that the use of online reviews leads to precise recommendations.

Terzi et al. [28] modified the traditional CF approach by identifying the similarities between users using their reviews on items, as an alternative to numerical ratings. More properly, two users are considered similar if both of them co-reviewed an item. The new similarity scores are then used as a weight in the rating prediction stage in CF.

Musto et al. [29] offered a multi-criteria CF method that makes use of users' reviews to produce a multi-faceted representation of users' interests. Furthermore, sentiment analysis and opinion mining frameworks were applied to extract relevant aspects and sentiment scores from users' reviews.

Macdonald and Ounis [30] applied a weak supervision process at the data pre-processing phase to combine both implicit and explicit users' feedback. This process was focusing on bridging the gap between the tasks of item rating prediction and item ranking. The proposed approach achieved raising the representation of less popular items in the recommendation list. Accordingly, the results showed a comparable accuracy in terms of rating prediction and item ranking as compared to other methods.

Margaris et al. [31] investigated a venue recommendation system for social network users by considering user reviews' features related to the venues (e.g. service, price, atmosphere, physical distance), in addition to CF score which entails likeness of users' tastes. Later, the recommendation process was provided based on both explicit rating scores and implicit scores estimated by handling textual review features. Other researchers [32], [33] used topic modeling to identify hidden topics from users' reviews exploiting topics based on latent Dirichlet allocation (LDA) for generating the topic distribution profile of users.

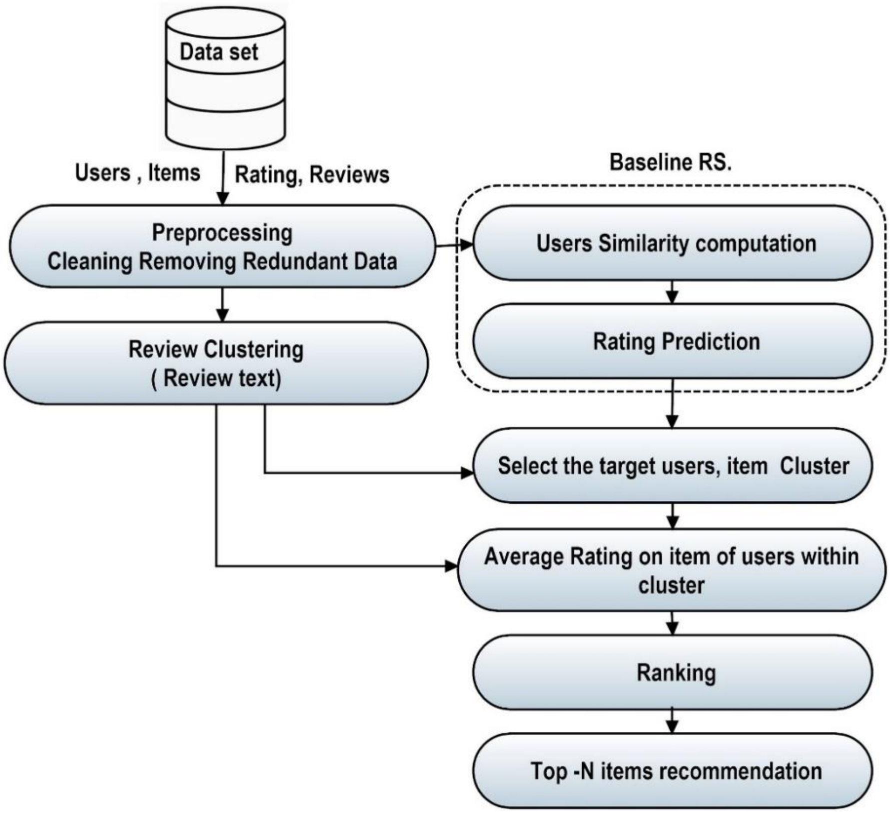



Figure 1. Proposed collaborative filtering framework.

## 3. PROPOSED METHOD

The main aim of this study is to propose a method to improve the recommendation accuracy by clustering users' review texts and integrating them with user ratings. Figure 1 shows the proposed collaborative filtering framework. The following subsection presents the proposed method step in more detail.

### 3.1 Data Pre-processing

The pre-processing step aims to refine the review text from parts that decrease the efficiency of the clustering and recommendation processes. To improve review clustering efficiency, several pre-processing steps have been made. First, redundant data and rows without review text are deleted. Next, all non-alphabetic characters, like emotion letters, smiles, finding punctuation, periods, hyphens and stop words, are eliminated from each review text. Then, the stems (roots) are identified and upper-case letters are converted into lower-case letters in the words in each review. Conversely, to avoid cold-start and sparsity problems in the recommender system, users who have fewer than 5 reviews or ratings are filtered out.

### 3.2 Standard Collaborative Filtering

Traditional approaches use the entire user-item database to identify the so-called “neighbourhood” of a new user or new item. Based on the neighbourhood distance or the correlation between two users or items, each neighbour receives a weight and then, the algorithm in some manner aggregates the preferences of the neighbours to produce a prediction or recommendation for the new user or (target user) [34]. Hence, when the task is to produce Top-N recommendations, these approaches tend to find the most similar (nearest neighbours) users or items. Because such an approach makes a prediction based on local similar users (neighbourhood) of the target user or similarities between items, it is commonly classified into user-based and item-based approaches [34]-[36].

The user-based collaborative filtering assumes that users who chose item A will be interested in item B if other users who chose item A are also interested in item B. On the other hand, item-based collaborative filtering looks at each item on the target user list of the chosen items and identifies other items that seem to be 'similar' to that item. The similarity of items depends on the closely matching attributes with the previously rated items by the target user.

### 3.3 Similarity Weighting

The most commonly used methods to calculate similarity among the two users $u$ and $v$ are the cosine-based and correlation-based similarity measures [37]. The similarity between user $u$ and $v$ is measured by calculating the cosine angle between users' corresponding rating vectors $u = (r_{n,1}, \dots, r_{n,N})$ and $v = (r_{k,1}, \dots, r_{k,N})$ defined as follows:

$$\cos(\theta) = \frac{\vec{u} \cdot \vec{v}}{\|\vec{u}\| \cdot \|\vec{v}\|} \tag{1}$$

Using Equation (1), the cosine similarity measure and the Pearson correlation coefficient between users $u$ and $v$ are defined respectively as follows:

$$\text{Sim}\ (u, v) = \frac{\sum_{i_m \in I_{u,v}} r_{u,m} * r_{v,m}}{\sqrt{\sum_{i_m \in I_{u,v}} r^2{}_{n,m}} * \sqrt{\sum_{i_m \in I_{u,v}} r^2{}_{v,m}}} \tag{2}$$

$$\text{Sim}(u, v) = \frac{\sum_{i_m \in I_{n,k}} (r_{n,m} - \bar{r}_n) * (r_{k,m} - \bar{r}_k)}{\sqrt{\sum_{i_m \in I_{n,k}} (r_{n,m} - \bar{r}_n)^2} * \sqrt{\sum_{i_m \in I_{n,k}} (r_{k,m} - \bar{r}_k)^2}} \tag{3}$$

where $I_{n,k}$ denotes the co-rated items between users $u$ and $v$. In other words, it denotes items rated by both users.

### 3.4 Rating Prediction

After computing the similarities between the target user/active user and the other users, k-nearest neighbours to the target user are identified. Generally, the CF estimates the rating of the unseen item for the target user based upon item rating from those k-nearest neighbours. As mentioned earlier, CF generates predictions based on the entire set of those items that have been rated or chosen by the target user. More specifically, the gain of the utility function $\hat{r}(u, i)$ of item $i \in I$ for user $u \in U$ is computed as an aggregate of the ratings $r_v$ of most similar users for user $u$ on item i. The utility function is defined as follows:

$$\hat{r}(u, i) = \bar{r}_u + \frac{\sum_{v \in \hat{U}} \text{sim}(u, v) * (r_v - \bar{r}_v)}{\sum_{v \in \hat{U}} |\text{sim}(u, v)|} \tag{4}$$

where $\bar{r}_u$ denotes the average rating of user $u$ and $\bar{r}_v$ the average rating of user $v$. The average rating $\bar{r}_v$ is defined as: $\bar{r}_v = \frac{1}{|I_v|} \sum_{i_m \in I_v} r_{v,m}$, where $I_{v,m} = \{i_m \in I \,|\, r_{v,m} \neq 0\}$.

### 3.5 Review Clustering

In this step, we conduct the user review clustering process for identifying clusters, each of which is composed of a group of users or items who/that possess similar reviewing preferences among each other. Hence, this process groups users or items into clusters, thus giving a new way to identify the neighbourhood similarities of users or items in the CF recommender system.

As mentioned earlier, the objective of this research is to propose a recommendation method that combines the explicit review text data with the implicit user rating data in CF recommendations. In more detail, the Top-N recommendation list is re-ranked according to the similarity between target user/item within a related cluster, through clustering users' review data using K-means++. In this situation, an appropriate decision is made with which items might be recommended or not based on like-minded users within the cluster to improve the quality of CF recommendation. In this stage, K-means++ is applied to cluster user/item reviews.

Thereby, to cluster user/item review text, first, we need to convert the text of the free-form reviews into structure data. This means to convert the text data into numerical values. This process is sometimes referred to as "vectorization". Among the popular vectorization processes is the term frequency-inverse document frequency (TF-IDF) measure [38]-[39]. Within the context of RS, the main idea of the TF_IDF measure is to estimate how important a keyword is to an item; the more occurrence of a keyword in a document, the more important it is. However, it also considers frequent terms that appear in many items and are not very relevant. Concretely, TF-IDF works as follows. Let $\mathrm{N}$ be the whole set of available documents that can be recommended to the user $\mathrm{u_n}$ and let $\mathrm{N_k}$ be the number of the documents in which the term $\mathrm{t_k}$ appears. First, the frequency $\mathrm{f_{k,m}}$ of each term occurring in the document $\mathrm{i_m} \in \mathrm{I}$ is counted. Note that if the term $\mathrm{t_k}$ does not appear in the text of the document $\mathrm{i_m}$, then $\mathrm{f_{k,m}} = 0$. The term frequency of each term $\mathrm{t_k}$ of the document $\mathrm{i_m}$ is computed as follows:

$$\mathrm{TF_{t_k m}} = \frac{\mathrm{f_{t_k,m}}}{\max \mathrm{f_m}} \tag{5}$$

where $\max \mathrm{f_m}$ indicates the maximum term frequency of all terms that appear in the document $\mathrm{i_m}$. In the TF measure, the more occurrence of a term in a specific document, the more important it is. However, considering terms that appear frequently in many documents tends to be less useful to determine whether the documents are relevant or irrelevant. On the other side, the inverse document frequency measure IDF is used to consider the influence of a given term in the entire collection of available documents. The IDF is regarded as a measure that minimizes the weight of terms that frequently appear in most documents, such as stop-words. Formally, IDF of the term $\mathrm{t_k}$ is computed as follows:

$$\mathrm{IDF_{t_k}} = \log \frac{\mathrm{N}}{\mathrm{N_k}} \tag{6}$$

Finally, the TF-IDF measure for the term $\mathrm{t_k}$ in a document $\mathrm{i_m}$ is defined as the combination of term-frequency and inverse document frequency [38], which is formally defined as follows:

$$\mathrm{TF - IDF(t_k, i_m)} = \mathrm{TF_{t_k,m}} \times \mathrm{IDF_{t_k}} \tag{7}$$

This method can be used to obtain terms frequently occurring in users' review text. Subsequently, using the TF-IDF, we can find out exactly what terms are important in each review. This step identifies the features of each review. Hence, the classification of reviews by TF-IDF value leads to finding a group of reviews with similar subject areas according to the importance of terms [40]. This is the reason why this research utilizes the K-means++ clustering algorithm to cluster users' reviews based on review topics. The K-means++ algorithm determines a center of the cluster that comprises a group of reviews with a specific topic and then assigns a review to a cluster based on the highest cosine similarity between the TF-IDF value of the review and the center value of each cluster. Afterwards, each review is associated with the corresponding reviewers and items.

### 3.6 Item Ranking and Recommendation

In Standard CF, the item ranking and recommendation step comes after the rating prediction of items that have not been evaluated by the target user. The items are ranked based upon the predicted rating values and then the Top-N items with the highest values are recommended to the target user for each recommendation interaction [41]. Accordingly, the goal of the Top-N recommendation is to obtain a list of the most relevant items allocated to user preferences. Different from the standard CF, the proposed approach recommends relevant items based on the result of review clustering and the preference propensity of each user by utilizing the estimated rating and user reviewing behaviour on items. As a result, either an incentive or a penalty is applied to each item in the Top-N recommendation list. Therefore, the Top-N list will be re-ranked based on users' reviewing preference clustering of users on items.

The proposed ranking method is demonstrated as follows:

- Assume that the standard CF decides whether to recommend a particular item ($\mathrm{i}$) to the target user ($\mathrm{u}$) or not. Normally, it checks whether the predicted rate $\hat{\mathrm{r}}(\mathrm{u}, \mathrm{i})$ is sufficiently large (i.e., $\hat{\mathrm{r}}$ >3, which means that the user likes the item), then item $\mathrm{i}$ will be recommended to the user $\mathrm{u}$.
- Under the ranking process based on review clustering, an incentive is given to item ( $\mathrm{i}$) to be recommended if the estimated rate of item ( $\mathrm{i}$) is greater than or equal to the average estimated rate of item ( $\mathrm{i}$) of users within the cluster related to the active user. Otherwise, item ( $\mathrm{i}$) will be dropped from the Top-N recommendation list.

## 4. EXPERIMENTS

In this section, different experiments are conducted over three real-world datasets obtained from Amazon to examine the performance of the CF recommendation accuracy after applying the users' review clustering. All the experiments were run on a machine equipped with Intel Xeon CPU family 6, model85, CPU MHz 2000.180 and 13,021GB of RAM. The programming language used is Python 3.7.

### 4.1 Datasets

The proposed approach was evaluated over a real-world dataset collected by Amazon.com. In this study, 3 different dataset categories are selected for the experiment. The datasets are available at https://s3. amazonaws.com/amazon-reviews-pds/tsv/index.txt. The Amazon dataset includes product review of user reviews on each product and metadata including a numerical rating scale (1-5 stars) which indicates the user's opinion. Hence, a low rating illustrates a negative opinion, while in contrast, a high rating illustrates an incredibly positive opinion. Because of the vast size of the data, it's a challenge to handle it all. Therefore, the RS is built using a dataset of 3 product categories with the largest number of reviews; namely, Books, Video DVD and Wireless. Table 1 presents the description of Amazon dataset categories.

Table 1. Description of Amazon dataset categories.

| Dataset | #Users | #Items | #Reviews |
|---|---|---|---|
| Books | 4,608044 | 2,264749 | 9,292094 |
| Video DVD | 2,071004 | 2,97525 | 4,622722 |
| Wireless | 5,193777 | 9,06086 | 8,110757 |

### 4.2 Methodology and Metrics

To assess the efficiency of our proposed approach, we applied the so-called back-testing strategy, which is well known in RS evaluation. The first step was to split the dataset into 5 folds. As a result, 20% was used as testing data and 80% was used as training data. The second step was dividing each user profile into 5 folds, such that 20 % of the items are being used as testing data, while the remaining items formed the training data. This step will guarantee that the recommendation process is not biased to a certain test/ training. Besides, it also guarantees that the proposed approach produces equal recommendations for all users, not only for the most active users. Afterwards, the results were averaged over the five folds. The efficiency of the proposed approach was evaluated by using well-known standard evaluation metrics; namely, Precision, Recall and F-measure; these metrics evaluate how actually an RS can produce a highly accurate prediction for relevant items as follows:

$$\text{precision} = \frac{\text{tp}}{\text{tp+fp}} \tag{8}$$

$$\text{Recall} = \frac{\text{tp}}{\text{tp+fn}} \tag{9}$$

where $\text{tp}$ (True positive is the number of relevant items that are to be recommended and are recommended correctly, whereas $\text{fp}$ (False positive) is the number of non-relevant items that should have not been recommended.

$\text{tn}$ (True negative) indicates the number of non-relevant items that should not have been recommended and were not recommended to the user and $\text{fn}$ (False negative) is the number of relevant items that are to be recommended but are not recommended correctly. We also considered the F1-measure metric which measures the accuracy of the test. The F1-measure links both recall and precision with equal weights in a single value. This metric reflects the weighted harmonic mean of precision and recall. The F1-measure is indicated by the following equation:

$$\text{F1} - \text{meaure} = 2 \times \frac{\text{Re call} \times \text{Pr ecision}}{\text{Re call} + \text{Pr ecision}} \tag{10}$$

### 4.3 Comparisons

To evaluate the proposed approach performance, we implemented a variety of existing most common Baseline CF recommendation algorithms to compare with. We compare with the following baselines:

**KNN**: We implemented standard K-Nearest Neighbourhood CF (KNN hereinafter) [42]. This model matches the target user choices against other users to identify a group of neighbourhood users. This is typically done using similarity metrics, such as cosine or Pearson's correlation coefficient. Once the group of neighbourhood users is identified, those items which gain a high rate or are selected by the group are then recommended to the target user.

**SVD**: This method is one of the state-of-art model-based approaches; Standard Singular Value Decomposition (SVD hereinafter) [43]. The advantage of model-based SVD is that this model not only incorporates rating information of similar users, but also leads to obtain a rating of other users who are considered to be not similar. In this case, several users get to be predictors for other user preference events without any overlap of co-rated items. The missing user ratings are prefilling with the rating data statistics. More details on the computation of SVD are given in [44]-[45]. This method is known as a baseline predictor in several works in the literature [46]-[48].

**NMF**: Finally, we compare the proposed approach with CF based on Non-negative matrix factorization (NMF hereinafter) [49], this model is very similar to the SVD model and is based on the idea of rating matrix manipulation. This method reduces the dimensionality of the user-item rating matrix to a low-dimensional space and then calculates similarities between users in this space, which can enhance the recommendation efficiency. It differs from the standard SVD method in investigating a non-negative update procedure based on each feature parameter concerned instead of the entire feature matrices.

## 5. RESULTS AND DISCUSSION

In this section, we provide the results of our experiments concerning Top-N recommendation quality of CF. We tested each method for various values of the N-recommended items. We vary the value of N (N = 5, 10, 15, 20, 25 and 30) for each user in the test set, since in the real scenario, users tend to click on items with higher ranks. Regarding the size of the K-Nearest neighbourhood values of the standard KNN recommender system, we conducted several experiments to choose the optimal value of K. The accuracy of prediction used is the root mean square error (RMSE). RMSE computes the mean value of the differences between the actual value and the predicted value of user rating. The RMSE is given by the following equation:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(r_i - \hat{r}_i)^2} \tag{11}$$

where, $r_i$ is the actual rating, $\hat{r}_i$ is the predicted rating and n is the number of ratings.

Table 2 shows the accuracy results of RMSE *versus* the size of the K-Nearest neighbourhood. As can be seen, on the three datasets, the RMSE values decrease when the size of the neighbourhood is increased. However, the accuracy deteriorates with $k$ values higher than 20 and there is no significant change with $k$ values higher than this value. Therefore, we considered the value of similar neighbourhood users K to be equal to 20.

Table 2. Accuracy results (RMSE) *versus* the size of the neighbourhood.

| Dataset \ K | Books | Video DVD | Wireless |
|---|---|---|---|
| 5 | 0.975 | 1.417 | 1.422 |
| 10 | 0.947 | 1.331 | 1.315 |
| 15 | 0.934 | 1.245 | 1.244 |
| 20 | 0.933 | 1.191 | 1.244 |
| 25 | 0.934 | 1.191 | 1.244 |
| 30 | 0.933 | 1.192 | 1.244 |
| 35 | 0.933 | 1.192 | 1.244 |
| 40 | 0.934 | 1.193 | 1.244 |
| 45 | 0.933 | 1.192 | 1.244 |
| 50 | 0.934 | 1.193 | 1.243 |

The results on the three different datasets, using Recall, Precision and F1-Measure matrices, are summarized and discussed. Table 3 presents the results of the comparison between our proposed method and the baseline approaches on the Amazon Books dataset. Hence, the best performance for each metric is shown in bold as the proposed approach abbreviation (Prop hereinafter).

Table 3. Amazon Books dataset results.

| | | N=5 | N=10 | N=15 | N=20 | N=25 | N=30 |
|---|---|---|---|---|---|---|---|
| Recall | KNN | 51.1 | 77.9 | 89.2 | 94.0 | 96.3 | 97.4 |
| | NMF | 50.8 | 76.4 | 86.9 | 91.3 | 93.3 | 94.3 |
| | SVD | 50.7 | 76.7 | 87.6 | 92.5 | 94.7 | 95.7 |
| | Prop | **52.9** | **78.9** | **89.7** | **94.2** | **96.4** | **97.4** |
| Precision | KNN | 93.8 | 93.8 | 93.7 | 93.7 | 93.7 | 93.6 |
| | NMF | 94.3 | 94.0 | 94.0 | 93.9 | 93.9 | 93.9 |
| | SVD | 94.2 | 94.0 | 93.9 | 93.8 | 93.7 | 93.7 |
| | Prop | **98.1** | **96.0** | **94.9** | **94.4** | **94.1** | **94.0** |
| F1 | KNN | 66.2 | 85.1 | 91.4 | 93.8 | 95.0 | 95.5 |
| | NMF | 66.0 | 84.3 | 90.3 | 92.6 | 93.6 | 94.1 |
| | SVD | 65.9 | 84.5 | 90.6 | 93.1 | 94.2 | 94.7 |
| | Prop | **68.7** | **86.6** | **92.2** | **94.3** | **95.2** | **95.7** |

The results in Table 3 demonstrate that the proposed approach performs better than baseline approaches in terms of Recall, Precision and F1-Measure for different values of Top-N recommendation. The proposed approach achieves the best performance when N=5. In terms of Recall, an improvement from 1.8% to 2.2 % is noticed compared to the baseline approaches. On the other hand, there is an improvement from 3.8% to 4.3 % in terms of Precision. In terms of the overall performance regarding F1-measure, the proposed approach achieves an improvement from 2.5% to 2.8%. Furthermore, it is observed that the progress of improvement has an inverse relation to the value of N. For example, the progress when N = 5 is larger than those when N = 30, for the baseline approaches and the proposed approach. This situation is due to that the most relevant items related to the target user are involved in the recommendation of Top-N values. Hence, the proposed approach can accomplish higher progress at smaller N values. The results in Table 3 demonstrate that the proposed approach is completely appropriate in a real-life scenario, since users are normally attracted first to a few number of high-ranked items [50].

We conducted two more experiments on other categories of the Amazon dataset named Video DVD dataset and Amazon Wireless dataset to study the performance of the proposed approach on other datasets with variant numbers of users, items and reviews. Table 4 and Table 5 present the result on the Amazon Video DVD dataset and Amazon Wireless dataset, respectively.

Table 4. Amazon video DVD dataset results.

| | | N=5 | N=10 | N=15 | N=20 | N=25 | N=30 |
|---|---|---|---|---|---|---|---|
| Recall | KNN | 81.6 | 89.1 | 91.3 | 92.2 | 92.8 | 93.1 |
| | NMF | 81.2 | 87.4 | 88.9 | 89.6 | 89.9 | 90.1 |
| | SVD | 81.7 | 88.2 | 89.7 | 90.5 | 90.8 | 91.1 |
| | Prop | **84.0** | **90.4** | **92.0** | **92.6** | **93.0** | **93.2** |
| Precision | KNN | 91.1 | 91.1 | 91.0 | 91.1 | 91.1 | 90.9 |
| | NMF | 91.6 | 91.3 | 91.3 | 91.3 | 91.3 | 91.2 |
| | SVD | 91.4 | 91.1 | 91.1 | 91.0 | 91.0 | 91.0 |
| | Prop | **93.2** | **91.8** | **91.5** | **91.3** | **91.2** | **91.1** |
| F1 | KNN | 86.1 | 90.1 | 91.2 | 91.7 | 91.9 | 92.0 |
| | NMF | 86.1 | 89.3 | 90.1 | 90.4 | 90.6 | 90.7 |
| | SVD | 86.3 | 89.6 | 90.4 | 90.7 | 90.9 | 91.1 |
| | Prop | **88.4** | **91.1** | **91.7** | **92.0** | **92.1** | **92.1** |

From the results in Table 4 and Table 5, we can see that the proposed approach outperforms the other baseline approaches in terms of recall, precision and F1-measure. In the case of smaller values of N, as mentioned earlier, the smaller value of N indicates a larger improvement in user satisfaction.

Table 5. Amazon Wireless dataset results.

| | | N=5 | N=10 | N=15 | N=20 | N=25 | N=30 |
|---|---|---|---|---|---|---|---|
| Recall | KNN | 88.1 | 91.3 | 92.4 | 92.7 | 93.0 | 93.0 |
| | NMF | 87.6 | 89.6 | 90.1 | 90.1 | 90.1 | 90.0 |
| | SVD | 88.6 | 90.8 | 91.2 | 91.3 | 91.3 | 91.4 |
| | Prop | **90.6** | **92.6** | **93.2** | **93.1** | **93.2** | **93.1** |
| Precision | KNN | 87.3 | 87.7 | 87.5 | 87.7 | 87.6 | 87.5 |
| | NMF | 87.8 | 87.9 | 87.8 | 87.9 | 87.8 | 87.8 |
| | SVD | 87.9 | 87.8 | 87.8 | 87.7 | 87.7 | 87.8 |
| | Prop | **88.3** | **87.9** | **87.8** | **87.9** | **87.8** | **87.8** |
| F1 | KNN | 87.7 | 89.5 | 89.9 | 90.1 | 90.2 | 90.2 |
| | NMF | 87.7 | 88.7 | 88.9 | 89.0 | 88.9 | 88.9 |
| | SVD | 88.2 | 89.3 | 89.4 | 89.5 | 89.5 | 89.6 |
| | Prop | **89.5** | **90.2** | **90.4** | **90.4** | **90.4** | **90.4** |

Finally, to examine whether the results obtained are statistically significant, a significance analysis was conducted in the form of a t-test for the proposed approach and the baseline approaches in terms of the F1-measure matrices. Hence, the F1-measure conveys the performance balance between both recall and precision. Table 6 presents the t-test results. As seen, the values of sig. (2-tailed) is less than 0.05. This ends in that the proposed approach presents a significant improvement when compared to the mentioned baseline approaches.

Table 6. T-test results on F1-meausre.

| | | | Mean | Std. Deviation | t | Sig. (2-tailed) |
|---|---|---|---|---|---|---|
| Amazon Books | Pair 1 | KNN | 87.8333 | 11.25996 | 2.584 | **0.049** |
| | | Prop | 88.7833 | 10.38488 | | |
| | Pair 2 | NMF | 86.8167 | 10.81026 | 10.808 | **0.0005** |
| | | Prop | 88.7833 | 10.38488 | | |
| | Pair 3 | SVD | 87.1667 | 11.06882 | 5.525 | **0.003** |
| | | Prop | 88.7833 | 10.38488 | | |
| Amazon Video DVD | Pair 1 | KNN | 90.5 | 2.26539 | 2.161 | 0.083 |
| | | Prop | 91.2333 | 1.43898 | | |
| | Pair 2 | NMF | 89.5333 | 1.75575 | 12.912 | **0.0005** |
| | | Prop | 91.2333 | 1.43898 | | |
| | Pair 3 | SVD | 89.8333 | 1.8085 | 9.037 | **0.0005** |
| | | Prop | 91.2333 | 1.43898 | | |
| Amazon Wireless | Pair 1 | KNN | 89.6 | 0.96747 | 2.471 | **0.05** |
| | | Prop | 90.2167 | 0.36009 | | |
| | Pair 2 | NMF | 88.6833 | 0.4916 | 27.49 | **0.0005** |
| | | Prop | 90.2167 | 0.36009 | | |
| | Pair 3 | SVD | 89.25 | 0.5244 | 13.521 | **0.0005** |
| | | Prop | 90.2167 | 0.36009 | | |

## 6. CONCLUSION

Nowadays, plenty of users often tend to provide their opinions on the Internet utilizing text. These heterogeneous recommending information sources beyond user rating data present opportunities and issues for traditional CF recommender systems. In this paper, we proposed a new CF approach by utilizing review text clustering and using numerical ratings. The proposed approach aims to recommend items to a target user based on the results of review clustering and the preference tendency of each user using the predicted rating and the users' reviewing behaviour on items. In such a case, the proposed approach bridges the gap between the ranking and the prediction tasks of recommender systems, in order to better grasp the best similar users to the target user, which leads to efficiently enhance the CF Top-N recommendation. The experimental results on three different datasets show a considerable improvement over the baseline CF approaches using just user explicit rating in terms of recall, precision and F1-measure.

## 7. FUTURE WORK

Our future work in this area will focus on studying other clustering algorithms, in addition to the complexity of the proposed approach. Another possible direction will focus on exploring and exploiting other alternative text features in reviews.

## REFERENCES

[1] X. Su and T. M. Khoshgoftaar, "A Survey of Collaborative Filtering Techniques," Advances in Artificial Intelligence, vol. 2009, Article ID: 421425, DOI: 10.1155/2009/421425, 2009.

[2] J. Bobadilla, F. Ortega, A. Hernando and A. Gutiérrez, "Recommender Systems Survey," Knowledge-based Systems, vol. 46, pp. 109-132, 2013.

[3] I. AlAgha and Yousef Abu-Samra, "Tag Recommendation for Short Arabic Text by Using Latent Semantic Analysis of Wikipedia," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 06, no. 02, pp. 165-181, 2020.

[4] K. Tarnowska, Z. W. Ras and L. Daniel, "Customer Loyalty Improvement," Proc. of Recommender System for Improving Customer Loyalty Book, pp. 7-11, Cham: Springer International Publishing, 2020.

[5] A. S. Ghabayen and S. M. Noah, "Using Tags for Measuring the Semantic Similarity of Users to Enhance Collaborative Filtering Recommender Systems," International Journal on Advanced Science, Engineering and Information Technology, vol. 7, no. 6, pp. 2063-2070, 2017.

[6] J. L. Herlocker, J. A. Konstan and J. Riedl, "Explaining Collaborative Filtering Recommendations," Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work, pp. 241-250, DOI: 10.1145/358916.358995, Philadelphia, Pennsylvania, USA, 2000.

[7] S. Natarajan, S. Vairavasundaram, S. Natarajan and A. H. Gandomi, "Resolving Data Sparsity and Cold Start Problem in Collaborative Filtering Recommender System Using Linked Open Data," Expert Systems with Applications, vol. 149, p. 113248, 2020.

[8] L. Al Qadi, H. El Rifai, S. Obaid and A. Elnagar, "A Scalable Shallow Learning Approach For Tagging Arabic News Articles," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 06, no. 03, pp. 263-280, 2020.

[9] G. Alshammari, J. L. Jorro-Aragoneses, N. Polatidis, S. Kapetanakis, E. Pimenidis and M. Petridis, "A Switching Multi-level Method for the Long Tail Recommendation Problem," Journal of Intelligent & Fuzzy Systems, vol. 37, pp. 7189-7198, 2019.

[10] J.-H. Su, W.-Y. Chang and V. S. Tseng, "Effective Social Content-based Collaborative Filtering for Music Recommendation," Intelligent Data Analysis, vol. 21, no. S1, pp. S195-S216, 2017.

[11] H.-A. Nuria, "A Literature Review of Word of Mouth and Electronic Word of Mouth: Implications for Consumer Behavior," Frontiers in Psychology, Mini Review, vol. 8, p. 1256, DOI: 10.3389/fpsyg.2017.01256, July 2017.

[12] A. Ghosal, A. Nandy, A. K. Das, S. Goswami and M. Panday, "A Short Review on Different Clustering Techniques and Their Applications," Proc. of Emerging Technology in Modelling and Graphics, pp. 69-83, Part of the Advances in Intelligent Systems and Computing Book Series, vol. 937, Springer, 2020.

[13] T. Nicholls and J. Bright, "Understanding News Story Chains Using Information Retrieval and Network

Clustering Techniques," Communication Methods and Measures, vol. 13, no. 1, pp. 43-59, 2019.

[14] M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. D. Trippe and J. B. Gutierrez, "A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques," KDD Bigdas, Halifax, Canada, arXiv:1707.02919, 2017.

[15] K. G. Dhal, A. Das, S. Ray and J. Gálvez, "Randomly Attracted Rough Firefly Algorithm for Histogram Based Fuzzy Image Clustering," Knowledge-based Systems, vol. 216, p. 106814, DOI: 10.1016/j.knosys.2021.106814, 2021.

[16] L. U, Y. Chai and J. Chen, "Improved Personalized Recommendation Based on User Attributes Clustering and Score Matrix Filling," Computer Standards & Interfaces, vol. 57, pp. 59-67, 2018.

[17] Q. Wang, W. Cao and Y. Liu, "A Novel Clustering Based Collaborative Filtering Recommendation System Algorithm," Proc. of Advanced Technologies, Embedded and Multimedia, for Human-centric Computing, pp. 673-680, Part of the Lecture Notes in Electrical Engineering Book Series, vol. 260, Springer, Dordrecht, Netherlands, 2014.

[18] B. M. Sarwar, G. Karypis, J. Konstan and J. Riedl, "Recommender Systems for Large-scale E-commerce: Scalable Neighborhood Formation Using Clustering," Proceedings of the 5th International Conference on Computer and Information Technology (ICCIT), vol. 1, pp. 291-324, 2002.

[19] Z. Cui et al., "Personalized Recommendation System Based on Collaborative Filtering for IoT Scenarios," IEEE Transactions on Services Computing, vol. 13, no. 4, pp. 685-695, 2020.

[20] T. Mohammadpour, A. M. Bidgoli, R. Enayatifar and H. H. S. Javadi, "Efficient Clustering in Collaborative Filtering Recommender System: Hybrid Method Based on Genetic Algorithm and Gravitational Emulation Local Search Algorithm," Genomics, vol. 111, no. 6, pp. 1902-1912, 2019.

[21] D. Arthur and S. Vassilvitskii, "K-Means++: The Advantages of Careful Seeding," Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1027-1035, New Orleans, USA, 2007.

[22] V. Vijayakumar, V. Subramaniyaswamy, R. Logesh and A. Sivapathi, "Effective Knowledge Based Recommender System for Tailored Multiple Point of Interest Recommendation," International Journal of Web Portals (IJWP), vol. 11, no. 1, pp. 1-18, 2019.

[23] M. Dong, X. Zeng, L. Koehl and J. Zhang, "An Interactive Knowledge-based Recommender System for Fashion Product Design in the Big Data Environment," Information Sciences, vol. 540, pp. 469-488, 2020.

[24] A. Hassani, P. D. Haghighi, S. Ling, P. P. Jayaraman and A. Zaslavsky, "Querying IoT Services: A Smart Carpark Recommender Use Case," Proc. of the IEEE 4th World Forum on Internet of Things (WF-IoT), pp. 619-624, Singapore, 2018.

[25] A. Bellogín and A. Said, "Information Retrieval and Recommender Systems," Proc. of Data Science in Practice, pp. 79-96, Part of the Studies in Big Data Book Series, vol. 46, Cham: Springer International Publishing, 2019.

[26] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton and J. Leskovec, "Graph Convolutional Neural Networks for Web-scale Recommender Systems," Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 974-983, DOI: 10.1145/3219819.3219890, 2018.

[27] M. Nilashi, O. Ibrahim, E. Yadegaridehkordi, S. Samad, E. Akbari and A. Alizadeh, "Travelers Decision Making Using Online Review in Social Network Sites: A Case on TripAdvisor," Journal of Computational Science, vol. 28, pp. 168-179, 2018.

[28] M. Terzi, M. Rowe, M.-A. Ferrario and J. Whittle, "Text-based User-kNN: Measuring User Similarity Based on Text Reviews," Proc. of the International Conference on User Modeling, Adaptation and Personalization, pp. 195-206, Part of the Lecture Notes in Computer Science Book Series, vol. 8538, Cham, Springer International Publishing, 2014.

[29] C. Musto, M. d. Gemmis, G. Semeraro and P. Lops, "A Multi-criteria Recommender System Exploiting Aspect-based Sentiment Analysis of Users' Reviews," Proceedings of the 11th ACM Conference on Recommender Systems, DOI: 10.1145/3109859.3109905, Como, Italy, 2017.

[30] A. H. Jadidinejad, C. Macdonald and I. Ounis, "Unifying Explicit and Implicit Feedback for Rating Prediction and Ranking Recommendation Tasks," Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval, [Online], Available: https://doi.org/10.1145/3341981.3344225, Santa Clara, CA, USA, 2019.

[31] D. Margaris, C. Vassilakis and D. Spiliotopoulos, "Handling Uncertainty in Social Media Textual Information for Improving Venue Recommendation Formulation Quality in Social Networks," Social Network Analysis and Mining, vol. 9, no. 1, p. 64, 2019.

[32] D. Chehal, P. Gupta and P. Gulati, "Implementation and Comparison of Topic Modeling Techniques Based on User Reviews in E-commerce Recommendations," Journal of Ambient Intelligence and Humanized Computing, vol. 2020, pp. 1-6, DOI: 10.1007/s12652-020-01956-6, 2020.

[33] M. Saraswat, S. Chakraverty and A. Sharma, "Review-based Topic Distribution Profile for Recommender Systems," Proc. of Advances in Data Sciences, Security and Applications, pp. 433-443, Part of the Lecture Notes in Electrical Engineering Book Series, vol. 612, Springer, Singapore, 2020.

[34] J. B. Schafer, D. Frankowski, J. Herlocker and S. Sen, "Collaborative Filtering Recommender Systems," Proc. of the Adaptive Web, pp. 291- 324, Part of the Lecture Notes in Computer Science Book Series, vol. 4321, Springer-Verlag, 2007.

[35] K. Peffers, T. Tuunanen, M. A. Rothenberger and S. Chatterjee, "A Design Science Research Methodology for Information Systems Research," Journal of Management Information Systems, vol. 24, no. 3, pp. 45-77, 2007.

[36] D. W. Aha, D. Kibler and M. K. Albert, "Instance-based Learning Algorithms," Machine Learning, vol. 6, no. 1, pp. 37-66, 1991.

[37] D. Jannach, M. Zanker, A. Felfernig and G. Friedrich, Recommender Systems: An Introduction, Cambridge University Press, DOI: 10.1017/CBO9780511763113, 2010.

[38] G. Salton, Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer, Addison-Wesley Longman Publishing Co., Inc., 1989.

[39] R. Baeza-Yates and B. Ribeiro-Neto, Modern Information Retrieval, New York: ACM Press: Addison-Wesley Longman Publishing, 1999.

[40] A. M. D. E. Hassanein and M. Nour, "A Proposed Model of Selecting Features for Classifying Arabic Text," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 05, no. 03, pp. 275-290, 2019.

[41] M. Srifi, A. Oussous, A. A. Lahcen and S. Mouline, "Recommender Systems Based on Collaborative Filtering Using Review Texts: A Survey," Information, vol. 11, no. 6, p. 317, 2020.

[42] M. Deshpande and G. Karypis, "Item-based Top-N Recommendation Algorithms," ACM Transactions on Information Systems, vol. 22, no. 1, pp. 143-177, 2004.

[43] B. Sarwar, G. Karypis, J. Konstan and J. Riedl, "Application of Dimensionality Reduction in Recommender System-a Case Study," Proceedings of the ACM WebKDD 2000 Web Mining for E-Commerce Workshop, DOI: 10.21236/ada439541, 2000.

[44] Y. Koren, R. Bell and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," Computer, vol. 42, no. 8, pp. 30-37, 2009.

[45] D. Bokde, S. Girase and D. Mukhopadhyay, "Matrix Factorization Model in Collaborative Filtering Algorithms: A Survey," Procedia Computer Science, vol. 49, pp. 136-146, 2015.

[46] R. Barathy and P. Chitra, "Applying Matrix Factorization in Collaborative Filtering Recommender Systems," Proc. of the IEEE 6th International Conference on Advanced Computing and Communication Systems (ICACCS), pp. 635-639, Coimbatore, India, 2020.

[47] V. X. Chen and T. Y. Tang, "Incorporating Singular Value Decomposition in User-based Collaborative Filtering Technique for a Movie Recommendation System: A Comparative Study," Proceedings of the 2019 International Conference on Pattern Recognition and Artificial Intelligence (PRAI'19), pp. 12-15, DOI: 10.1145/3357777.3357782, 2019.

[48] Z. Xian, Q. Li, G. Li and L. Li, "New Collaborative Filtering Algorithms Based on SVD++ and Differential Privacy," Mathematical Problems in Engineering, vol. 2017, pp. 136-146, 2017.

[49] X. Luo, M. Zhou, Y. Xia and Q. Zhu, "An Efficient Non-negative Matrix-factorization-based Approach to Collaborative Filtering for Recommender Systems," IEEE Transactions on Industrial Informatics, vol. 10, no. 2, pp. 1273-1284, 2014.

[50] W. Niu et al., "Context-aware Service Ranking in Wireless Sensor Networks," Journal of Network and Systems Management, vol. 22, no. 1, pp. 50-74, 2014.

**ملخص البحث:**

لقد جلب النّمو السريع لعالم الإنترنت والحوسبة مدى واسعاً من الخيارات لمستخدمي الإنترنت للحصول على المعلومات ذات الاهتمام. ومع ذلك، فإنّ الكمّ الهائل من المعلومات الجديدة التي تنطلق كل يوم فيما يُعرف "بالبيانات الضّخمة" هو أكبر من قدرة الإنسان على معالجة المعلومات. ونتيجةً لذلك، يصبح الأمر أصعب فأصعب بالنسبة للمستخدمين أن يحصلوا على المعلومات المطلوبة بسرعة، وهم يواجهون أيضاً مشكلة الحِمل الزّائد من المعلومات. وتلعب أنظمة الفلترة التعاونيّة دوراً مهماً في التّغلُّب على ظاهرة الحِمل الزّائد من المعلومات عن طريق تزويد المستخدمين بالمعلومات ذات العلاقة بناءً على تفضيلاتهم. وتُعدّ الفلترة التعاونيّة من بين أفضل طُرق التّوْصِيَة التي تعمل على أتمتة العمليّة المتعلقة بنموذج "الكلمة المنطوقة".

إنّ أكثر المهمات حَسْماً في الفلترة التعاونيّة هي إيجاد مستخدمين متماثلين في تفضيلاتهم، ومن ثمّ توقُّع تصنيفات المستخدمين لإنشاء قائمة شخصيّة يتمّ فيها ترتيب البنود الواردة في نصوص المستخدمين ووضعها تحت تصرُّف المستخدمين.

لقد ركزت الدراسات السابقة بشكلٍ شبه حصريّ على تلك المهمات بشكلٍ منفصل عن بعضها البعض من أجل تحسين جودة التّوصية. إلّا أننا نرى أن المُهمتَيْن آنفَتَي الذّكر ليستا مستقلّتين تماماً، وإنّما هما جزء من عمليّة مُدْمَجة. من هنا، تهدف هذه الورقة الى إقتراح طريقة توصية تجسر الفجوة بين مّهمة تُوقع التّصنيفات ومّهمة التّرتيب من أجل فهمٍ أفضل للمستخدمين المماثلين للمستخدم الهَدَف، وذلك عن طريق تمشيط المعلومات المتعلقة بالنّصوص التي تصدر عن المستخدمين ووضْعها في مجموعاتٍ أو عناقيد في هيئة تصنيفات رقميّة لتحسين طُرق الفلترة التعاونيّة المقترحة في أدبيّات الموضوع. وقد أُجريت التجارب على ثلاثٍ من مجموعات البيانات التابعة لأمازون، وأسفرت النتائج عن تحسُّنٍ ملحوظ حققته الطريقة المقترحة مقارنةً بالطُرق القائمة.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 07, No. 02, June 2021.

# ASSOCIATIVE CLASSIFICATION IN MULTI-LABEL CLASSIFICATION: AN INVESTIGATIVE STUDY

Raed Alazaidah[1], Mohammed Amin Almaiah[2] and Mo'ath Al-Luwaici[3]

*(Received: 9-Mar.-2021, Revised: 24-Apr.-2021, Accepted: 5-May-2021)*

## *ABSTRACT*

*Multi-label classification (MLC) is a very interesting and important domain that has attracted many researchers in the last two decades. Several single-label classification algorithms that belong to different learning strategies have been adapted to handle the problem of MLC. Surprisingly, no Associative Classification (AC) algorithm has been adapted to handle the MLC problem, where AC algorithms have shown a high predictive performance compared with other learning strategies in single-label classification. In this paper, a deep investigation regarding utilizing AC in MLC is presented. An evaluation of several AC algorithms on three multi-label datasets with respect to five discretization techniques revealed that utilizing AC algorithms in MLC is very promising compared with other algorithms from different learning strategies.*

## *KEYWORDS*

*Prediction, Machine learning, Multi-label classification, Associative classification, Learning strategies.*

## 1. INTRODUCTION

Classification is a very interesting task in data mining that involves assigning the class label of an unseen instance as accurately as possible, based on a labeled historical training set [1]-[2].

In general, classification could be divided into three main types the first type of which is called binary classification and comprises only two class labels. The second type is called multi-class classification and comprises more than two class labels in a dataset. Both binary classification and multi-class classification have been known as a conventional single label classification [3]. In single-label classification, class labels are considered to be mutually exclusive; that is, each instance in the dataset is associated with only one class label [4]. The third type is called MLC. MLC does not assume labels in the dataset to be mutually exclusive and hence, an instance in a multi-label dataset could be associated with more than one class label at the same time [5].

MLC has several distinguishable features over single-label classification. First, class labels in MLC are not considered to be mutually exclusive as in single-label classification and hence, class labels in MLC do have some kind of correlations and dependencies [6]. Second, the problem search space of a single-label classification problem is quite limited when compared with the large problem search space of the MLC problem [7]. The problem search space of the MLC problem equals $n^q$, where $q$ represents the total number of the class labels in the dataset. On the other side, the problem search space of binary classification equals 2 and for multi-class classification equals $q$. Finally, the complexity of MLC is very high compared with the complexity of single label classification [8]-[9].

Two main approaches are being used to handle MLC. The first approach adapts a single-label classification algorithm to handle a multi-label dataset, while the second approach transforms the multi-label dataset into one single-label dataset or more and then, this approach applies one single-label classifier or more on the transformed datasets, where the outputs of the single-label classifiers on the transformed datasets are aggregated to form the final prediction [10]. Regardless of the approach being used to handle multi-label datasets, the choosing step of the single-label classification algorithm (base classifier) is crucial in determining the accuracy of the proposed MLC algorithm [11].

Many base classifiers have been utilized in MLC, whether by adapting the single-label base classifier to handle multi-label datasets or by applying them to the transformed versions of the multi-label dataset.

1. R. Alazaidah is with the AI Department, Faculty of Science and IT, Irbid National University. Email: raeddiab@yahoo.com
2. M. A. Almaiah is with the Department of Computer Networks and Communications, College of Computer Sciences and Information Technology, King Faisal University, Al-Ahsa, Saudi Arabia.
3. M. Al-Luwaici is with the Institute of Engineering Mathematics, Uni. Malaysia Perlis (UniMAP), Malaysia. Email: wese5@yahoo.com

These base classifiers follow several learning approaches, such as decision trees, neural networks, fuzzy-based learning, lazy learning, statistical learning, support machine learning and several other learning approaches.

Surprisingly, the AC approach, which has been proven to produce high accurate rules and has the ability of discovering hidden knowledge that could not be discovered by other learning strategies [12]-[13], has been weakly utilized in MLC. According to [14], no AC algorithm has the ability of generating multi-label rules and hence, no AC algorithm can handle the problem of MLC. Nevertheless, few research studies that attempted to handle the problem of MLC could be found in the literature. Unfortunately, most of these attempts could not be recognized as effective MLC algorithms, as explained in Section 2, part C.

Therefore, this paper is interested in investigating the applicability of the AC learning approach in solving the problem of MLC, either by adapting one of the AC algorithms to handle MLC problems or by utilizing AC algorithms in classifying the transformed versions of the multi-label datasets and then, aggregating the outputs of these classifiers to generate multi-label rules.

Specifically, this paper aims to meet two main objectives. The first is to evaluate several AC algorithms on three multi-label datasets, with respect to five discretization techniques. The evaluation procedure considers two criteria: accuracy metric and running time. The second objective is to compare the performance of the most promising AC algorithms with other algorithms from several learning strategies, based on the accuracy metric, to determine the applicability of AC in solving the problem of MLC.

The rest of the paper is organized as follows: Section 2 presents some related work, while Section 3 introduces the empirical analysis. Section 4 presents the conclusion and lists some possible future work.

## 2. LITERATURE REVIEW

In this section, a brief-yet comprehensive-overview of the MLC domain is presented in sub-section A. Also, a quick review of AC learning approach and the considered AC algorithms is introduced in sub-section B. Finally, attempts to utilize AC in MLC are presented in sub-section C.

### A. Overview of MLC

MLC is a general type of classification that allows the examples (instances) in a dataset to be associated with more than one class label at the same time [15], [6]. Hence, the goal in MLC is to learn a function from a set of instances, where each instance could be associated with one or more class labels [16].

MLC was motivated at first by text categorization and medical diagnosis [17]. Recently, more scholars have paid great attention toward the problem of MLC; due to its importance in real-world applications [18]. In many domains, where single-label classification failed to solve the classification problem, MLC did [19]. For example, single-label classification may tag an email message as either a *work* or a *research project* but not both, whereas the fact is, the message could be tagged as both *work* and *research project* simultaneously, which MLC does. Nowadays, MLC is increasingly required by modern applications, such as music categorization into emotions [20], semantic video annotation [21], direct marketing [22] and protein function classification [23].

Two main general approaches are being used to handle MLC problems. The first approach is called the Problem Transformation Method (PTM), while the second approach is called the Algorithm Adaptation Method (AAM) [24]. The former transforms a multi-label dataset into a single-label dataset by using different transformation methods, such as Least Frequent Label (LFL), Most Frequent Label (MFL) or by choosing any label randomly [24]. Then, any of the shelf single-label classifiers could be used to classify the transformed dataset [25]. The latter adapts a specific single-label classification algorithm to handle a multi-label dataset [26]. Using PTMs is preferable over using AAMs; because the former are simpler, more general and not domain-specific like AAMs [27].

The step of choosing the base classifier is vital in both PTMs and AAMs. In fact, in datasets with low cardinality, such as Scene, Genbase and Emotions, the accuracy of the base classifier highly affects the final accuracy of the multi-label prediction step. Also, the accuracy of the base classifier in the high-cardinality dataset, like Yeast and Emotions, highly affects the classification step by determining the prediction step of the other labels that have been discarded due to the transformation step.

Therefore, several single-label classification algorithms have been utilized in the domain of MLC as base classifiers, such as C4.5, KNN, PART and several other single-label classification algorithms. Surprisingly, very few research studies have utilized AC in MLC [14], [28]-[29]. The next sub-section briefly overviews the AC learning approach.

## B. Associative Classification

Associative Classification (AC) is a learning approach that integrates the task of mining association rules with the task of classification [30]. Recently, AC has attracted many researchers for two main reasons. First, AC is capable of producing higher accurate rules than other learning approaches. Second, AC generates rules that are easier to be understood by the different types of users [31]. Thus, several classification algorithms have been proposed under the AC approach of learning. Even though these AC-based algorithms have shown high predictive performance in conventional single-label classification problems, unfortunately, they have never been adapted to handle a MLC problem [14].

In general, any AC algorithm comprises three phases. In the first phase, the algorithm searches the training data for any associations between the attributes' values and the class labels. The discovered associations are generated as Class Association Rules (CARs) in an "IF-THEN" format [32]-[33]. After generating the complete set of CARs, pruning and ranking procedures are used to prune weak rules according to some specific thresholds, such as *Support* and *Confidence* and rank the remaining strong rules according to their *Support*, *Confidence* and the number of the conditions in the antecedent of the rule or any other ranking criteria (Phase 2). The final output of the second phase is the classifier, which comprises a set of CARs. Lastly, the classifier is tested against a new and independent dataset to verify its effectiveness in predicting new unseen instances [34]. Figure 1 shows the main general phases for any AC algorithm and Table 1 shows some main concepts and definitions related to the AC.

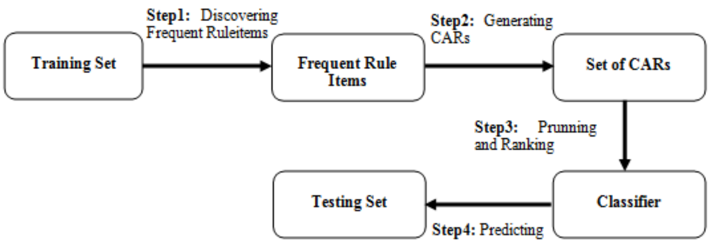



Figure 1. General steps for AC algorithms.

Table 1. Main definitions and concepts related to AC learning approach.

| Concept | Definition |
|---|---|
| Item | An association between an attribute in the dataset and its value ($A_i$,$a_i$) or a combination of several attributes' values ($A_1$,$a_1$), ($A_7$,$a_7$), ($A_9$,$a_9$). |
| Rule | An "IF-Then" rule that has a combination of items in the antecedent and one class label only in the consequent. |
| Actual Occurrence (AccOccur) | Number of cases in the training dataset that matches the antecedent of a rule. |
| Support Count (SuppCount) | Number of cases in the training dataset that match the antecedent of a rule and belong to a specific class label. |
| Minimum Support (MinSupp) | A user predefined threshold. A rule r passes the minsup threshold if SuppCount(r)/n >= MinSupp, where n: number of instances in the training set. |
| Minimum Confidence (MinConf) | A user predefined threshold. A rule r passes the MinConf threshold if SuppCount(r)/AccOccur(r) >= MinConf. |
| Frequent Item | An item in the training dataset that passes the MinSupp threshold. |

In general, AC-based algorithms start with discovering frequent items that comprise only a single value; i.e., items <A1,$a_1$>, <A2,$a_2$> and <A2,$x_1$>. Any item that passes the user predefined MinSupp threshold is said to be a frequent single item.

For example, in Table 2, if the MinSupp equals 0.4, then the SuppCount will be 4, because there are 10 instances (cases) in the dataset. Therefore, the following are the single frequent items: <A1,$a_1$>, <A1,$a_2$>, <A2,$x_1$> and <A2,$x_2$>. After that, based on the discovered single frequent items, a new pass over the dataset is carried out to discover frequent triples of items and so on. Thus, most AC algorithms perform several passes over the training set to generate the frequent items that satisfy the user predefined MinSupp threshold.

The next step is to generate the complete set of CARs that satisfy the MinConf threshold based on the discovered frequent items. For example, the following rule could be generated from Table 2, considering that MinConf=0.8: <A1,a1> and <A2,x1> ➔ C1.

Finally, after generating all CARs, a ranking and pruning step is applied on the discovered CARs to keep the most accurate CARs and remove the others.

Table 2. Transactional training dataset.

| Row ID | A1 | A2 | Class |
|---|---|---|---|
| 1 | a1 | x1 | C1 |
| 2 | a1 | x2 | C1 |
| 3 | a1 | x1 | C1 |
| 4 | a2 | x2 | C3 |
| 5 | a2 | x1 | C4 |
| 6 | a2 | x2 | C2 |
| 7 | a2 | x2 | C2 |
| 8 | a1 | x1 | C1 |
| 9 | a2 | x2 | C2 |
| 10 | a1 | x1 | C1 |

Several research studies have shown that AC has two distinguishable features over other methods and approaches of classification [14], [35]. The first one is its simplicity in representing the knowledge in the form of "IF-THEN" rules and its high interpretability. The second distinguishable feature is its great ability to find hidden and additional information, which leads to minimizing the error rate of the classifier and hence highly improving the classification step.

Classification Based on Associations (CBA), which is one of the first algorithms that combined the tasks of Association Rule Mining (ARM) and Classification, was proposed in [36]. Since then, many other algorithms have been proposed based on the concept of merging ARM with classification. CBA managed to utilize the *Apriori* algorithm [37] in a classification dataset through applying three main steps. In the first step, any continuous attribute (if any) in the dataset is discretized. The second step of CBA involves generating all CARs. CARs consider only those rules that have any combination of items in the left-hand side (antecedent) and only one of the classes in the right-hand side (consequent). CARs are chosen based on user-defined thresholds called *Support* and *Confidence*, in which the value of the *Support* threshold is usually very low and the value of the *Confidence* value is high. The third step aims to build a single-label classifier based on the previously discovered CARs [30].

CBA was improved later in [38] by eliminating two weaknesses related to the original CBA. The first weakness is using one value for the *minsup* threshold, which might cause imbalanced class distribution. This weakness has been tackled in the adapted version through using multiple *minsup* thresholds. The second weakness of the original CBA is the exponential growth of the number of rules generated by CBA. This weakness has been tackled by merging CBA with a decision tree as in C4.5, which has led to more accurate rules. The adapted version of CBA has been called CBA2 or msCBA, short for multiple support CBA.

Although CBA2 has shown excellent performance in single-label classification when compared with other algorithms that follow other learning strategies [30], unfortunately, CBA2 does not have the capability to handle multi- label datasets. CBA2 assumes that only one class label is associated with an input instance. Thus, it produces single-label rules with one class label as a consequence of the rule. Hence, to adapt the CBA2 algorithm to handle multi-label datasets, this assumption should be avoided. Also, the CBA2 algorithm captures the associations between features (attributes) and class labels globally, where local associations and dependencies are proven to have a better performance than global associations and dependencies [39].

Yin and Han (2003) [40] proposed an AC algorithm that has been called CPAR, short for Classification based on Predictive Association Rules. CPAR guarantees the generation of more rules, because the training set is allowed to be covered by several rules instead of one single rule, which leads to an improvement in the classification accuracy. CPAR managed to do that by enhancing the First-Order Inductive Learner (FOIL) and considering all the positive cases associated with the generated rule instead of discarding them as in other AC algorithms. Also, CPAR can generate simultaneous multiple rules at the same time by considering the value of all attributes with the largest FOIL-gain instead of considering only one attribute value as in FOIL.

The classification based on Multiple Association Rules (CMAR) algorithm is another AC algorithm that was proposed in [41]. CMAR was the first AC-based algorithm that utilized the FP-growth technique to capture the hidden associations among the features and the class labels. CMAR used a prefix tree data structure called C-tree to save the learned rules. An extensive evaluation based on 26 UCI datasets revealed that CMAR has a competitive performance compared with the CBA and C4.5 algorithms.

In [42], a new AC-based algorithm was presented. The algorithm was called FCRA, short for finding Fuzzy Classification Rules based on the *Apriori* algorithm. FCRA proposed a new data mining technique that captures fuzzy classification rules based on the *Apriori* algorithm. FCRA utilizes a genetic algorithm to automatically determine the threshold of the minimum fuzzy support. An evaluation of the FCRA algorithm on Iris dataset revealed its superior performance compared with other classification algorithms.

A fuzzy-based AC algorithm that enhances the understandability of the generated classifier by reducing the total number of the classification rules was presented in [43]. Classification with Fuzzy Association Rules (CFAR) utilizes the concept of fuzzy logic in solving the so-called "sharp boundary" problem in ARM techniques with quantitative attributes’ domains. CFAR has been compared against CBA and showed a better performance in terms of understandability represented by the total number of the generated classification rules.

## C. Utilizing AC in MLC

One of the most popular algorithms that utilizes AC to handle the problem of MLC is the Multiclass Multilabel Associative Classification (MMAC) algorithm [29]. MMAC comprises three steps. First, it transforms the multi-label dataset into a single-label dataset, using *copy* as a problem transformation method. Second, it trains a single-label associative classifier to predict a single-label using "IF-THEN" rules. Finally, it merges the predictions of rules that have the same antecedent to form a rule with more than one label in the consequence of the rule. It is worth mentioning that all the datasets that have been used to evaluate MMAC are single label datasets and MMAC has never been tested against multi-label datasets. Also, MMAC assumes class labels to be mutually exclusive and ignores any dependencies among labels, which makes it unsuitable for large datasets with high number of instances and labels.

In [44], a new multi-label algorithm based on AC was introduced and dubbed the Multi-label Classifier based on Associative Classification (MCAC). The algorithm uses a novel rule discovery method that generates and discovers multi-label rules from a single label dataset, without performing the learning step in the dataset. These multi-label rules represent vital information that is usually ignored by most previous AC algorithms. As in MMAC, this algorithm has been tested against single-label datasets and never considered the dependencies among labels.

In [45], a Correlatives Lazy Associative Classifier (CLAC) algorithm was introduced. CLAC is based on two approaches of classification: lazy learning that delays the reasoning process until a new test instance is given and associative classification that merges the association rule mining task with the classification task. In CLAC, the CARs do not have more than one label and consequently, CLAC assigns a value to each CAR based on its *Support* and *Confidence* and the associated class label. Then, CLAC adds the predicted class of the test instance to the instance as a new feature and uses the new test instance with the added feature (class label) to predict a new class label and so on until no further class label can be found. CLAC was evaluated against three textual datasets and achieved better performance compared with the BoosTexter algorithm [46].

In [47], another algorithm that followed the approach of AC was presented. The algorithm produced multi-label association rules by considering all the labels with a probability greater than or equal to (0.5). In fact, their algorithm is similar to MMAC in all of its steps with only one difference in the evaluation of the algorithm. MMAC has been evaluated using single-label datasets, while this algorithm has been evaluated using only one multi-label dataset (Scene). The authors concluded that using AC with MLC will lead to good performance, but generalizing this conclusion is difficult *via* an experiment on only one dataset with specific features and characteristics.

## 3. EMPIRICAL ANALYSIS

In this section, a comprehensive description of the research conducted is presented. At first, a description of the considered multi-label datasets and the settings of the six AC classifiers considered is introduced. Then, an evaluation of the results of the six AC classifiers is presented. Finally, a discussion regarding the evaluation results is provided.

The accuracy of the classification task in the domain of MLC is still low when compared to other types of classification, like the binary classification and the multi-class classification. Therefore, the main evaluation metric in this investigation study is the accuracy metric. Also, since all AC algorithms highly depend on the discretization technique being used, this paper studies and attempts to identify the most appropriate discretization technique which leads to the best accuracy results. Finally, the running time for the considered algorithms is considered to get the complete picture regarding the significance of utilizing the approach of AC in handling the problem of MLC.

### A. Settings and Datasets

Three multi-label datasets with different characteristics are considered in this paper. Each dataset has been transformed into a single-label dataset based on a novel transformation method called High Standard Deviation First (HSDF) [11], in which the label space of the multi-label dataset is extracted first. Then, for each label (item) in the extracted label space, the Predictive Apriori [48] algorithm is applied to capture all positive pairwise associations in the form (IF X=1, THEN Y=1). After that, the standard deviation of the accuracies of the captured positive associations is computed for each label. Finally, the labels are ordered based on the computed standard deviation in a descending order and the input dataset is transformed into a single-label dataset based on the order determined earlier.

Table 3 shows the main characteristics of the considered multi-label datasets in this paper. All datasets are available in Mulan, a multi-label dataset repository [49]. Datasets could be downloaded from http://mulan.sourceforge.net/datasets-mlc.html. Finally, it is worth mentioning that the training datasets and the testing datasets have been chosen according to the datasets author recommendation, where 2/3 of the dataset has been used as a training set and 1/3 of the dataset has been used as a testing set.

Table 3. Datasets characteristics.

| Dataset | Instances | Attributes | Labels | LCard | Domain |
|---|---|---|---|---|---|
| Scene | 2712 | 294 | 6 | 1.074 | Image |
| Emotions | 593 | 72 | 6 | 1.868 | Media |
| Flags | 194 | 19 | 7 | 3.392 | Image |

Six AC-based classifiers have been considered in this paper. These classifiers are: CBA, CBA2, CMAR, CPAR, FCRA and CFAR. These classifiers have been used with their default settings as they have been implemented in KEEL [50]. KEEL is an open source Java software that can be used in a wide range of data mining tasks. KEEL is short for Knowledge Extraction based on Evolutionary Learning.

Each classifier has been trained on each transformed version of the considered dataset five times and each time a different discretization technique is used. Five discretization techniques are considered in this paper: Chi2-D [51], Bayesian-D [52], Ameva-D [53], 1R-D [54] and E-Chi2-D [55].

### B. Evaluation of Several AC-based Classifiers and other Classifiers from Different Learning Approaches

Table 4 to Table 6 show the results of the evaluation of the six AC classifiers on the three considered multi-label datasets using accuracy metric. Accuracy measures the percentage of those labels that were correctly predicted, with respect to the total number of labels and averaged over all instances. Accuracy is computed using the following equation:

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN} \tag{1}$$

where TP=number of the true positive predictions, TN=number of the true negative predictions, FP=number of the false positive predictions and FN=number of the false negative predictions.

Table 4 shows the accuracy rates of the six different AC-based classifiers on the *Scene* dataset, with respect to 5 discretization techniques. The *Scene* dataset comprises 2712 instances and 294 attributes.

Table 4. Accuracy rates of the six AC algorithms on the *Scene* dataset.

| **Algorithm** | **Chi2-D** | **Bayesian-D** | **Ameva-D** | **1R-D** | **E-Chi2-D** | **Average** |
|---|---|---|---|---|---|---|
| **CBA2** | 0.840 | **0.991** | 0.801 | 0.810 | 0.643 | **0.817** |
| **CBA** | 0.774 | 0.817 | 0.830 | 0.753 | 0.633 | 0.761 |
| **CMAR** | 0.776 | 0.669 | 0.758 | 0.712 | 0.615 | 0.706 |
| **CPAR** | 0.730 | 0.753 | 0.742 | 0.669 | 0.607 | 0.700 |
| **FCRA** | 0.559 | 0.591 | 0.562 | 0.523 | 0.541 | 0.555 |
| **CFAR** | 0.273 | 0.136 | 0.270 | 0.145 | 0.027 | 0.170 |
| **Averag** | 0.659 | 0.660 | **0.661** | 0.602 | 0.511 | |

According to Table 4, CBA2 has the best accuracy average on the Scene dataset. Considering the discretization techniques, it can be clearly noted from the table that Ameva-D is the best discretization technique among the 5 considered techniques. Nevertheless, Chi2-D and Bayesian-D show nearly equivalent results to Ameva-D. Finally, the highest accuracy was observed with the CBA2 algorithm when using Bayesian-D as a discretization technique.

Table 5 shows the accuracy rates for the considered AC classifiers on the *Emotions* dataset, with respect to 5 discretization techniques. The *Emotions* dataset comprises 593 instances and 72 attributes.

Table 5. Accuracy rates of the six AC algorithms on the *Emotions* dataset.

| **Algorithm** | **Chi2-D** | **Bayesian-D** | **Ameva-D** | **1R-D** | **E-Chi2-D** | **Average** |
|---|---|---|---|---|---|---|
| **CBA2** | **0.966** | 0.877 | 0.815 | 0.455 | 0.953 | **0.813** |
| **CBA** | 0.598 | 0.526 | 0.613 | 0.447 | 0.624 | 0.562 |
| **CMAR** | 0.529 | 0.396 | 0.529 | 0.258 | 0.526 | 0.448 |
| **CPAR** | 0.603 | 0.560 | 0.562 | 0.429 | 0.598 | 0.550 |
| **FCRA** | 0.294 | 0.388 | 0.452 | 0.416 | 0.342 | 0.378 |
| **CFAR** | 0.209 | 0.209 | 0.209 | 0.209 | 0.209 | 0.209 |
| **Average** | 0.533 | 0.493 | 0.530 | 0.369 | **0.542** | |

Table 5 clearly shows that CBA2 has the highest accuracy among the six AC classifiers on the *Emotions* dataset, especially when using Chi2-D discretization technique.

For the discretization techniques, E-Chi2-D shows the best results, with a competitive performance from the Chi2-D and Ameva-D techniques. The best accuracy has been observed with the CBA2 algorithm when using Chi2-D as a discretization technique.

Table 6 shows the accuracy rates of the six different AC-based classifiers on the *Flags* dataset, with respect to 5 discretization techniques. The *Flags* dataset comprises 194 instances and 19 attributes.

Table 6. Accuracy rates of the six AC algorithms on the *Flags* dataset.

| **Algorithm** | **Chi2-D** | **Bayesian-D** | **Ameva-D** | **1R-D** | **E-Chi2-D** | **Average** |
|---|---|---|---|---|---|---|
| **CBA2** | **0.912** | 0.855 | 0.865 | 0.835 | 0.855 | **0.864** |
| **CBA** | 0.798 | 0.752 | 0.768 | 0.737 | 0.752 | 0.761 |
| **CMAR** | 0.768 | 0.721 | 0.747 | 0.680 | 0.721 | 0.727 |
| **CPAR** | 0.608 | 0.572 | 0.592 | 0.603 | 0.572 | 0.589 |
| **FCRA** | 0.510 | 0.510 | 0.510 | 0.510 | 0.510 | 0.510 |
| **CFAR** | 0.185 | 0.185 | 0.185 | 0.185 | 0.185 | 0.185 |
| **Average** | **0.630** | 0.599 | 0.611 | 0.592 | 0.599 | |

Table 6 clearly shows that CBA2 has the best accuracy on the *Flags* dataset, especially when using Chi2-D as a discretization technique. Considering the discretization techniques, Chi2-D has the best results, with a competitive performance from the Ameva-D technique. The best accuracy has been observed with CBA2 algorithm when using Chi2-D as a discretization technique.

Based on the accuracy results for the six AC-based classifiers on the three multi-label datasets, the conclusion can be made that CBA2 algorithm is the best AC algorithm in handling multi-label datasets. Table 7 shows the running time in seconds (time needed to build the classifier) for the six AC classifiers on the three datasets, with respect to the discretization technique being used.

Table 7. Running time for several AC classifiers on the considered datasets.

| **Dataset** | **Algorithm** | **Chi2-D** | **Bayesian-D** | **Ameva-D** | **1R-D** | **E-Chi2-D** | **Average** |
|---|---|---|---|---|---|---|---|
| **Scene** | **CBA2** | 5 | 6 | 3 | 8 | 4 | 5.2 |
| | **CBA** | 7 | 105 | 21 | 15 | 13 | 32.2 |
| | **CMAR** | 773 | 2488 | 1895 | 3722 | 691 | 1913.8 |
| | **CPAR** | 1 | 1 | 1 | 1 | 0 | 0.8 |
| | **FCRA** | 204 | 106 | 212 | 7263 | 5620 | 2681 |
| | **CFAR** | 11 | 1 | 36 | 100 | 1508 | 331.2 |
| | **Average** | **166.8** | 451.2 | 361.3 | 1851.5 | 1306.0 | |
| **Emotions** | **CBA2** | 0 | 0 | 0 | 0 | 1 | 0.2 |
| | **CBA** | 0 | 0 | 0 | 0 | 0 | 0 |
| | **CMAR** | 0 | 0 | 0 | 0 | 0 | 0 |
| | **CPAR** | 0 | 0 | 0 | 0 | 1 | 0.2 |
| | **FCRA** | 12 | 54 | 13 | 81 | 12 | 34.4 |
| | **CFAR** | 0 | 0 | 0 | 1 | 0 | 0.2 |
| | **Average** | **2.0** | 9.0 | 2.2 | 13.7 | 2.3 | |
| **Flags** | **CBA2** | 47 | 6 | 42 | 10 | 13 | 23.6 |
| | **CBA** | 3 | 1 | 3 | 2 | 2 | 2.2 |
| | **CMAR** | 1 | 1 | 2 | 2 | 1 | 1.4 |
| | **CPAR** | 0 | 0 | 0 | 0 | 0 | 0 |
| | **FCRA** | 42 | 62 | 32 | 99 | 71 | 61.2 |
| | **CFAR** | 2 | 2 | 2 | 5 | 5 | 3.2 |
| | **Average** | 15.8 | **12.0** | 13.5 | 19.7 | 15.3 | |

In general, CPAR shows the best running times on the three datasets considering the five discretization techniques. Among the discretization techniques, Chi2-D has the best running time on the *Scene* and *Emotions* datasets, while it has an acceptable running time on the *Flags* datasets. The CBA2 algorithm has an acceptable running time among the six considered AC classifiers.

Considering the accuracy and the running-time criteria in the era of distributed computing and high-speed processors, the conclusion can be drawn that CBA2 is the best AC classifier to be adapted to handle the problem of MLC. To make this conclusion reasonable, Figure 2 presents a comparison between the best accuracy results maintained for CBA2 algorithm and several other algorithms that belong to different learning approaches on the three datasets considered in this paper. CBA2 was compared to 12 algorithms that belong to six different learning strategies, where the decision tree learning approach was represented by the C4.5 [56] and DT-GA [57] algorithms. From the neural network learning approach, two algorithms have been considered: GANN [58] and NNEP [59] and fuzzy learning approach was represented by FURIA [60] and WF [61] algorithms. Also, two algorithms that belong to lazy learning have been considered: KNN [62] and KNN-Adaptive [63]. The statistical learning approach was represented by Logistic [64] and LDA [65] algorithms. Finally, the Support Vector Machine (SVM) learning approach was represented by the C-SVM [66] and SMO [67] algorithms.

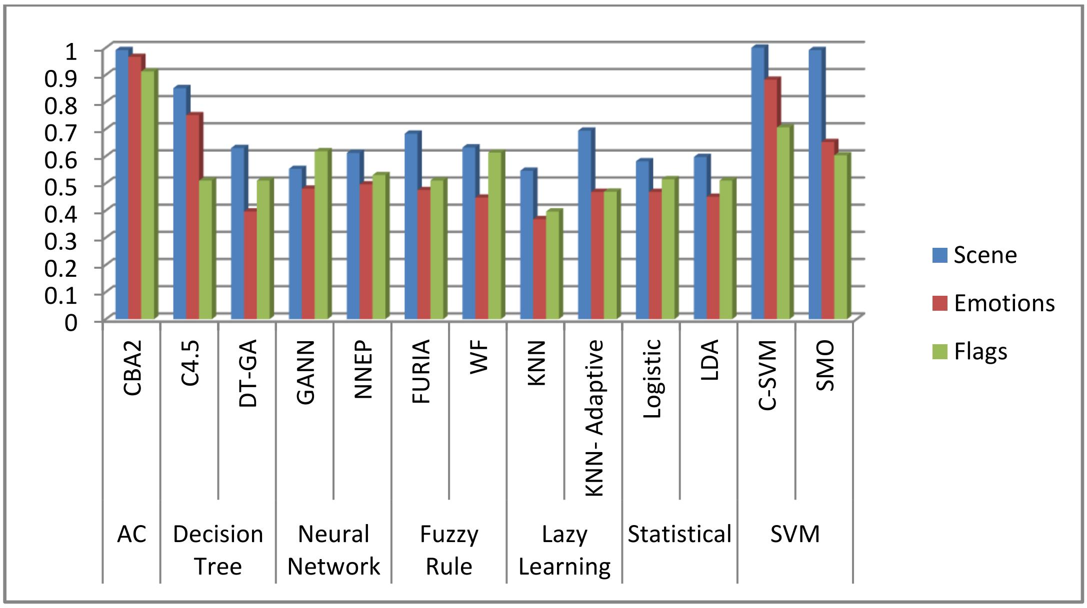



Figure 2. Accuracy rates of CBA2 and several other algorithms on the three datasets.

From Figure 2, clearly the CBA2 algorithm has a superior performance on the *Emotions* and *Flags* datasets. Also, the performance of the CBA algorithm is excellent on the *Scene* dataset, where it has the second-best accuracy after the C-SVM algorithm.

What is distinguishable about CBA2 is that it maintains the same level of performance regardless the characteristics of the datasets, which makes it an excellent choice to handle different multi-label datasets with different characteristics.

To summarize, CBA2 is better than the 12 other algorithms that belong to 6 learning approaches based on the accuracy metric. This fact reveals the significance of adapting the CBA2 algorithm to handle the problem of MLC.

## C. Results' Discussion

In General, CBA2 shows a superior performance among the considered AC classifiers on the three datasets based on the accuracy metric. The accuracy of CBA2 has been greatly affected by the discretization technique being used. The results showed that Chi2-D is the most appropriate discretization technique to be used with CBA2 to handle multi-label datasets.

With respect to running time, CPAR has the best running time among the six AC classifiers. Other AC classifiers such as CBA, CBA2 and CMAR, have acceptable running times. Nevertheless, the accuracy of the classification task is more significant than the complexity of the multi-label classifier and its running time, especially in an era of distributed computing and high-efficiency processors. Therefore, CBA2 will be the most promising AC classifier to be adapted to handle the problem of MLC.

Also, a significant issue in determining the applicability of AC algorithm in handling the problem of MLC is the total number of generated rules [14], [30].

Table 8 shows the total number of rules generated by the best four AC classifiers on the three considered datasets with respect to the best three discretization techniques.

Table 8 shows that the total number of rules varies across the four algorithms as well as across the three discretization techniques. The CBA algorithm has the lower values on the three datasets, which makes it an appropriate choice to handle the problem of MLC with respect to the size of the generated classifier. Nevertheless, the accuracy of CBA is less than the accuracy of CBA2. Hence, a trade-off must made between the accuracy results and the size of the classifier results. However, if powerful pruning techniques are utilized, CBA2 will be the best choice to handle MLC problems. Therefore, future work should investigate the most appropriate pruning techniques to be used with AC classifiers to handle MLC datasets that usually suffer from high-dimensionality problems [3], [68].

Table 8. Total number of rules generated by different AC classifiers.

| **Dataset** | **Algorithm** | **Chi2-D** | **Bayesian-D** | **Ameva-D** | **Average** |
|---|---|---|---|---|---|
| **Scene** | **CBA2** | 235 | 330 | 219 | 261 |
| | **CBA** | 196 | **155** | 201 | **184** |
| | **CMAR** | 1554 | 1276 | 1598 | 1476 |
| | **CPAR** | 698 | 1127 | 841 | 889 |
| | **Average** | **671** | 722 | 715 | |
| **Emotions** | **CBA2** | 248 | 204 | 171 | 208 |
| | **CBA** | 78 | **41** | 71 | **63** |
| | **CMAR** | 195 | 141 | 259 | 198 |
| | **CPAR** | 755 | 673 | 487 | 638 |
| | **Average** | 319 | 265 | **247** | |
| **Flags** | **CBA2** | 104 | 91 | 90 | 95 |
| | **CBA** | 73 | 64 | **60** | **66** |
| | **CMAR** | 481 | 421 | 443 | 448 |
| | **CPAR** | 247 | 213 | 210 | 223 |
| | **Average** | 226 | **197** | 201 | |

Finally, based on the accuracy of the several AC classifiers on the three datasets, with respect to the accuracy of other algorithms from different learning approaches and strategies, the assumption can be made that AC approach could be more appropriate to be used in the domain of MLC than other learning approaches, especially in a form of ensemble classifiers with ensemble discretization techniques.

## 4. CONCLUSION AND FUTURE WORK

In this paper, an investigation regarding the applicability of AC in solving the problem of MLC has been presented. Six different AC-based classifiers have been evaluated on three multi-label datasets, with respect to five well-known discretization techniques.

Based on the evaluation results, it can be concluded that AC learning approach achieved a superior performance with respect to the accuracy metric compared with the six learning approaches which have been considered in this paper, which indicates that adapting AC to handle the problem of MLC is a promising research work.

Among the considered AC classifiers, CBA2 has shown the best accuracy on the three considered datasets. Also, determining the discretization technique that is optimal to handle multi-label datasets is a crucial decision, where Chi2-D has shown to have an excellent performance when compared with other discretization techniques.

Future work could be done in several areas. First, the CBA2 algorithm could be adapted to handle multi-label datasets. Second, other promising future work is to propose a new MLC algorithm based on an ensemble of several AC classifiers. Third, an investigation regarding the best pruning technique to be used with multi-label datasets that suffer from high number of attributes will be good for future investigation.

## REFERENCES

[1] L. Al Qadi, H. El Rifai, S. Obaid and A. Elnagar, "A Scalable Shallow Learning Approach for Tagging Arabic News," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 06, no. 03, pp. 263-280, September 2020.

[2] L. T. Nguyen, B. Vo, T. Mai and T. L. Nguyen, "A Weighted Approach for Class Association Rules," Proc. of Modern Approaches for Intelligent Information and Database Systems, pp. 213-222, Part of the Studies in Computational Intelligence Book Series, vol. 769, Springer, Cham, 2018.

[3] E. Gibaja and S. Ventura, "A Tutorial on Multilabel Learning," ACM Computing Surveys, vol. 47, no. 3, pp. 1-39, 2015.

[4] R. Sousa and J. Gama, "Multi-label Classification from High-speed Data Streams with Adaptive Model Rules and Random Rules," Progress in Artificial Intelligence, vol. 7, pp. 177-187, 2018.

[5] R. Alazaidah, F. K. Ahmad and M. F. M. Mohsen, "A Comparative Analysis between the Three Main Approaches that are Being Used to Solve the Problem of Multi Label Classification," International Journal of Soft Computing, vol. 12, no. 4, pp. 218-223, 2017.

[6] J. Huang, G. Li, S. Wang, Z. Xue and Q. Huang, "Multi-label Classification by Exploiting Local Positive and Negative Pairwise Label Correlation," Neurocomputing, vol. 257, pp. 164-174, 2017.

[7] R. Alazaidah, F. K. Ahmad, M. F. M. Mohsen and A. K. Junoh, "Evaluating Conditional and Unconditional Correlations Capturing Strategies in Multi Label Classification," Journal of Telecommunication, Electronic and Computer Engineering (JTEC), vol. 10, no. 2-4, pp. 47-51, 2018.

[8] Y. Qu, G. Yue, C. Shang, L.Yang, R. Zwiggelaar and Q. Shen, "Multi-criterion Mammographic Risk Analysis Supported with Multi-label Fuzzy-rough Feature Selection," Artificial Intelligence in Medicine, vol. 100, Article ID: 101722, DOI: 10.1016/j.artmed.2019.101722, 2019.

[9] R. Alazaidah, F. K. Ahmad, M. F. M. Mohsin and W. A. AlZoubi, "Multi-label Ranking Method Based on Positive Class Correlations," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 6, no. 4, pp. 377-391, Dec. 2020.

[10] R. Alazaidah, F. Thabtah and Q. Al-Radaideh, "A Multi-label Classification Approach Based on Correlations among Labels," International Journal of Advanced Computer Science and Applications, vol. 6, no. 2, pp. 52-59, 2015.

[11] R. Alazaidah, F. Ahmad and M. Mohsin, "Multi Label Ranking Based on Positive Pairwise Correlations among Labels," The International Arab Journal of Information Technology, vol. 17, no. 4, DOI: 10.34028/iajit/17/4/2, 2019.

[12] M. Atzmueller, N. Hayat, M. Trojahn and D. Kroll, "Explicative Human Activity Recognition Using Adaptive Association Rule-based Classification," Proc. of the IEEE International Conference on Future IoT Technologies (Future IoT), pp. 1-6, DOI: 10.1109/FIOT.2018.8325603, Eger, Hungary, 2018.

[13] Y. Chengxin, G. Yan, Y. Jianguo and R. Xiaoting, "A New Recommendation System on the Basis of Consumer Initiative Decision Based on an Associative Classification Approach," Industrial Management & Data Systems, vol. 118, no. 1, pp.188-203, 2018.

[14] N. Abdelhamid, A. A. Jabbar and F. Thabtah, "Associative Classification Common Research Challenges," Proceedings of the 45th IEEE International Conference on Parallel Processing Workshops, pp.432-437, Philadelphia, USA, 2016.

[15] G. Corani and M. Scanagatta, "Air Pollution Prediction *via* Multi-label Classification," Environmental Modelling & Software, vol. 80, pp. 259-264, 2016.

[16] E. C. Gonçalves, A. Plastino and A. A. Freitas, "A Genetic Algorithm for Optimizing the Label Ordering in Multi-label Classifier Chains," Proceedings of the 25th IEEE International Conference on Tools with Artificial Intelligence, pp. 469-476, Herndon, USA, 2013.

[17] X. Li, J. Ouyang and X. Zhou, "Centroid Prior Topic Model for Multi-label Classification," Pattern Recognition Letters, vol. 62, pp. 8-13, 2015.

[18] S. Ali and A. Majid, "Can–Evo–Ens: Classifier Stacking Based Evolutionary Ensemble System for

Prediction of Human Breast Cancer Using Amino Acid Sequences," Journal of Biomedical Informatics, vol. 54, pp. 256-269, 2015.

[19] E. A. Tanaka, S. R. Nozawa, A. A. Macedo and J. A. Baranauskas, "A Multi-label Approach Using Binary Relevance and Decision Trees Applied to Functional Genomics," Journal of Biomedical Informatics, vol. 54, pp. 85-95, 2015.

[20] K. Trohidis, G. Tsoumakas, G. Kalliris and I. P. Vlahavas, "Multi-label Classification of Music into Emotions," Proceedings of the 9th International Conference on Music Informative Retrieval, pp. 325-330, Philadelphia, PN: Drexel University, 2008.

[21] A. Dimou, G. Tsoumakas, V. Mezaris, I. Kompatsiaris and I. Vlahavas, "An Empirical Study of Multi-label Learning Methods for Video Annotation," Proceedings of the 7th IEEE International Workshop on Content-based Multimedia Indexing, pp. 19-24, Chania, Greece, 2009.

[22] T. Li, C. Zhang and S. Zhu, "Empirical Studies on Multi-label Classification," Proc. of the 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06), DOI: 10.1109/ICTAI.2006.55, Arlington, USA, 2006.

[23] Z. Barutcuoglu, R. E. Schapire and O. G. Troyanskaya, "Hierarchical Multi-label Prediction of Gene Function," Bioinformatics, vol. 22, no. 7, pp. 830-836, 2006.

[24] J. Lee, H. Kim, N. R. Kim and J. H. Lee, "An Approach for Multi-label Classification by Directed Acyclic Graph with Label Correlation Maximization," Information Sciences, vol. 351, pp. 101-114, 2016.

[25] M. Alluwaici, A. K. Junoh, F. K. Ahmad, M. F. M. Mohsen and R. Alazaidah, "Open Research Directions for Multi Label Learning," Proc. of the IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), pp. 125-128, DOI: 10.1109/ISCAIE.2018.8405456, Penang, 2018.

[26] S. Xu, X. Yang, H. Yu, D. J. Yu, J. Yang and E. C. Tsang, "Multi-label Learning with Label-specific Feature Reduction," Knowledge-based Systems, vol. 104, pp. 52-61, 2016.

[27] J. Read, B. Pfahringer, G. Holmes and E. Frank, "Classifier Chains for Multi-label Classification," Machine Learning, vol. 85, no. 3, Article no. 333, 2011.

[28] F. Thabtah, "Challenges and Interesting Research Directions in Associative Classification," Proceedings of the 6th IEEE Int. Conference on Data Mining Workshops, pp. 785-792, Hong Kong, China, 2006.

[29] F. A. Thabtah, P. Cowling and Y. Peng, "MMAC: A New Multi-class, Multi-label Associative Classification Approach," Proceedings of the 4th IEEE International Conference on Data Mining, pp. 217-224, Brighton, UK, 2004.

[30] N. Abdelhamid and F. Thabtah, "Associative Classification Approaches: Review and Comparison," Journal of Information & Knowledge Management, vol. 13, no. 3, pp. 1-30, 2014.

[31] Y. Yang, A. Stein, V. A. Tolpekin and Y. Zhang, "High-resolution Remote Sensing Image Classification Using Associative Hierarchical CRF Considering Segmentation Quality," IEEE Geoscience and Remote Sensing Letters, vol. 15, no. 5, pp. 754-758, 2018.

[32] P. B. Shekhawat and S. S. Dhande, "A Classification Technique Using Associative Classification," International Journal of Computer Applications, vol. 20, no. 5, pp. 20-28, 2011.

[33] Y. W. C. Chien and Y. L. Chen, "Mining Associative Classification Rules with Stock Trading Data: A GA-based Method," Knowledge-based Systems, vol. 23, no. 6, pp. 605-614, 2010.

[34] F. Thabtah, Q. Mahmood, L. McCluskey and H. Abdel-Jaber, "A New Classification Based on Association Algorithm," J. of Information & Knowledge Management, vol. 9, no. 1, pp. 55-64, 2010.

[35] R. AlShboul, F. Thabtah, N. Abdelhamid and M. Al-Diabat, "A Visualization Cybersecurity Method Based on Features' Dissimilarity," Computers & Security, vol. 77, pp. 289-303, 2018.

[36] B. Liu, W. Hsu and Y. Ma, "Integrating Classification and Association Rule Mining," Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining, pp.80-86, American Association for Artificial Intelligence, 1998.

[37] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," Proceedings of the 20th International Conference on Very Large Databases, pp. 487-499, Santiago, Chile, 1994.

[38] B. Liu, Y. Ma and C. K. Wong, "Improving an Association Rule Based Classifier," Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery, pp. 504-509, Berlin, Heidelberg: Springer-Verlag, 2009.

[39] S. J. Huang, Z. H, Zhou and Z. Zhou, "Multi-label Learning by Exploiting Label Correlations Locally," Proceedings of the 26th AAAI Conf. on Artificial Intelligence, pp. 949-955, Palo Alto, CA: AAAI, 2012.

[40] X. Yin and J. Han, "CPAR: Classification Based on Predictive Association Rules," Proceedings of the 2003 SIAM International Conference on Data Mining, pp. 331-335, Society for Industrial and Applied Mathematics, Philadelphia, PN, USA, 2003.

[41] W. Li, J. Han and J. Pei, "CMAR: Accurate and Efficient Classification Based on Multiple Class-association Rules," Proc. of the IEEE Int. Conf. on Data Mining, pp. 369-376, San Jose, USA, 2001.

[42] Y. C. Hu, R. S. Chen and G. H. Tzeng, "Finding Fuzzy Classification Rules Using Data Mining Techniques," Pattern Recognition Letters, vol. 24, no. 1-3, pp. 509-519, 2003.

[43] Z. Chen and G. Chen, "Building an Associative Classifier Based on Fuzzy Association Rules," International Journal of Computational Intelligence Systems, vol. 1, no. 3, pp. 262-273, 2008.

[44] N. Abdelhamid, A. Ayesh and W. Hadi, "Multi-label Rules Algorithm Based Associative Classification," Parallel Processing Letters, vol. 24, no. 1, pp. 1-21, 2014.

[45] A. Veloso, W. Meira, M. Gonçalves and M. Zaki, "Multi-label Lazy Associative Classification," Proc. of the European Conference on Principles of Data Mining and Knowledge Discovery, pp. 605-612, Berlin, Heidelberg:Springer-Verlag, 2007.

[46] R. E. Schapire and Y. Singer, "BoosTexter: A Boosting-based System for Text Categorization," Machine Learning, vol. 39, no. 2-3, pp. 135-168, 2000.

[47] X. Li, D. Qin and C. Yu, "ACCF: Associative Classification Based on Closed Frequent Itemsets," Proceeding of the 5th IEEE International Conference on Fuzzy Systems and Knowledge Discovery, pp. 380-384, Jinan, China, 2008.

[48] J. Han, J. Pei, Y. Yin and R. Mao, "Mining Frequent Patterns without Candidate Generation: A Frequent-pattern Tree Approach," Data Mining and Knowledge Discovery, vol. 8, no. 1, pp. 53-87, 2004.

[49] G. Tsoumakas, E. Spyromitros-Xioufis, J. Vilcek and I. Vlahavas, "Mulan: A Java Library for Multi-label Learning," Journal of Machine Learning Research, vol. 12, no. July, pp. 2411-2414, 2011.

[50] I. Triguero, S. González, J. M. Moyano, S. García, J. Alcalá-Fdez, J. Luengo et al., "KEEL 3.0: An Open Source Software for Multi-stage Analysis in Data Mining," International Journal of Computational Intelligence Systems, vol. 10, no. 1, pp. 1238-1249, 2017.

[51] H. Liu and R. Setiono, "Feature Selection *via* Discretization," IEEE Transactions on Knowledge and Data Engineering, vol. 9, no. 4, pp. 642-645, 1997.

[52] X. Wu, "A Bayesian Discretizer for Real-valued Attributes," The Computer Journal, vol. 39, no. 8, pp. 688-691, 1996.

[53] L. Gonzalez-Abril, F. J. Cuberos, F. Velasco and J. A. Ortega, "Ameva: An Autonomous Discretization Algorithm," Expert Systems with Applications, vol. 36, no. 3, pp. 5327-5332, 2009.

[54] R. C. Holte, "Very Simple Classification Rules Perform Well on Most Commonly Used Datasets," Machine Learning, vol. 11, no. 1, pp. 63-90, 1993.

[55] C. T. Su and J. H. Hsu, "An Extended Chi2 Algorithm for Discretization of Real Value Attributes," IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 3, pp. 437-441, 2005.

[56] J. R. Quinlan, "Combining Instance-based and Model-based Learning," Proceedings of the 10th International Conference on Machine Learning, pp. 236-243, Burlington, MA: Morgan Kauffman, 1993.

[57] D. R. Carvalho and A. A. Freitas, "A Hybrid Decision Tree/Genetic Algorithm Method for Data Mining," Information Sciences, vol. 163, no. 1-3, pp. 13-35, 2004.

[58] G. F. Miller, P. M. Todd and S. U. Hegde, "Designing Neural Networks Using Genetic Algorithms," Proceedings of the 3rd International Conference on Genetic Algorithms, pp. 379-384, Hillsdale, IL: Morgan Kaufmann, 1989.

[59] F. J. Martínez-Estudillo, C. Hervás-Martínez, P. A. Gutiérrez and A. C. Martínez-Estudillo, "Evolutionary Product-unit Neural Network Classifiers," Neurocomputing, vol. 72, no. 1-3, pp. 548-561, 2008.

[60] J. Hühn and E. Hüllermeier, "FURIA: An Algorithm for Unordered Fuzzy Rule Induction," Data Mining and Knowledge Discovery, vol. 19, no. 3, pp. 293-319, 2009.

[61] T. Nakashima, G. Schaefer, Y. Yokota and H. Ishibuchi, "A Weighted Fuzzy Classifier and Its Applica-

tion to Image Processing Tasks," Fuzzy Sets and Systems, vol. 158, no. 3, pp. 284-294, 2007.

[62] T. Cover and P. Hart, "Nearest Neighbor Pattern Classification," IEEE Transactions on Information Theory, vol. 13, no. 1, pp. 21-27, 1967.

[63] J. Wang, P. Neskovic and L. N. Cooper, "Improving Nearest Neighbor Rule with a Simple Adaptive Distance Measure," Pattern Recognition Letters, vol. 28, no. 2, pp. 207-213, 2007.

[64] S. Le Cessie and J. C. Van Houwelingen, "Ridge Estimators in Logistic Regression," Applied Statistics, vol. 41, no. 1, pp. 191-201, 1992.

[65] G. J. McLachlan, Discriminant Analysis and Statistical Pattern Recognition, vol. 544. John Wiley & Sons, ISBN: 978-0-471-69115-0, 2004.

[66] C. Cortes and V. Vapnik, "Support-vector Networks," Machine Learning, vol. 20, no. 3, pp. 273-297, 1995.

[67] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya and K. R. K. Murthy, "Improvements to Platt's SMO Algorithm for SVM Classifier Design," Neural Computation, vol. 13, no. 3, pp. 637-649, 2001.

[68] R. Alazaidah and F. K. Ahmad, "Trending Challenges in Multi Label Classification," International Journal of Advanced Computer Science and Applications, vol. 7, no. 10, pp. 127-131, 2016.

**ملخص البحث:**

يُعّد التّصنيف متعدّد العناوين (MLC) مجالاً مثيراً ومُهماً جذب اهتمام الكثير من الباحثين في العقدين الأخيرين. فقد تمّ تكييف عدّة خوارزميّات في تصنيف مفردة العنوان تنتمي الى استراتيجيّات تعلّم مختلفة واستخدامها للتّعامل مع مشكلة التّصنيف متعدّد العناوين. ومن المفاجيء أنّه لم يتمّ استخدام أي خوارزمية تصنيف تشاركي للتّعامل مع مشكلة التّصنيف مُتعدّد العناوين؛ علماً بأنّ خوارزميّات التّصنيف التّشاركي أظهرت أداءً توقعياًّ عالياً مقارنة بغيرها من استراتيجيّات التعلّم في مجال التّصنيف مُفرد العنوان. هذه الورقة عبارة عن استقصاءٍ عميق لاستخدام التّصنيف التّشاركي في مجال التّصنيف متعدّد العناوين. فقد جرى تقييم عدّة خوارزميات تصنيف تشاركي على ثلاث مجموعات بيانات متعددة العناوين باستخدام خمس تقنيات تجريد، وبينت النتائج أن استخدام خوارزميّات التّصنيف التّشاركي في مجال التّصنيف متعدّد العناوين هو مجال واعد إذا قورن باستخدام خوارزميّات أخرى من استراتيجيّات تعلّم مختلفة.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 07, No. 02, June 2021.

# AUTOMATIC GENERATION OF UML DIAGRAMS FROM SCENARIO-BASED USER REQUIREMENTS

Abdelkareem M. Alashqar

*(Received: 18-Mar.-2021, Revised: 9-May-2021 and 19-May-2021, Accepted: 22-May-2021)*


## *ABSTRACT*

*Effective software modeling tools are necessary for successful achievement of software engineering activities, especially when working in the analysis and design phase. Automating these tools facilitates work, makes it more productive and reduces cost and time of development. This paper aims at the development and validation of a method and a software tool for automatic generation of UML diagrams when following the approach of object-oriented development. These diagrams are generated from scenario-based requirements in order to facilitate the modeling process. So, a template of scenario-based requirements and its components are identified and constructed. Then a method including an algorithm is designed and implemented based on natural language processing (NLP) to generate UML diagrams automatically from the scenario-based requirements. The diagrams include sequence and class diagrams. The ability, performance and benefits of the proposed method and the software tool are reported by experimental results.*



## *KEYWORDS*

*Software engineering, Object-oriented, UML, Use cases, Scenarios, Natural language processing.*


## 1. INTRODUCTION

Currently, object-oriented (OO) approaches are popular in developing software systems. The Unified Modeling Language (UML) is a powerful notation that is utilized in the analysis and design of OO software development process. Traditionally, software developers perform a lot of work in understanding the documented software requirements in order to build the required UML diagrams before drawing them using Computer Aided Software Engineering (CASE) tools. However, additional work is also needed by software developers for specific understanding and gaining familiarity and experience of these CASE tools. Because software requirements are typically written in natural languages such as English, Natural Language Processing (NLP) tools can be utilized for automatic generation of software models, such as UML diagrams.

The authors in [1] argue that a complete automation of constructing UML diagrams using NLP appears to be impossible and hence more advanced automation is needed to be achieved. Software requirements in the OO development process are widely elicited and analyzed using UML use cases 0. Software functional requirements are described in terms of use cases and actors [3] and written in a restrictive way using system scenarios. A scenario represents a particular path in one of the use cases' set that constitutes the main use case diagram of the software being developed. Scenarios have a powerful method in sharing the needs of stakeholders and in describing the story of functional behavior [2]. UML sequence diagram is normally linked to the realization of the use case of the system being developed. It shows the objects involved in the functionality of the scenario and their interactions organized in time sequence in addition to the messages exchanged between these objects [4].

One way for depicting the class diagram is to capture objects involved in the sequence diagram and make aggregations' relationships based on the interactions among objects. Sequence diagram always helps in depicting the interaction (dynamic) view, while class diagram always helps in depicting the structural (static) view of the software. However, there is a lack of work in automatic generation of dynamic views of the software being developed, because most of work goes to automatically building a static view of the software such as its structure. Sequence diagram helps software developers verify whether the identified classes are sufficient for achieving user requirements written in scenario format or not [5].

This paper introduces a method that includes a designed algorithm based on NLP for analyzing and

A. M. Alashqar is with Faculty of Information Technology, The Islamic University of Gaza, Palestine. Email: aashgar@yahoo.com

extracting information from scenario-based user requirements written in English text. The method will be developed as a software tool for automatic generation of UML (AGUML) diagrams, especially sequence diagrams and class diagrams.

The rest of this paper is organized as follows: Section 2 presents related work. Section 3 introduces the research methodology where the algorithm of analyzing scenario-based requirements is designed. The main structure of our software tool for automatic generation of UML diagrams (AGUML) is described in Section 4. Section 5 presents experiments performed by AGUML, in addition to the analysis of results. Section 6 concludes the paper and provides a future work outlook.

## 2. RELATED WORK

In this section, we provide a literature review for some published research works in the field of generating UML diagrams automatically and how NLP can help in achieving this automation. Specifically, we focus on the research works that tackle the problem of generating sequence and class diagrams from written user requirements.

The UML is a powerful notation that is used particularly in OO analysis and design activities when developing software systems. However, it is considered a semi-formal language that needs semantics when constructing and verifying its different types of diagrams [6].

The notion of using NLP in OO modeling has been initially proposed by [7][8][9], where the authors mentioned that the object and its properties can be recognized by searching nouns, while the object's operations and the interactions among objects can be recognized by searching verbs in written software requirements.

There have been many works that adopt these founded concepts in building CASE tools for automatic generation of UML diagrams from informal natural language. Most of these tools focus on generating class diagrams, such as REBUILDER UML [10], CM-Builder [11], LIDA [12], MOVA [13], UMLG [14], RACE [15], DC-Builder [16] and RAPID [17]. The authors in [18] developed a tool named GOOAL based on a semi-formal language (4WL) for generating class models automatically from semi-structured natural language. Their tool also can construct initial and general sequence diagrams.

For other types of UML diagrams, the authors in [19] developed a plugin for automatic transformation of user stories into UML use case diagrams depending on the basis of NLP techniques. Whereas, the authors in [20] proposed a semi-automated approach for generating use case diagrams from user requirements that are mainly written in Arabic.

Recently, the authors in [21] developed an algorithm for reading and processing user stories stored in a text file, then generating an Extensible Markup Language (XMI) file for each individual user story and finally transforming the XMI file into use case diagram. The authors in [22] proposed a semi-automated approach for generating sequence diagrams from user requirements written in Arabic. The generated sequence diagrams are expressed using XMI format in order to be drawn using specialized software drawing tools.

In this paper, a method that includes an algorithm and a software tool for generating UML sequence and class diagrams is proposed. The differences between our work and the previously stated related published works is that our proposed method utilizes the software requirements that are written as user scenarios. User scenarios are considered a powerful technique for writing detailed software descriptions when adopting use case approach in the process of requirements' elicitation and analysis when developing software. To our knowledge, no such work for automatic generation of sequence diagrams from user scenarios is achieved. Moreover, our work introduces a way for constructing class diagrams from the generated sequence diagrams by understanding the relationships and interactions among the participating objects. Moreover, our developed tool has a complete capability for generating UML disarms in order to be used in user documentation.

## 3. RESEARCH METHODOLOGY

The scenario-based requirements template and its components are defined based on a theoretical perspective and previous work. An algorithm is designed and implemented as a CASE tool named AGUML in order to process the scenario text using NLP and semantics. Then, the accuracy of the algorithm is presented with the analysis of performed experimental results.

### 3.1 Use Case and Scenarios

Scenarios are a powerful solution to the complexity of software development. They are used as telling stories to help system stakeholders share a complete view about the software being developed and to help in avoiding any missing problems [23]. The scenario is defined as: "one sequence of events that is one possible pathway through a use case" [24]. Therefore, a use case may include one or more scenarios. A scenario can be used as a tool for determining and validating software requirements. There is a variety of scenario descriptions used in the literature. However, the description of a scenario should include at least the following parts [25]-[26]:

- Initial Assumption: This part includes a description of software users' expectations at the start point of the scenario.
- Basic Path: This part depicts the normal sequence flow of events as expected by software users. It may also be called "Sunny Day Scenario" or "Happy Path".
- Alternate Path: This part includes the alternative set of steps that may be produced as a result of testing events of the basic path. These steps always run in parallel with the basic path. It may also be called "Rainy Day Scenario" or "Unhappy Path".
- Exception Path: This part includes output results for unsuccessful steps in the basic path.
- System State on Completion: This part includes the description of the system state at the end point of the scenario.

The author in [27] observed that developers often think of application through a relatively small number of typical interactions with users, such as the following:

1) User does …
2) Application does …
3) Application does …
4) User does …
5) Application does …
6) etc.

And these are called use cases, which are often used in conjunction with OO analysis and design methods. Use cases can be used as a starting point of requirements' analysis in order to derive classes from them [27]. A use case is always identified by its name, the actor which is the user type of the software and the interaction between the actor and the software. For instance, "Add new student" can be a typical use case for the course registration system with the "Registrar" as an actor. This use case may include the following sequence of steps:

1) Application shows an options' screen.
2) Registrar selects addition from options' screen.
3) The system presents a student entry screen.
4) Registrar enters student data.
5) Registrar submits student data.
6) System saves student data into database.

As seen from the above scenario example, each step in the scenario represents an English sentence of the form "Subject and Predicate".

### 3.2 Algorithm Design

We have designed an NLP algorithm that has a capability of analyzing scenario-based requirements written in restrictive English text to extract objects, their interactions and the messages passed among them in order to draw UML sequence and class diagrams. The following rules should be met before applying the NLP algorithm:

- The scenario is written as a sequence of English language sentences, where the structure of the scenario represents a typical normal flow of interactions between the user and the software, as stated in the previous subsection.
- Each sentence is separated from the next one by the normal end of line.
- Each sentence represents an action occurring in the software. This action can usually be initiated

externally by the user or internally by an object inside the software and appears in the active voice format.

The pseudocode of the proposed NLP algorithm is shown in Figure 1 and the main steps are described as follows:

1) At the beginning, the algorithm reads the scenario text, then it normalizes it (line 2). The normalization process produces a uniform word for some other different words that are equivalent. For example, replace any word in the set: "application", "applications", "system", "systems", "software" and "package" with a uniform word: "application". This is because these different words are considered equivalent based on user intention of software requirements.

Then, the algorithm declares and initializes some buffers to be used in storing important information during scenario processing (line 3), where, "umlSequenceScript" and "umlClassScript" buffers are used to store a specialized script as text for the whole scenario to be utilized later in drawing the needed UML diagrams. The "actor" buffer is used to store the name of actor as text if it exists. As a rule, there is only one actor for each scenario.

```
1.  Read the scenario text
2.  Normalize the scenario text
3.  Initialize and Set umlSequenceScript = "", umlClassScript = "",  actor = ""
4.  FOR each sentence in the normalized scenario text
5.  Produce a list of tokens
6.      Produce a parsed tree (pTree) from tokens
7.      If (pTree) has S tree (sTree)
8.          and (sTree) has NP (npTree) and VP (vpTree) subtrees respectively
9.          and (vpTree) has at least one NP subtree
10.             Initialize and Set parts[] = {"", "", "", ""}
11.             Get all words that starts with NN label from (npTree)
12.                 Then concatenate them and set them to parts[0]
13.             Get tagged word that labeled with NNP from (npTree) and Set it to actor
14.             Get word labeled starts with VB from (vpTree) and set it to parts[2]
15.             If (vpTree) has only NP labeled tree
16.                 Get all words that starts with NN label from (vpTree)
17.                     Then concatenate them and set them to parts[1]
18.             END IF
19.             If (vpTree) has two NP labeled trees
20.                 Get all words that starts with NN label from the first child of (vpTree)
21.                     Then concatenate them and set them to parts[3]
22.                 Get all words that starts with NN label from the second child of (vpTree)
23.                     Then concatenate them and set them to parts[1]
24.             END IF
25.     END IF
26.     Construct uml script for sequence from parts[] and add it to umlSequenceScript
27.     Construct uml script for class from parts[] and add it to umlClassScript
28. END FOR
29. IF Actor is not empty
30.     Add it as a prefix to umlSequenceScript
31. END IF
```

Figure 1. Algorithm pseudo-code for automatic generation of UML diagrams.

2) The algorithm then splits the scenario into sentences and for each individual sentence repeats important processing steps (lines 4-28) as follows:

The loop starts by tokenizing the sentence. For example, the output of the scenario sentence: "The user

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 07, No. 02, June 2021.

logs into the system" will be tokenized into: [The] [user] [logs] [into] [the] [system]. Then, morphological analysis is applied on each tokenized statement to identify the different parts of speech (POS). In POS process, each word in the sentence is tagged or labeled individually such as: (The/DT) [user/NN] [logs/VBZ] [into/IN] [the/DT] [system/NN], where "DT" means determiner, "NN" means noun, singular or mass, "VBZ" means verb or third person singular present and "IN" means preposition or subordinating conjunction. Then, for semantic analysis, a syntactical and lexical analysis is performed to produce a parsed tree form such as the following which is a typical output from Stanford CoreNLP [28] parser which is a suite of NLP software tools:

```
(ROOT
   (S
      (NP (DT The) (NN user))
      (VP (VBZ logs)
         (PP (IN into)
            (NP (DT the) (NN system))))))
```

When parsing a sentence, its parts are labeled at clause level such as "S" which means simple declarative clause and at phrase level such as "NP", "VP" and "PP" which mean noun phrase, verbal phrase and prepositional phrase, respectively. Based on the used tagger, leaf nodes of the parsed tree such as "logs" have a depth of zero, tagged words such as "VBZ logs" have a depth of 1 and phrasal nodes such as "VP (VBZ logs)" have a depth greater than or equal 2.

Then, for each separate parsed sentence from the scenario, the algorithm checks whether it includes a simple sentence structure of "Subject" part and "Predicate" part. The subject includes a noun or a pronoun in addition to the words describing it. The predicate part includes the verb in addition to other words telling more about the subject. Furthermore, a "simple subject" should be included in the subject part as noun and a "simple predicate" should be included in the predicate part as verb. So, the parsed sentence must be a simple declarative clause "S" and include a noun phrase "NP" followed by verbal phrase "VP" and the verbal phrase "VP" must include at least one noun phrase "NP" (lines 7-9). If the parsed sentence does not satisfy this rule, the algorithm marks it as invalid, informs the system user to change the sentence structure and continues processing the subsequent steps. The valid structure of the parsed sentence helps in identifying the main parts of the sentence, such as objects, operations and messages exchanged among objects.

Then the algorithm declares and initializes the "parts[]" buffer, which is an array of four elements of type text to store data about participating objects, operations performed and messages passed among objects. The data of "parts[]" provides important information for producing a specialized script that will be added to "umlSequenceScript" and "umlClassScript" buffers.

After that, a semantic analysis is performed in order to identify the main parts of the parsed sentence, such as the actor that initiates the scenario and the participating objects, their interactions and the named messages exchanged among them during their interaction. There are usually two participating objects; one is named "caller" that makes a request and the other is named "receiver" that receives the request from the former. Determining the interactions among objects is also defined in this step. The type of interaction that is always defined as an operation between objects is analyzed in addition to type of data passed among these objects.

To define the "caller" object, the labeled "NP" subtree of the main parsed tree "S" is searched for words the tags of which start with "NN" label, then their concatenation is stored in "parts[0]" buffer. The searched labels are "NN", "NNS", "NNP" and "NNPS" which represent singular, plural, singular proper and plural proper nouns, respectively. It is important to note that there may be more than one word that describes the "caller" object such as "main screen" (lines 11-12). The "NP" subtree is also searched for a word the tag of which starts with "NNP" label to consider it as an actor and if it is found, it will be stored in "actor" buffer (line 13). Note that the actor is set once for the overall scenario which always represents a role that initiates the scenario.

Then, the labeled "VP" subtree of the main parsed tree "S" is firstly processed to capture the operation between the "caller" and the "receiver" objects by searching a word the tag of which starts with "VB" label and storing it in "parts[2]" buffer (line 14). The labels are "VB", "VBD", "VBG", "VBN", "VBP" and "VBZ" which represent base form, past tense, gerund or present participle, past participle, non-3$^{rd}$

person singular present and 3rd person singular present verbs, respectively. After that, if the "VP" subtree of the main tree "S" includes only one "NP" subtree, search it for the "NN" tagged words, consider their concatenation as a "receiver" object and store it in the "parts[1]" buffer. However, if it includes more than one "NP" subtree, consider the "NN" tagged words of the first "NP" as the message passed between the interacting objects, store their combination in "parts[3]", consider the "NN" tagged words of the second "NP" as the "receiver" object and store their combination in "parts[1]".

3) After the loop of sentence processing finishes, the four stored elements in "parts[]" buffer, which are the "caller" object, the "receiver" object, the operation and the message, are used to construct a one-line specialized script that will be added to "umlSequenceScript" and "umlClassScript" buffers. This script will be fed later to the PlantUML [29] plugin for generating the required UML diagrams. This script has the format of "caller -> receiver : operations message". For example, the result of processing this scenario sentence "Registrar selects addition from options' screen" is "Registrar -> OptionsScreen : Selects Addition". This script structure follows the format invented by the PlantUML tool.

4) At the end of processing the whole scenario, if an actor is found, it will be added as a prefix to the "umlSequenceScript" buffer content. As an example, for the following scenario:

> Student selects "register for a course" from main menu.
> Application retrieves courses from database.
> Software presents courses into courses list window.
> Student selects courses from the courses list window.
> Student submits the selected courses.
> Application saves selected courses into database.

After processing the scenario with our proposed algorithm, the complete script for the sequence diagram is as follows:

```
Actor Student
Student -> MainMenu : Selects RegisterCourse
Application -> Database : Retrieves Courses
Application -> CoursesListScreen : Shows Courses
Student -> CoursesListScreen : Selects Courses
Student -> SelectedCourses : Submits
Application -> Database : Saves SelectedCourses
```

And the complete script for the class diagram is as follows:

```
Student o-- MainMenu
MainMenu : Selects ()
Application o-- Database
Database : Retrieves ()
Application o-- CoursesListScreen
CoursesListScreen : Shows ()
Student o-- CoursesListScreen
CoursesListScreen : Selects ()
Student o-- SelectedCourses
SelectedCourses : Submits ()
Application o-- Database
Database : Saves ()
```

It is important to denote that the sequence diagram is considered as the basis for drawing the class diagram.

## 4. SYSTEM DESIGN AND IMPLEMENTATION

We have developed a system for implementing the previously stated algorithm that essentially processes scenario-based requirements written in English text, then automatically draws UML diagrams. The system is named AGUML and has a capability of reading scenario-based user requirements. It also has a capability of applying NLP on this text and making syntactical and semantical analysis to produce UML diagrams. The main components of the system architecture are shown in Figure 2. Descriptions of the GUML system components are in the following subsections.

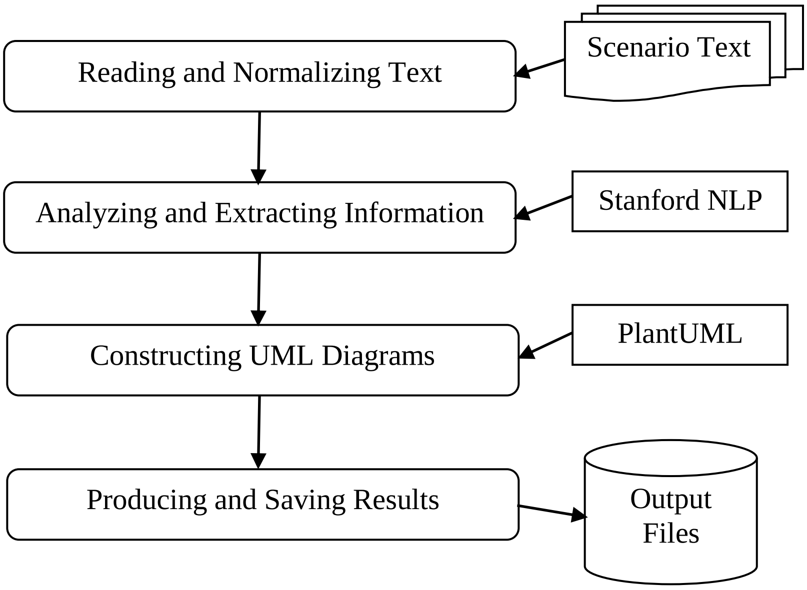



Figure 2. AGUML system architecture.

### 4.1 Reading and Normalizing Scenario Text

This component takes a scenario from user as text input. The user can directly write the scenario text or retrieve it from an existing file. Then, the text is normalized to convert the equivalent words into a uniform format.

### 4.2 NLP of Normalized Text

It firstly separates each sentence of the scenario text, produces POS tagging and parses this sentence to produce a tree in order to help in syntactic and semantic recognition. This component also checks the validity of sentence structure by deciding whether it conforms to the form of noun phrase followed by a verbal phrase and the verbal phrase has at least one noun phrase. And hence, the user will be informed of invalid scenario sentences.

### 4.3 Analyzing and Extracting Information

This component takes the tagged and parsed text as input from the previous component, then defines the objects, their interactions and the type of data passed among them. It mainly analyzes the noun phrase and the verbal phrase in each parsed scenario text and the tagged words of each phrase. The actor of the scenario is defined in this module if it exists. If the NLP results produce more than one actor, only one equivalent uniform actor will be assigned, because the scenario is always initiated by one actor. Actor can be identified by searching proper noun in the first part (noun phrase) of the scenario sentence. The analysis and extraction of this important information is achieved with the help of Stanford CoreNLP software tools.

### 4.4 Constructing UML Diagrams

This component takes information about diagrams from the previous component and produces a specialized script. This script is next processed by the included PlantUML third-party component to draw sequence and class diagrams.

### 4.5 Producing and Saving Results

This module helps the user save the updated scenario text, sequence diagram and class diagram in order to be used for documenting purposes that will support the subsequent tasks of the software development process. It also permits for updating these results later.

We have implemented the system using Java programming language with the help of Eclipse IDE [30].

## 5. Experimental Results

This section provides the analysis of our experimental results. The experiments were mainly conducted to check the accuracy of our designed algorithm that has been implemented in the AGUML system. The main window of AGUML software tool is shown in Figure 3. Using this window, the user can write a scenario as separate sentences of English text or can load it from previously stored text file. When pressing the button "Update UML", our software tool will process the scenario text and produce the UML diagrams automatically. AGUML can show sequence diagram, class diagram or both of them simultaneously based on user selection. Figure 3 shows how sequence diagram appears after processing a scenario. The actor appears on the left side of the diagram and it is represented using stick person symbol. The "caller and "receiver" objects appear on the top and on the bottom of the diagram and they are represented as annotated rectangles. The name of the operation and the message passed between objects are represented using leftward arrow symbol that links the "caller" object and the "receiver" object.

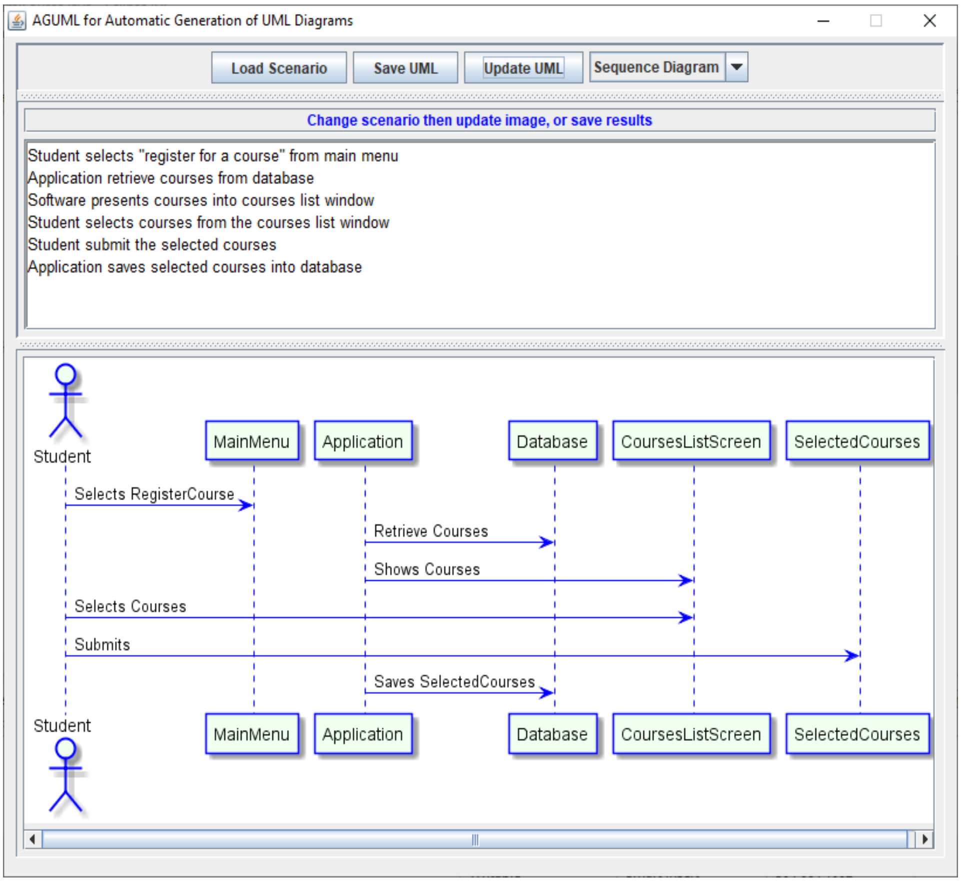



Figure 3. Generation of UML sequence diagram from scenario text.

The class diagram of the same scenario is shown in Figure 4. As said previously, the components of the class diagram are constructed depending on the developed sequence diagram. So, the same classes identified in the sequence diagram will also appear in the new class diagram. Furthermore, the operations between objects shown in the sequence diagram will be added as operations inside the classes of the class diagram. They are actually added inside the class which provides the functionality for the requesting class; in other words, in the class where the head of leftward arrow ends in the sequence diagram.

AGUML also can provide the user with the ability of storing important results into files in order to be used in software documentation as in well as any future requirements updates. These results include updated scenario text, UML-generated diagrams and UML input script. Furthermore, when processing

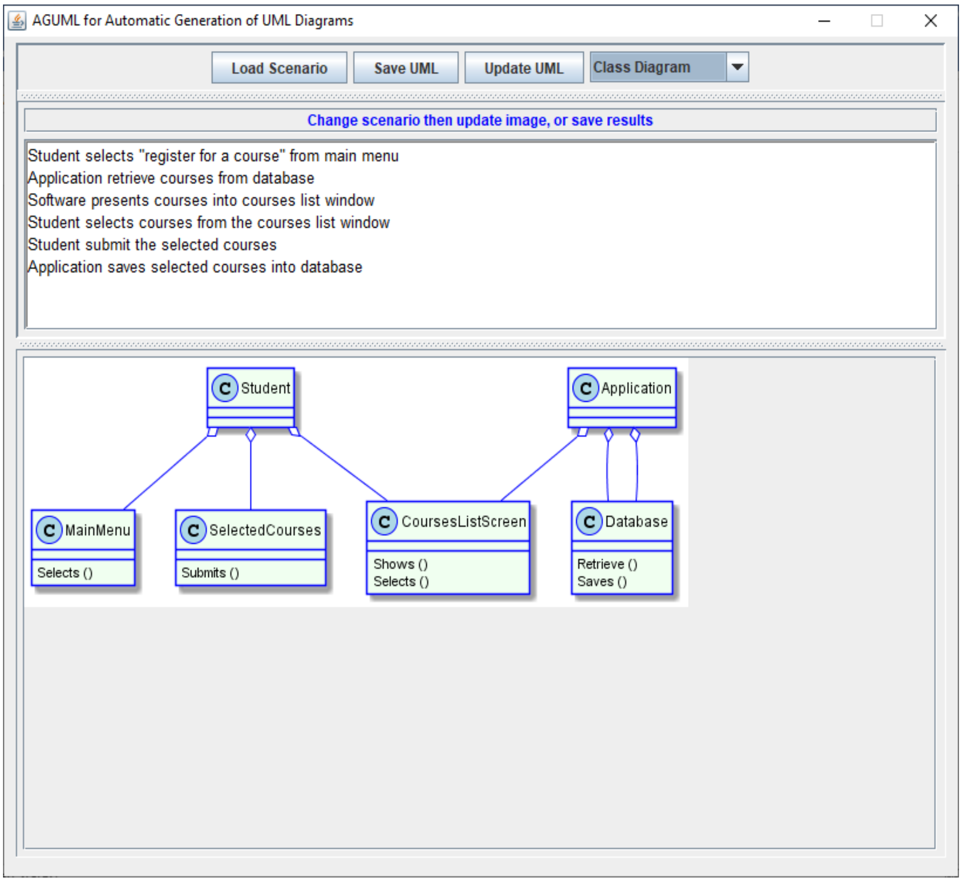



Figure 4. Generation of UML class diagram from scenario text.

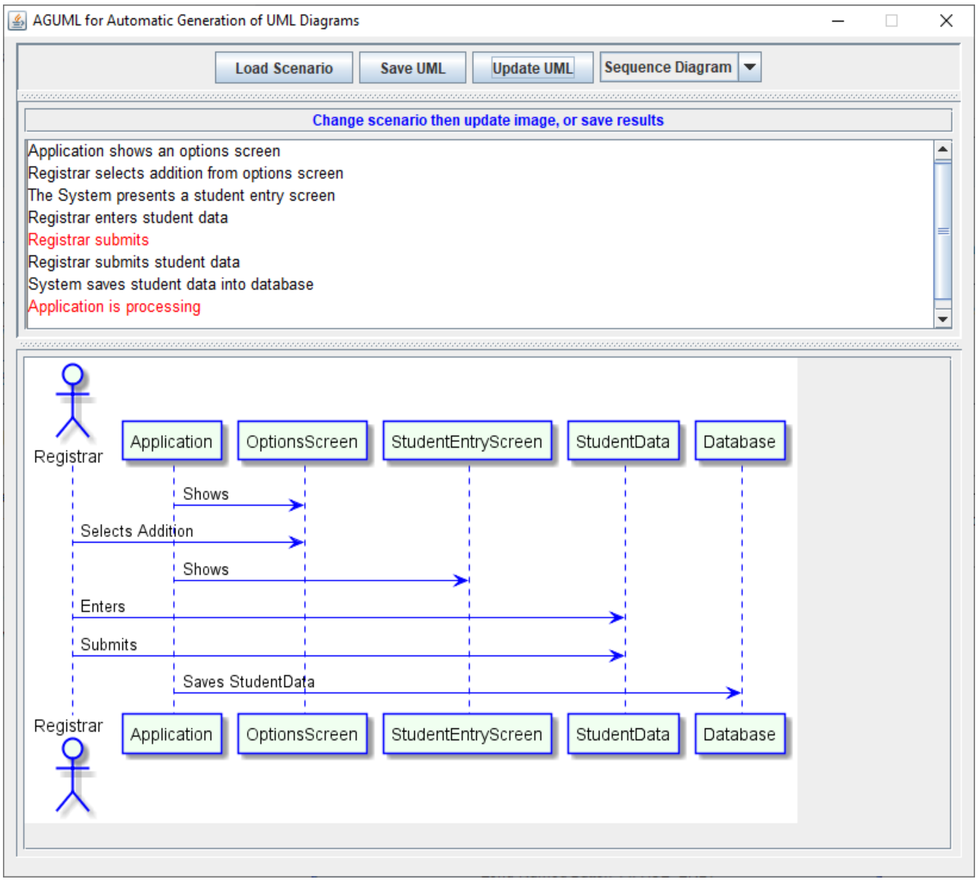



Figure 5. Invalid scenario sentences are shown in red text.

scenario text, AGUML can notify user with any invalid English sentence in order to help user in correcting the invalid sentence, as shown in Figure 5. However, AGUML has the ability of generating UML diagrams for only valid scenario sentences.

To assess the accuracy of AGUML and the designed algorithm, we have evaluated it against five different real-life scenarios collected from different sources in the literature. The first scenario describes "add student" use case and the second scenario describes "register for a course" use case and they are part of defined requirements for educational and registration services in a university [4]. The third scenario describes "check out item" use case and the fourth scenario describes "view file content" use case and they are part of defined requirements for videos' store services [5]. The fifth scenario defines the steps for verifying user by login name and password [31]. Each scenario includes a hybrid of simple, average and complex English sentences and they are labeled from "SC1" to "SC5" in Table 1.

After checking the sentence validity for each scenario, we have considered five types of criteria for evaluating AGUML which are actor identification, "caller" object identification, "receiver" object identification, operation identification and message identification.

Firstly, we have developed sequence diagrams manually for each scenario based on our experience and provided counts for each one of the five elements in each individual diagram. These counts are found inside the columns labeled with "M" in Table 1. Similarly, we have provided counts for each of the five elements in each individual scenario generated automatically by AGUML, where their identifications match correctly the manual identifications. The counts of automatic identifications are found inside the columns labeled with "A" in Table 1. The accuracy is then calculated by dividing the count of automatic generation by the count of manual generation for each element [14].

Table 1. Accuracy results of AGUML for automatic generation of UML diagrams.

| **Scenario** | **SC1** | | **SC2** | | **SC3** | | **SC4** | | **SC5** | | **Accuracy** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Element** | M | A | M | A | M | A | M | A | M | A | |
| Actor | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 80% |
| Caller object | 3 | 3 | 2 | 2 | 2 | 2 | 2 | 1 | 2 | 2 | 90% |
| Receiver object | 5 | 5 | 4 | 4 | 5 | 3 | 7 | 7 | 6 | 5 | 88% |
| Operation | 7 | 7 | 6 | 6 | 6 | 4 | 7 | 6 | 7 | 7 | 90% |
| Messages | 2 | 2 | 5 | 5 | 0 | 0 | 1 | 0 | 1 | 0 | 77% |

As shown in Table 1, the accuracy ranges from 77% to 90%, where the "caller" object identification and operation identification have the highest accuracy scores.

After reviewing the POS tree for some sentences, it is noticed that the adopted tagger tool produces tags for some words in a manner that is different from our intension. For example, it considers "user" in "The user enters a username and password" sentence of "SC5" and "SC4" as "NN" and hence, it is not considered as an actor. Also, the sentence "Clerk swipes bar code" of "SC3" was tagged as:

```
(ROOT
  (S
    (VP (VB Clerk)
      (NP (JJ swipes) (NN bar) (NN code)))))
```

This means that the tagger considers "Clerk" word as verb and "swipes" word as adjective and hence, the AGUML reports it as an invalid sentence. Moreover, "Application stores record" sentence of the same scenario was tagged as:

```
(ROOT
  (NP
    (NP (NN application) (NNS stores))
    (NP (NN record))))
  (NP (NP) (NP)) stores as NNS
```

And hence, the tagger does not consider it as an adequate sentence structure labeled with "S". Although the AGUML accuracy is affected by the tagger behavior in some cases as seen in these examples, it is generally accepted, especially in the case of building UML sequence diagrams.

## 6. CONCLUSION AND FUTURE WORK

In this paper, we have proposed and developed a system called AGUML for automatic generation of UML diagrams from scenario-based user requirements written in English natural language. We have designed an algorithm and implemented it in the AGUML. It has a capability of producing syntactical, lexical and semantic analysis for extracting important information for identifying actors, classes/objects and their interactions as well as the messages exchanged among them. The main purpose of AGUML is to facilitate work performed in the analysis and design activities for the software being developed. AGUML has an interactive user interface to help user write a scenario directly or upload it from an existing file. Then, AGUML can process the scenario and produce the UML diagrams automatically. The resulting diagrams can be stored easily in files in order to be used later for the purposes of software documentation. The accuracy of AGUML and the designed algorithm has been tested using five different scenarios.

An extension to this work is needed for enhancing the designed algorithm in order to permit the user for writing less restrictive scenario text and considering other variants of scenario main flow such as alternative flow. The capability of our proposed system can be also enhanced in order to produce other UML diagrams.

## REFERENCES

[1] D. K. Deeptimahanti and M. A. Babar, "An Automated Tool for Generating UML Models from Natural Language Requirements," Proc. of the IEEE/ACM Int. Conf. on Automated Soft. Eng., pp. 680-682, Auckland, New Zealand, 2009.

[2] I. F. Alexander and N. Maiden, Scenarios, Stories, Use Cases: Through the Systems' Development Life-cycle, ISBN: 978-0-470-86194-3, John Wiley & Sons, 2005.

[3] H. Gomaa, Software Modeling and Design: UML, Use Cases, Patterns and Software Architectures, DOI: 10.1017/CBO9780511779183, Cambridge University Press, 2011.

[4] T. Quatrani, Visual Modeling with Rational Rose 2002 and UML, 3rd Edition, Pearson Education India, ISBN-10: 0201729326, 2002.

[5] E. J. Braude, Software Design: From Programming to Architecture, John Wiley and Sons, 2004.

[6] R. Elmansouri, S. Meghzili, A. Chaoui, A. Belghiat and O. Hedjazi, "Integrating UML 2.0 Activity Diagrams and Pi-Calculus for Modeling and Verification of Software Systems Using TGG", Jordanian Journal of Computers and Information Technology (JJCIT), vol. 06, no. 04, pp. 326-344, Dec. 2020.

[7] R. J. Abbott, "Program Design by Informal English Descriptions," Communications of the ACM, vol. 26, no. 11, pp. 882-894, 1983.

[8] E. Buchholz, A. Düsterhöft and B. Thalheim, "Capturing Information on Behavior with the RADD-NLI: A Linguistic and Knowledge-based Approach," Data & Knowledge Engineering, vol. 23, no. 1, pp. 33-46, 1997.

[9] S. Nanduri and S. Rugaber, "Requirements' Validation *via* Automated Natural Language Parsing," Journal of Management Information Systems, vol. 12, no. 3, pp. 9-19, 1995.

[10] A. Oliveira, N. Seco and P. Gomes, "A CBR Approach to Text to Class Diagram Translation," Proc. of the TCBR Workshop at the 8th European Conference on Case-based Reasoning, Turkey, Sep. 2006.

[11] H. M. Harmain and R. Gaizauskas, "CM-builder: A Natural Language-based Case Tool for Object-oriented Analysis," Automated Software Engineering, vol. 10, no. 2, pp. 157-181, 2003.

[12] S. P. Overmyer, L. Benoit and R. Owen, "Conceptual Modeling through Linguistic Analysis Using LIDA," Proceedings of the 23rd IEEE Int. Conf. on Soft. Eng. (ICSE 2001), pp. 401-410, May 2001.

[13] M. Clavel, M. S. E. González and V. T. da Silva, "The MOVA Tool: A Rewriting-based UML Modeling, Measuring and Validation Tool," Proc. of the XII JISBD, pp. 393-394, Spain, 2007.

[14] I. S. Bajwa, A. Samad and S. Mumtaz, "Object Oriented Software Modeling Using NLP Based Knowledge Extraction," European Journal of Scientific Research, vol. 35, no. 1, pp. 22-33, 2009.

[15] M. Ibrahim and R. Ahmad, "Class Diagram Extraction from Textual Requirements Using Natural Language Processing (NLP) Techniques," Proc. of the 2nd IEEE Int. Conf. on Computer Research and Development, pp. 200-204, Kuala Lumpur, Malaysia, May 2010.

[16] H. Herchi and W. B. Abdessalem, "From User Requirements to UML Class Diagram," Proc. of the

International Conference on Computer Related Knowledge, arXiv Preprint arXiv:1211.0713, 2012.

[17] P. More and R. Phalnikar, "Generating UML Diagrams from Natural Language Specifications," International Journal of Applied Information Systems, vol. 1, no. 8, pp. 19-23, 2012.

[18] H. G. Perez-Gonzalez and J. K. Kalita, "Automatically Generating Object Models from Natural Language Analysis," Proc. of the Companion of the 17th Annual ACM SIGPLAN Conf. on Object-oriented Programming, Systems, Lang. and Appl. (OOPSLA '02), pp. 86-87, Seattle, USA, Nov. 2002.

[19] M. Elallaoui, K. Nafil and R. Touahni, "Automatic Transformation of User Stories into UML Use Case Diagrams Using NLP Techniques," Procedia Computer Science, vol. 130, pp. 42-49, 2018.

[20] S. Jabbarin and N. Arman, "Constructing Use Case Models from Arabic User Requirements in a Semi-automated Approach," Proc. of the IEEE World Congress on Computer Applications and Information Systems (WCCAIS), pp. 1-4, Hammamet, Tunisia, Jan. 2014.

[21] M. Elallaoui, K. Nafil and R. Touahni, "Automatic Generation of UML Sequence Diagrams from User Stories in Scrum Process," Proc. of the 10th IEEE International Conference on Intelligent Systems: Theories and Applications (SITA), pp. 1-6, Rabat, Morocco, Oct. 2015.

[22] N. Alami, N. Arman and F. Khamayseh, "Generating Sequence Diagrams from Arabic User Requirements Using MADA+ TOKAN Tool," The International Arab Journal of Information Technology, vol. 17, no. 1, pp. 65-72, Jan. 2020.

[23] I. Alexander and N. Maiden, "Scenarios, Stories and Use Cases: The Modern Basis for System Development," Computing and Control Engineering, vol. 15, no. 5, pp. 24-29, 2004.

[24] A. G. Sutcliffe, N. A Maiden, S. Minocha and D. Manuel, "Supporting Scenario-based Requirements' Engineering," IEEE Transactions on Software Engineering, vol. 24, no. 12, pp. 1072-1088, 1998.

[25] Spark Systems, "Writing Use Case Scenarios for Model Driven Development," [Online], Available: https://sparxsystems.com/downloads/quick/writing-structured-use-case-scenarios-mdd.pdf, 2010.

[26] I. Sommerville, Software Engineering, 10th Edition, Addison-Wesley, 2015.

[27] I. Jacobson, Object-oriented Software Engineering: A Use Case Driven Approach, Pearson Education India, 1993.

[28] Stanford CoreNLP, "A Suite of Core NLP Tools," [Online], Available: http://stanfordnlp.github.io/CoreNLP.

[29] PlantUML, [Online], Available: http://plantuml.com.

[30] Eclipse Foundation, [Online], Available: https://www.eclipse.org.

[31] G. Schneider and J. P. Winters, Applying Use Cases: A Practical Guide, Pearson Education, 2001.

**ملخص البحث:**

تعدّ أدوات النّمذجة الفعالة للبرمجيات ضرورية جداً للتّحقيق النّاجح لأنشطة هندسة البرمجيات، وبخاصة في مرحلة التّحليل والتّصميم. وإنّ أتمتة هذه الأدوات تسهل العمل، وتجعله أعلى إنتاجيةً، وتخفّض زمن التّطوير وكلفته. وتهدف هذه الورقة الى تطوير طريقةٍ والتحقق منها -بالإضافة الى أداةٍ برمجية- للتّوليد الاتوماتيكي لمخططات لغة النّمذجة الموحدة عند اتباع النّهج الموجَّه نحو الهدف في التطوير. ويتم توليد هذه المخططات من المتطلبات المبنيّة على السيناريوهات؛ من أجل تسهيل عملية النّمذجة. وهكذا يتم تحديد نموذجٍ من المتطلبات المبنية على السيناريوهات ومكوناته وبناؤها. ثم يتم تصميم طريقةٍ تتضمن خوارزمية وتطبيقها بناءً على معالجة اللغات الطبيعية لتوليد مخططات لغة النّمذجة الموحدة أوتوماتيكياً من المتطلبات المبنية على السيناريوهات، وتشمل مخططات التتابع ومخططات الصِّنف. وجرى التحقق من قدرة الطريقة المقترحة وأدائها وفوائدها، وكذلك الأداة البرمجية، عن طريق نتائج تجارب عملية أجريت لهذا الغرض.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 07, No. 02, June 2021.

# RECOGNITION OF ARABIC HANDWRITTEN CHARACTERS USING RESIDUAL NEURAL NETWORKS

Ahmad T. Al-Taani* and Sadeem T. Ahmad

*(Received: 8-Mar.-2021, Revised: 8-May-2021, Accepted: 22-May-2021)*


## *ABSTRACT*

*This study proposes the use of Residual Neural Networks (ResNets) to recognize Arabic offline isolated handwritten characters including Arabic digits. ResNets is a deep learning approach which showed effectiveness in many applications more than conventional machine learning approaches. The proposed approach consists of three main phases: pre-processing phase, training the ResNet on the training set and testing the trained ResNet on the datasets. The evaluation of the proposed approach is performed on three available datasets: MADBase, AIA9K and AHCD. The proposed approach achieved accuracies of 99.8%, 99.05% and 99.55% on these datasets, respectively. It also achieved a validation accuracy of 98.9% on the constructed dataset based on the three datasets.*



## *KEYWORDS*

*Residual networks, Deep learning, Deep neural networks, Arabic handwritten characters, Characters recognition.*


## 1. INTRODUCTION

Optical Character Recognition (OCR) is an electronic conversion of images of printed/handwritten text into computer-encoded text. Handwritten OCR system is divided into online and offline recognizers based on the input method. Online data is made through devices, such as tablets, computer mouse or electronic pen, while offline data is collected from scanned images of typed/handwritten documents. The recent OCR approaches are mainly applying conventional machine learning or deep learning approaches. Conventional machine learning approaches, such as Multi-layer Perceptron (MLP) and Support Vector Machine (SVM) approaches require expert engineers and specialists.

OCR is a multidisciplinary area of research in artificial intelligence, computer vision and pattern recognition. OCR is frequently used as a day of data entry from printed papers, like invoices, passport documents, mails, …etc. OCR is also a common method of digitizing printed texts, so that they can be electronically edited, searched, …etc. OCR is used in many applications, like machine translation, text mining and cognitive computing.

Processing Arabic language has lately gained attention from scholars with the increase in Arabic scholars. It became necessary all over the world and is almost digitizing massive amounts of information that is daily being processed. An example of this information comprises medical records, license plate recognition, checks to verify and old documents for digital libraries. Isolated characters have more complicated features to detect and different shapes for each letter based on the context of the character. The total number of expanded characters reaches 84 shapes that are made of the 28 basic letters.

Several studies that used deep neural networks for Arabic character recognition (Convolutional Neural Networks (CNNs), Deep Belief Networks (DBNs), Recurrent Neural Networks (RNNs)) gave promising results. Convolutional approaches automatically extract features from raw images. CNN-based architecture provides an end-to-end solution without the need to have a handcrafted-feature extraction or data representation transformation in contrast to many different conventional approaches [1]. CNN OCR systems must include four essential components: convolutional layer, pooling layer, fully connected layer and loss function that is added in the last layer. Such systems provide an effective performance by applying a drop-out of layers and control the size of the CNN.

One of the first CNN architectures was the LeNet-5 architecture which was introduced by Lecun et al.[2] and is primarily implemented for the OCR system of handwritten zip code digits by the U.S. postal

A. T. Al-Taani* and S. T. Ahmad are with the Department of Computer Science, Yarmouk University, Irbid, Jordan. Emails: ahmadta@yu.edu.jo and ahmadta@yu.edu.jo
*A. T. Al-Tanni is currently with the Department of Data Science and Artificial Intelligence, Princess Sumaya University for Technology, Amman, Jordan. Email: a.altaani@psut.edu.jo

services data and is also applied in face recognition systems. It consists of five layers, including two convolutional layers, two pooling layers and finally, a fully connected layer. Another popular architecture called AlexNet was constructed by Krizhevsky et al. [3]. It is considered one of the first networks that comprise a sort of depth. It is composed of 5 convolutional layers, 3 pooling layers and the last convolutional layer is followed by three fully connected layers.

Wu et al. [4] proposed a model called the Directly Connected Convolutional Neural Networks (DCCNNs) model. The obtained results pertaining to their proposed model are compared to the results related to the previous models, where it is proved to take a less computational time for recognition and training large images/datasets. The comparison consists of five different datasets, including the MNIST dataset for isolated handwritten digits. By using the MNIST dataset, an accuracy of 98.96% is slightly improved and is 1.3 to 1.4 times faster than the conventional CNNs.

Ashiquzzaman and Tushar [5] proposed an offline Arabic numeral recognizer by applying CNNs. They also modified the Multi-layer Perception (MLP) by applying a dropout regularization for solving the overfitting problem. The proposed approach is trained and tested on the CMATERDB 3.3.1 Arabic handwritten digit dataset. The proposed approach achieved an accuracy of 97.4%, while the modified MLP approach achieved an accuracy of 93.8%.

Mudhsh & Almodfer [6] proposed the VGGnet architecture for Arabic OCR handwritten alphanumeric characters. Their model was applied to the MADBase database with an accuracy of 99.66% and to the HACDB database with an accuracy of 97.32%.

Younis & Alkhateeb [7], Tomimori et al. [8] and Eladel et al. [9] introduced a handwritten digital recognition model by using CNNs. They modified the architecture of the network based on their own experiments and trials. Younis & Alkhateeb [7] created a simple CNN architecture for Arabic handwritten digit recognition and face recognition models. To train their model, they used MNIST dataset, which achieved an accuracy of 98.11%.

Eladel et al. [9] proposed an approach that has an impact on improving the classification accuracy and testing speed. Their results for the MNIST achieved an accuracy of 95.7% and the CIFAR-10 achieved an accuracy of 99.71%.

Younis [10] presented a deep neural network-based handwritten Arabic character recognition system. ResNet-18 architecture was applied with batch normalization for regularization and dropout to prevent overfitting. He obtained recognition accuracies of 94.8% with AHCD database and 97.6% with AIA9k.

Elleuch et al. [11] investigated the Deep Belief Neural Networks (DBNNs) approach for Arabic handwritten character/word recognition. The DBNN approach is trained and tested on the HACDB and IFN/ENIT databases. The obtained results of the two experiments showed that a 2.1% error rate resulted for characters, but for words, the error rate exceeded 40% concluding that the proposed DBNN approach is still unready to deal with high-level dimensional data.

Tagougui & Kherallah [12] proposed a model that consisted of the DBN approach and BottleNeck feature classifier for the Arabic handwritten character OCR. The LMCA database was used for training and testing. The experimental results showed that the proposed approach outperformed some previous approaches.

Karthik and Srikanta [13] proposed the Deep Belief Network (DBN) approach to recognize different handwritten Kannada characters. The experiment achieved an average accuracy of 95% by using raw pixels and an accuracy of 96.41% tested on a dataset consisting of 18,800 samples.

Recently, Mustafa and Elbashir [14] used the CNN for the recognition of Arabic names. The dataset (SUST-ARG) containing 8028 Arabic names was used for training and testing the proposed approach. Experimental results showed that the proposed approach has achieved an accuracy of 99.14%.

Another recent work on Arabic handwriting recognition was proposed by Ghanim et al. [15]. They proposed a multistage cascading approach for Arabic offline handwritten character recognition. The Hierarchical Agglomerative Clustering (HAC) algorithm was used to cluster and rank the dataset. Then, six different deep CNN approaches were used in the recognition process. The IFN/ENIT Arabic dataset was used to evaluate and compare the six deep CNN approaches. The proposed approach achieved promising results in terms of computation time and complexity, as well as classification results.

Most recent research on Arabic handwritten recognition is done by Altwaijry and Al-Turaiki [16]. In this study, they presented a novel dataset consisting of 47,434 Arabic characters written by children aged from 7 to 12 years. This dataset was used to train and test the proposed approach for the recognition of Arabic handwritten characters using CNN. The proposed approach achieved an accuracy of 88% on the AHCD and the Hijja datasets, respectively.

Akouaydi et al. [17] proposed the use of CNN based on Beta-elliptic parameters and fuzzy elementary perceptual codes for the recognition of Arabic online characters. An accuracy of 98.90% was achieved using two Arabic datasets; LMCA and MAYASTROUN.

According to the previous studies, it is confirmed that convolutional approaches perform more effectively when applying image recognition through different related approaches. In the case of offline/online character recognition, it is created in the form of 2D-vector, which is more suitable. The DBNs require extra efforts for transforming the data, followed by exceeding computational complexity. Results motivate to use one of their improved architectures, which are anticipated to be effective and appropriate for this research problem.

Table 1 gives a summary of the key literature analyzed in this research. This Table shows the relationships between the key studies on Arabic OCR as well as their limitations. The scope of these studies is the recognition of Arabic offline handwritten characters. It can be seen that [6] has achieved the highest accuracy on the MADBase dataset among other approaches, while [1] and [5] achieved the next best results. Approaches in [6] and [16] obtained lower efficiency compared to other approaches.

Table 1. A summary of key literature.

| **Ref.** | **Approach** | **Scope** | **Limitations** | **Results** |
|---|---|---|---|---|
| [1] | CNN | Offline Arabic handwritten character recognition | Depends on the hyper-parameters' tuning and the size of the dataset. | 97.32% on OIHAC dataset |
| [5] | CNN | Identifying offline handwritten numbers based on conducted experiments | The model is not enhanced more than the limit already enhanced. | 97.4% on CMATERDB dataset |
| [6] | CNNs / VGGnet | Arabic handwritten alphanumeric character recognition | The system is simple and generic and does not perform effectively for words. | 99.66% on MADBase dataset and 97.32% on HACDB dataset |
| [10] | ResNet-18 | Offline Arabic handwritten character recognition | Low efficiency compared to other approaches. | 94.8% on AIA9k dataset and 97.6% on AHCD dataset |
| [16] | CNN | Arabic handwritten character recognition | Small dataset and low efficiency compared to other approaches. | 88% on a small artificial dataset |

Arabic handwritten text recognition is one of the hot topics and challenging areas of the fields of pattern recognition and image processing. Deep neural networks as based on CNNs and DBNs show promising results in Arabic character/numeral recognition process in terms of accuracy and speed. Different machine learning approaches have been proposed for the recognition of Arabic characters. Most of these studies are realized to efficiently perform for smaller datasets. However, the critical issue that is encountered is that such approaches require an extensive effort from a domain expert engineer to design a feature extractor that transforms the raw data into an appropriate representation or feature vector from which a classifier could possibly recognize the input pattern [28]. In this study, we propose the use of ResNets for the recognition of Arabic handwritten characters using large standard datasets of isolated Arabic handwritten characters. ResNet is used in this paper, since it is one of the recent architectures of deep learning and not used yet for Arabic OCR. The significance of this paper is that it fills the gaps in recent related research and improves the classification rates of previous approaches. The implemented model deals with a higher-level data when being embedded to real applications which identify words rather than isolated characters only. It also could be used in the field of computer vision, such as handwriting recognition and natural object classification.

The rest of the paper is organized as follows: Section 2 presents the proposed methodology, experimental results are presented and discussed in Section 3 and conclusions drawn from this study and directions for future work and presented in Section 4.

## 2. METHODOLOGY

### 2.1 Overall Research Design

The proposed handwritten Arabic OCR approach using the ResNet architecture is presented. The first step is to conduct the dataset preparation and pre-processing that includes resizing, colour binarization, noise reduction and finally, classification into separate classes. The second step is to pass the dataset through to the deep ResNet and to train the network on the 76 determined classes including characters and digits. The final step is to test the trained network and evaluate its overall performance based on accuracy, precision and recall. Figure 1 shows the overall architecture of the proposed approach.

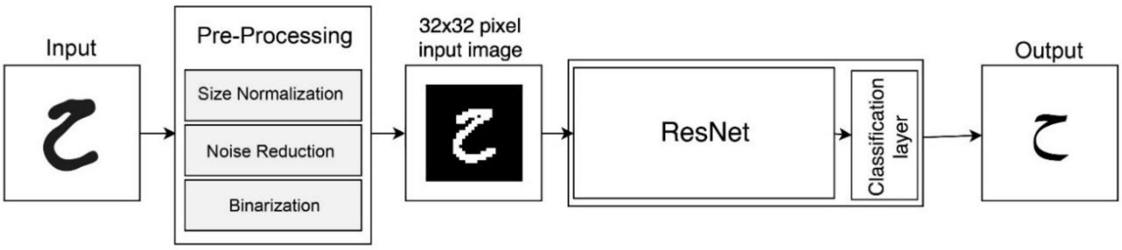



Figure 1. The overall architecture of the proposed approach.

Image pre-processing is performed for all characters in the dataset. This step includes noise reduction, color binarization, scaling and cropping the handwritten characters. To eliminate noise, thin features must be preserved as much as possible. Character edge pixels are identified as target pixels for the smoothing process, where such a process is only executed for the color difference. After that, the size of the target pixels is calculated for brightness and contrast operations. The final two steps are resizing the images to be of the size 32×32 pixels and the color depth of the images must be unified and should be black or white.

The Residual Neural Network (ResNet) is defined as a modularized architecture. An example of a generic ResNet, which stacks units with a similar connection shape, is proposed by He et al. [18]. ResNets won the 1st place in the ILSVRC 2015 ImageNet classification competition with an error rate of 3.57%. It consists of a set of residual blocks that are expressed in Equations 1 and 2.

$$y_1 = h(x_I) + F(x_I, W_I) \tag{1}$$

$$x_{I+1} = f(y_I) \tag{2}$$

where $x_I$ and $x_{I+1}$ denote the input and output of the i$^{th}$ unit which represents the feature value, $F$ denotes the residual function, $h(x_I)$ represents the identity mapping and $f$ denotes the activation function. Convolutional layers are combined with a down-sampling layer, then with an activation function. The Rectified Linear Unit (ReLU) is defined as a popular used activation function and is formulated in Equation 3.

$$f(x) = \max(0, x) \tag{3}$$

Pre-processed input character image 32×32 pixels are considered to form a 2D-vector and by keeping the spatial order of pixels, they can pass through to the pre-trained ResNet units. The maps of features are generated by convolutions with feature extraction kernels of 3×3.

After each stack of residual blocks, a pooling layer is attached. L2-pooling is a way of summarizing information from the convolutional layer by taking the square root of the squares' summation of the activations in a 3×3 region. It is used after each few stacks of residual blocks in order to reduce complexity [19].

Residual block is a special feature and identity mapping of the ResNet. It helps with training the network more effectively and faster, as it is also anticipated to simplify learning. Additionally, it solves the optimization problem by avoiding zero mapping and model closer to identity [20].

It is used only with input/output layers of the same dimension. Rather than expecting stacked layers to approximate $H(x)$, by Equation 4, layers are approximated by applying residual functions.

$$F(x) = H(x) - x \tag{4}$$

The residual block basically consists of a shortcut connection and a sequence of layers including convolutional layers. There exist a variety of forms of residual blocks, including two specific types: the standard block and the bottleneck block. Based on the implementation performed by He et al. [21], after each convolutional layer, a Batch Normalization (BN) layer is attached. The BN is adopted so that dropout is eliminated. Ioffe & Szegedy [22] studied the effect of the BN, which acts as a regulator speeding up training and reducing the over-fitting problem.

When building a residual block, the effect of depth and width must be considered. To acquire fewer parameters and decrease depth, bottleneck blocks are used to produce a thinner network. They consist of three convolutional layers; a 1×1 layer for down-sampling channel dimension, a 3×3 layer and a 1×1 layer for up-sampling the channel dimension.

## 2.2 Convolutional Layer

Convolution is defined as a mathematical operation that does an integral part of the product of two signals with one of them being flipped. The input of the convolutional layer is divided into equal regions of pixels (neurons), where each region is attached to a neuron of a hidden-layer and the convolved output is assigned. For example, in Figure 2, with 3×3 kernel slides (assuming Stride = 1) all over the input, the output represents the dot-product of a selected 3×3 image-input and flipped kernel. Additionally, it creates a convolved feature map (hidden layer) and so on for the rest of the kernels in which more hidden layers based on learned features are created.

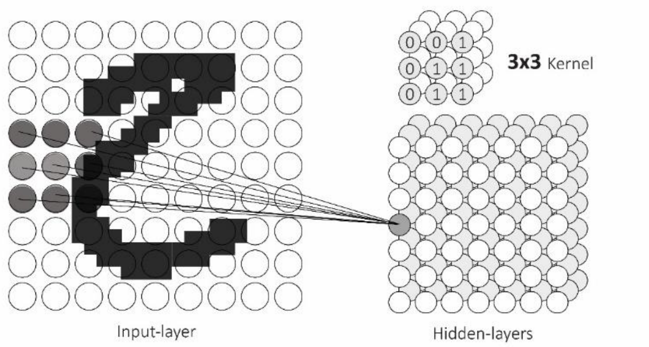



Figure 2. The convolutional layer.

Activation function is defined as a non-linear transformation and it basically decides whether a neuron should be activated or not. The Rectified Linear Unit (ReLU) is widely used in computer vision-related problems where deep networks are used, since it converges faster than the previous activations, such as sigmoid and is cheaper to be computed. As a result, it is leading to encounter a faster training time. However, it has a downside of causing dead neuron, once the neuron is always negative.

According to a study conducted by Xu et al. [23], different types of rectified activation functions in convolutional network are tested and compared in terms of accuracy and error rate. The activations comprise the standard ReLU, Leaky (LReLU), Parametric (PReLU) and Randomised Leaky (RReLU). The aim is to prove the effect of a non-zero slope for negative parts in rectified activation units on improving results. Based on their experimental results, the RReLU can overcome the other activations, but the PReLU is suspected not to function with smaller datasets. In this research, the LReLU activation is used, as shown in Equation 5 (Maas et al. [24]), where, $a_i$denotes the small, positive defined number and denotes a set that is highly based on experiments.

$$f(x_i) = \begin{cases} x_i & x \geq 0 \\ \frac{x_i}{a_i} & x < 0 \end{cases} \quad (5)$$

where: $a_i \epsilon (1, +\infty)$.

Batch normalization layer adjusts and scales the activations. It produces activations with a stable distribution throughout training, by enforcing the values of each layer to represent the same distribution. It solves the problem of internal covariance shift, which represents the amount by which the hidden unit values shift around. It is applied before non-linear layers.

### 2.3 Classification Layer

The classification layer basically consists of three parts. The first part represents the flattening layer that takes the output of the last residual block after applying the activation function and the average pooling. The flattening layer transforms the output into a 1-D vector to be used in the subsequent layer, which represents a fully connected layer. It consists of 1000-feature maps made up by the global average pooling, where each neuron is connected to the entire neurons within the next layer. Finally, the softmax activation function is responsible for predicting the final output (see Equation 6). It squashes the outputs of the layer beforehand along towards the range between zero and one for each neuron and the entire assigned values after applying Softmax must have a summation of one. A normal distribution of the values simplifies scattered predictions. There exist 76 classes (output units) in the thesis's case that represents Arabic characters and digits.

$$\sigma(z)_i = \frac{e^{z_j}}{\sum_{K=1}^{K} e^{K_j}} \quad (6)$$

where *Z* denotes the vector of the inputs related to the output layer and *j* indexes the output units *j*=1, 2, … k.

## 3. Experimental Results

In this section, the results of the proposed approach based on the Arabic handwritten dataset are presented. The accuracy and validation pertaining to the proposed approach are found for 76 classes of many different handwritten characters and digits. The dataset used in this study is also presented. Data analysis and interpretation are presented as well.

### 3.1 Dataset

A new constructed dataset from some datasets used by other researchers is used to evaluate the proposed approach. The new dataset consists of letters and digits from 0 to 9 for covering the shortage that is found through existing ones. Modified letters as Al-Hamza (ء) and tā' marbūṭah (ة) are not included in most datasets. The collected samples are re-categorized to 76 classes, including the entire contextual cases of letters.

The digits' dataset is extracted from the MADBase proposed by El-Sawy et al. [25] and which represents the largest found dataset. The characters are extracted from the AHCD proposed by El-Sawy et al. [26], the DBAHCL approach proposed by Lamghari and Raghay [27], the OIHAC approach proposed by Boufenar et al. [28] and the AIA9k approach proposed by Torki et al. [29].

Images are initially pre-processed and segmented and hence, the dimensions are unified into an image size of 32×32 pixels and the colours are converted into black and white, as shown in Figure 3. Finally, a random subset is labelled and separated into the 76 classes. The total quantity of the dataset reaches 10340 images, which are split into 80% training (8320 images) and 20% validation (2100 images). Table 2 summarizes the characteristics of the datasets used in the construction of the new used dataset.

### 3.2 Data Analysis and Interpretation

The results are evaluated by measuring the standard deviation of the layer responses, where the obtained results reveal the strengths and weaknesses of residual functions among the involved layers. Responses represent each layer's output after providing nonlinear functions, such as the ReLU and Addition (see Equation 7).

$$y_i = h(x_I) + F(x_I, W_I) \tag{7}$$

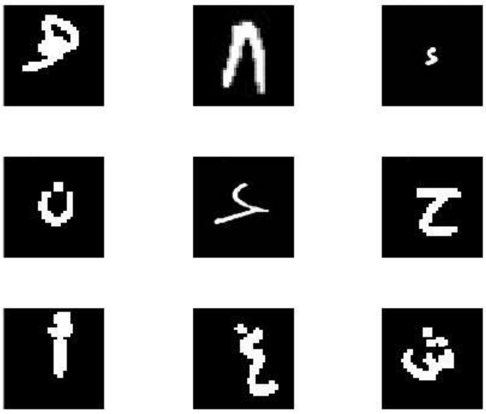


Figure 3. Randomly selected images.

Table 2. Datasets characteristics.

| Dataset | Characteristics |
|---|---|
| MADBase [25] | The number of samples: 70000<br>Classes: 10<br>Dimensions: 32×32 |
| AHCD [26] | The number of samples: 16800<br>Classes: 28<br>Dimensions: 32×32<br>Colour space: Grey |
| DBAHCL [27] | The number of samples: 5400<br>Classes: 54<br>Dimensions: 80×80<br>Colour space: RGB |
| OIHAC [28] | The number of samples: 5600<br>Classes: 28<br>Dimensions: 128×128<br>Colour space: Grey |
| AIA9k [29] | The number of samples: 8737<br>Classes: 28<br>Dimensions: 32×32<br>Colour space: Grey |

However, residual responses are generally closer to zero, particularly for deeper networks where each layer tends to modify the signal loss [4]. Basically, the evaluation and analysis pertaining to the results apply the Cross Entropy (Equation 8) as a loss function that finds the distance between the predicted probability and the real one.

$$H(p, q) = -\sum\nolimits_x p(x) \times \log q(x) \tag{8}$$

where $p(x)$ denotes the desired probability and $q(x)$ denotes the actual probability. Additionally, the performance of the produced method is evaluated based on the Recall and Precision parameters (Sokolova & Lapalme [30].

Precision (Equation 9) represents the average per-class agreement pertaining to the data class labels including those containing a classifier. The values used to describe Recall, shown in Equation 10, describe the average per-class effectiveness of a classifier to identify class labels.

$$Precision = \frac{\sum_i^l \frac{tp_i}{tp_i + fp_i}}{l} \tag{9}$$

$$Recall = \frac{\sum_i^l \frac{tp_i}{tp_i + fn_i}}{l} \tag{10}$$

### 3.3 Experiments

In this research, a 117-layer residual network is constructed with a network width of 15 units. Deep learning networks are deeply requiring big data to model the training datasets. Unfortunately, many applications do not have big datasets, such as OCR. Data augmentation comprises a group of techniques that enhance the size and quality of training datasets to build better deep learning models. Data augmentation can improve the performance of deep learning models and expand limited datasets to take advantage of the capabilities of big datasets [31]. In this paper, data augmentation is done by rotating the training examples horizontally and vertically using different rotation angles resulting in increasing the dataset to about 31,260 images divided into 24960 for training and 6300 for testing. For the subsequent experiments, the constructed network is used where the network is trained on publicly available datasets to prove the efficiency related to this method.

In terms of the training parameters' setting, the network is initially trained for 80 epochs and after that, for 90 epochs in the following experiment and finally, for 200 epochs. An epoch represents the number of passes through the training set before convergence. The learning rate is set proportional to the mini-batch size as it drops after the 60$^{th}$ epoch, where the validation accuracy drops in a few iterations before improving through the subsequent iterations. The learning rate is initially set to 0.1. The learning rate and batch size can implicitly influence the noise that is derived from performing the Stochastic Gradient Decent. The mini-batch size is set to 20, the proportional to the training set size related to each character. Table 3 summarizes the training parameters that are previously described.

Table 3. The tuning of training options.

| Option | Value |
|---|---|
| Initial Learning Rate | 0.1 |
| Max Epochs | 80, 90, 200 |
| Mini Batch Size | 20 |
| Learning Rate Drop Factor<br>(Changed after 60 epochs) | 0.01 |

The experiments are performed on a computer with an Intel core i5, 2.8 GHz processor and 8 GB of RAM. Table 4 shows the trained network results through training. The lowest training error (loss) was 0.025% and the validation accuracy reached 98.9% achieved in the fourth quarter of the experiment after 200 training epochs.

Table 4. Trained network results.

| Epochs | Accuracy | Loss | Validation Error |
|---|---|---|---|
| 50 | 90.1% | 0.23% | 9.9% |
| 100 | 97.8% | 0.14% | 2.2% |
| 150 | 98.7% | 0.05% | 1.3% |
| 200 | 98.91% | 0.025% | 1.09% |

The training process required around 13 hours to terminate the entire determined iterations of 80800 iterations per epoch. The reason behind this process is the complexity of the network and the computations that are needed to train each layer which is considered a deep network with over 100 layers and the expanded number of samples after performing the data augmentation process.

### 3.4 Results

The results are summarized based on validation error, validation accuracy and loss. The experiment was sectioned into 4 phases throughout the 200 epochs. Table 5 shows a validation error of 9.9% for 50 epochs, 2.2% for 100 epochs, 1.3% for 150 epochs and 1.09% for 200 epochs. The lowest training error (loss) was 0.025% and the validation accuracy reached 98.9% achieved in the fourth quarter of the experiment after 200 training epochs. The overall progress of the experiment is shown in Figure 4, that displays the validation accuracy progress through the training process and Figure 5 shows the loss where it converges to 0.025% through the training process.

Table 5. A summary of results.

| Epochs | Accuracy | Loss | Validation Error |
|---|---|---|---|
| 50 | 90.1% | 0.23% | 9.9% |
| 100 | 97.8% | 0.14% | 2.2% |
| 150 | 98.7% | 0.05% | 1.3% |
| 200 | 98.91% | 0.025% | 1.09% |

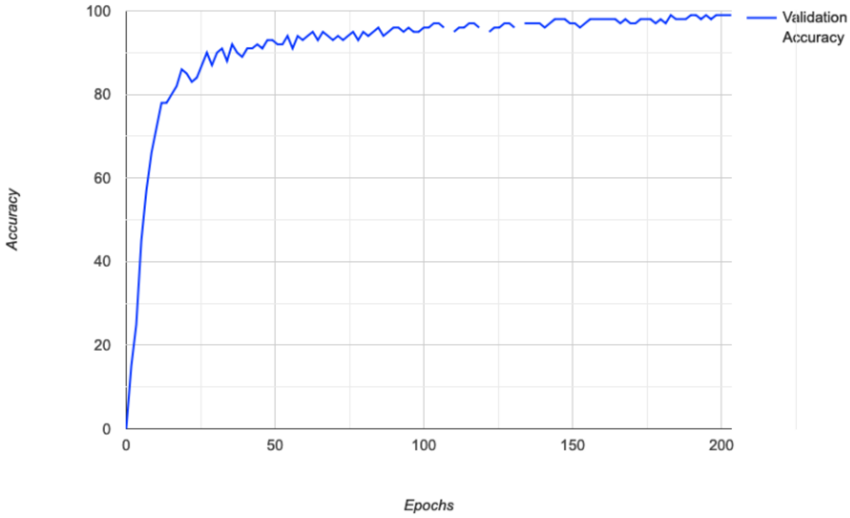



Figure 4. The experiment validation accuracy.

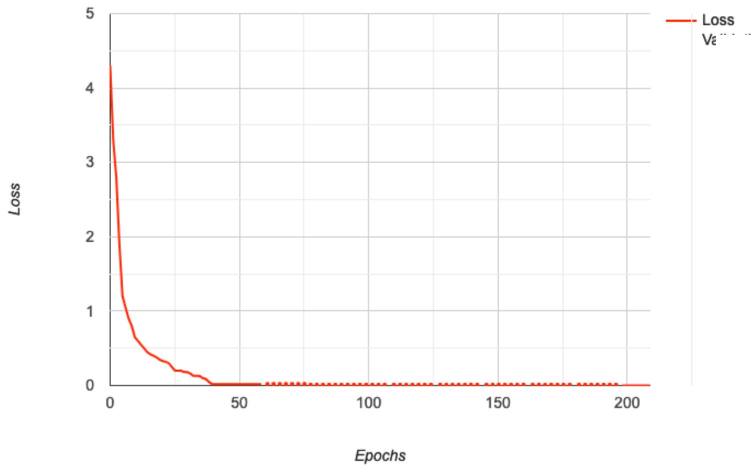



Figure 5. The experiment loss mean.

## 3.5 Comparison with Other Approaches

Experiments are performed on available datasets and the results of the proposed approach are compared with those of previous works using the same datasets (Table 6).

Comparison of the proposed approach with other approaches is shown in Figure 6. The proposed approach obtained an accuracy of 99.55% on the AHCD dataset, whereas Younis [10] obtained an accuracy of 94.8% on the same dataset using deep CNN and batch normalization approaches. On the AIA9k dataset, the proposed approach achieved an accuracy of 99.05%, while Younis [10] obtained an accuracy of 97.6%. Using the MADBase dataset, the proposed approach achieved an accuracy of 99.80%, while other researchers obtained less accuracy; Mudhsh & Almodfer [6] obtained an accuracy of 99.66% and Younis [10] achieved an accuracy of 97.6%. It can be inferred from the results that the proposed approach achieved more accurate results compared to other approaches.

Table 6. Comparisons with other approaches.

| Dataset | Reference | Approach | Epochs | Classification Accuracy |
|---|---|---|---|---|
| AHCD | [10] | Deep CNN | 18 | 94.8% |
| | Proposed Approach | ResNets | 50 | 99.55% |
| AIA9K | [10] | Deep CNN | 18 | 97.6% |
| | Proposed Approach | ResNets | 50 | 99.05% |
| MADBase | [10] | Deep CNN | 18 | 97.6% |
| | [6] | CNN/ VGGnet | N.A. | 99.66% |
| | [32] | Deep CNN | N.A. | 99.30% |
| | Proposed Approach | ResNets | 50 | 99.80% |

## 4. Discussion

It can be inferred form the obtained results that they seem extremely promising as shown previously in terms of the validation accuracy and loss. Precision and recall of each misclassified character are displayed in Table 7. It is seen that characters share several and similar features that are most often being misclassified. For example, the character (ـف) is mistaken with characters (ـم) and (ـه). Accordingly, it is previously mentioned that a character could be mistaken with another character by a dot in cases of a variety of writing styles, which represents one of the major challenges that is related to the language itself. Additionally, it is assumed in this thesis that the reason behind this mistake refers to gaining a small training/testing dataset for characters that are written within the context, where contextual characters only possess 80 training samples and 20 samples for validation.

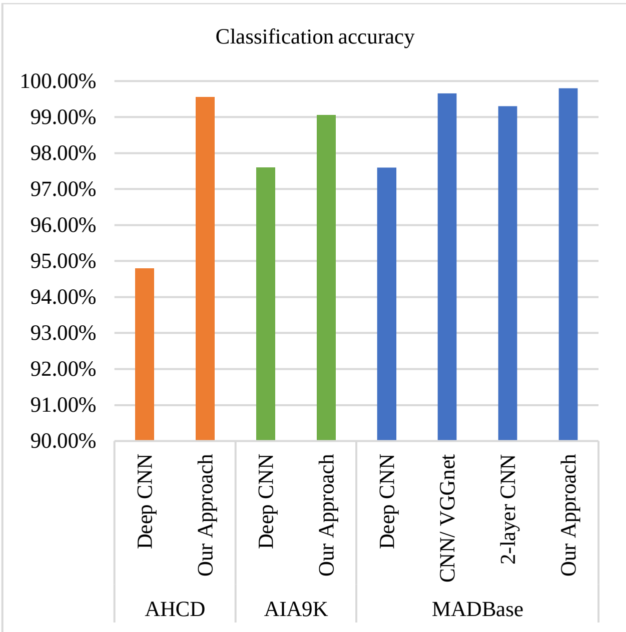



Figure 6. Results' comparison.

On the other hand, the overall recall error is seen to be satisfying. However, two characters are classified twice with wrong labels per character and the rest of the characters. As seen in Table 7, a summary of misclassified characters is shown with the mean average of precision and recall. Also, a summary of the error mean of precision and recall is shown in Figures 7 and 8.

As seen in the comparison with other approaches related to deep learning networks, it is shown that the proposed approach achieved more accurate results. It is previously claimed that the addition of residual blocks that have a special skip connection (or identity mapping) can affect the emerging results positively. Nonetheless, stacking convolutional layers to a definite number performs efficiently while the network is still deepened. The ResNet architecture shows that the network proceeds deeper with more than 150 layers if there is a large training dataset available. Additionally, by using the batch normalisation as a regularisation method for limiting the over-fitting problem and replacing the ReLU activation function with the LReLU function makes several neurons be as active as possible, which we tested on both activations and observed a noticeable improvement in the results.

Table 7. Misclassified labels, precision and recall percentages.

| Character | Precision % | Character | Recall % |
|---|---|---|---|
| حـ | 95.2 | حـ | 95 |
| ـحـ | 95.2 | ـع | 95 |
| ـد | 95.2 | ظ | 95 |
| ث | 97.6 | سـ | 95 |
| ص | 95.2 | ـس | 95 |
| ـص | 94.7 | ـص | 90 |
| ط | 97.6 | ـط | 95 |
| ـعـ | 95.2 | ـفـ | 95 |
| فـ | 90.9 | مـ | 95 |
| ـك | 95.2 | ـم | 95 |
| مـ | 95 | ـن | 95 |
| ـن | 90.5 | ن | 97.5 |
| ـو | 95.2 | هـ | 95 |
| ي | 95 | ـه | 95 |
| ٣ | 97.6 | و | 95 |
| ٧ | 97.6 | ي | 95 |
| ء | 95 | ـي | 95 |
| ٤ | 97 | | |

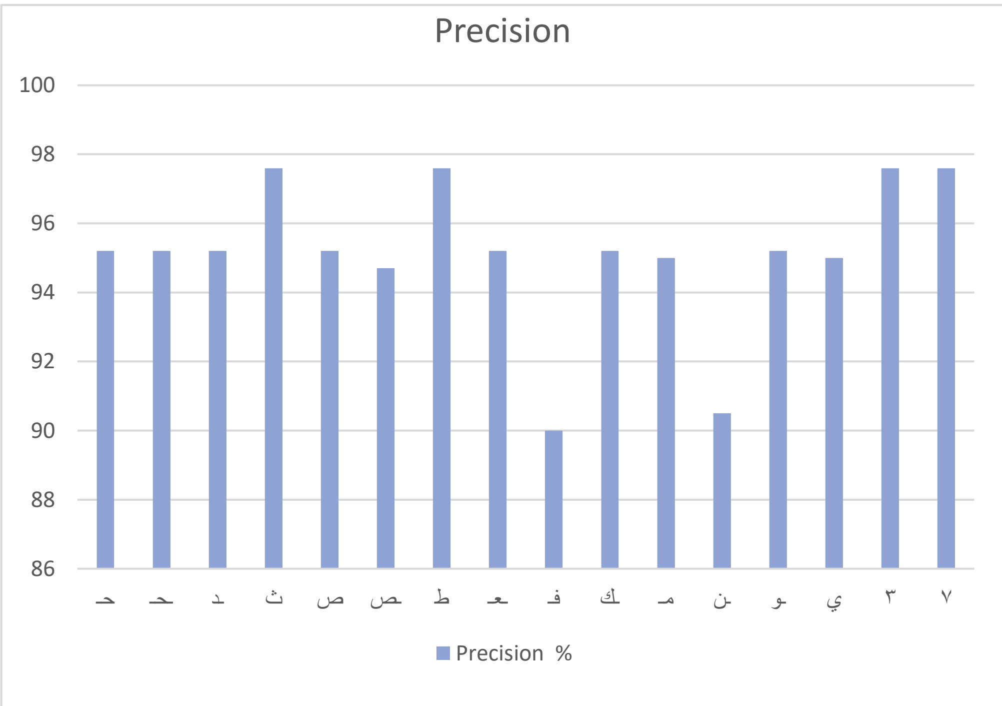



Figure 7. Precision percentage of misclassified characters.

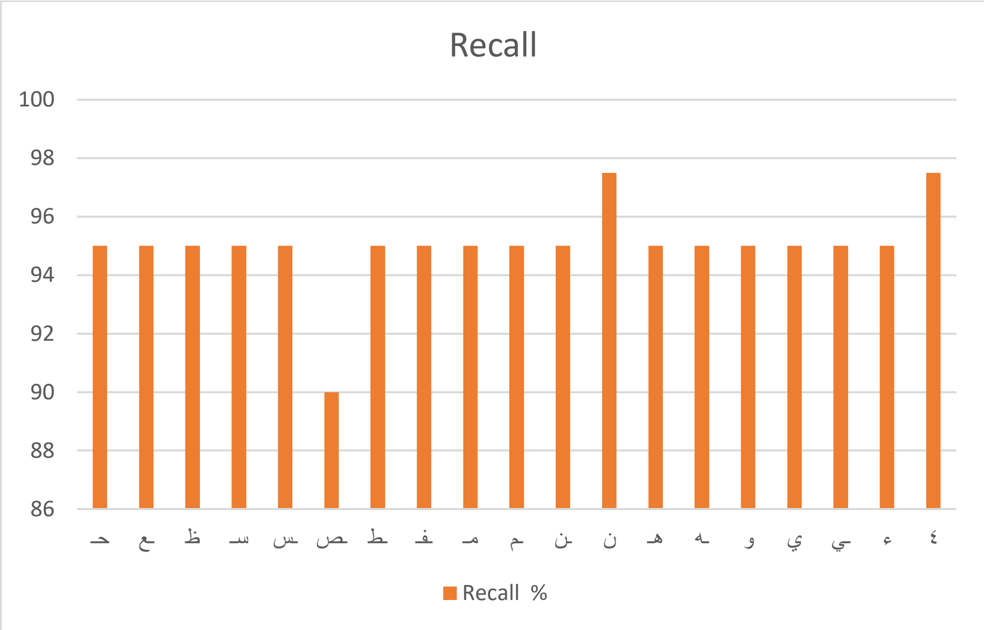



Figure 8. Recall percentage of misclassified characters.

## 5. Conclusions

In this research, a ResNet approach is constructed to recognize isolated Arabic handwritten characters in different contextual cases and digits. The existing architecture is modified as required in terms of several layers, network parameters, regularization techniques and activations being regularly used. The network is trained on a dataset that is collected from previously published datasets in order to obtain an inclusive dataset. The cursive nature of Arabic scripts, the variety of writing styles of each person and the excessive need of enriching Arabic language processing resources all represent robust motivations to start carrying out this study.

Previous researchers discussed different approaches for solving this problem. The focus of this study is on researchers who incubate deep learning methods. Starting from famous convolutional networks, such as the LeCun's LeNet and VGGNets networks, the experiments show that deeper networks could work more efficiently with the chosen input size and problems, such as vanishing gradient and over-fitting problems.

Previous researchers suggested solutions, such as eliminating the use of a dropout layer to deal with the over-fitting problem by using the right activation function, such as the ReLU function and BN for regularization to solve such problems (e.g. the vanishing gradient problem). Moreover, the addition of a shortcut connection into the core building block produces an obvious improvement compared to previous approaches. The main findings and objectives of this study can be summarized as follows:

- In this research, we discussed the problem of recognizing handwritten Arabic characters and eastern Arabic digits. The results of this approach outperformed previous scholars' approaches' results.
- Deeper networks must be handled correctly to avoid problems such as over-fitting. In this research, it was proved that the addition of a shortcut connection could handle deeper networks and improve accuracy. Deeper networks can learn features in different levels of abstraction. However, wider, shallower networks perform good in memorization but not in generalization.
- The experiments with the constructed dataset that included both characters and digits proved that the approach used succeeded for this case as seen in the results' section. Moreover, there are many similar features between eastern Arabic digits and characters, which implies the accuracy of the approach.

Directions for future work include working on a segmentation-free technique that is capable of recognizing the sequence of characters (words) with the addition of long-short term memory network specifications.

## REFERENCES

[1] C. Boufenar and M. Batouche, "Investigation on Deep Learning for Off-line Handwritten Arabic Character Recognition Using Theano Research Platform," Intelligent Systems and Computer Vision (ISCV), pp. 1-6, DOI: 10.1109/ISACV.2017.8054902, Fez, Morocco, 2017.

[2] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based Learning Applied to Document Recognition," Proc. of the IEEE, vol. 86, no. 11, pp. 2278-2324, DOI: 10.1109/5.726791, 1998.

[3] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural networks," Communications of the ACM, vol. 60, no. 6, pp. 84–90, DOI: 10.1145/3065386, 2017.

[4] S. Wu, S. Zhong and Y. Liu, "Deep Residual Learning for Image Steganalysis," Multimedia Tools and Applications, vol. 77, pp. 10437-10453, 2018.

[5] A. Shiquzzaman and A. K. Tushar, "Handwritten Arabic Numeral Recognition Using Deep Learning Neural Networks," Proc. of the IEEE International Conference on Imaging, Vision & Pattern Recognition (icIVPR), pp. 1-4, DOI: 10.1109/ICIVPR.2017.7890866, Dhaka, Bangladesh, 2017.

[6] M. Mudhsh and R. Almodfer, "Arabic Handwritten Alphanumeric Character Recognition Using Very Deep Neural Network," Information, vol. 8, no. 3, pp. 105-118, DOI: 10.3390/info8030105, 2017.

[7] K. Younis and A. N. Alkhateeb, "A New Implementation of Deep Neural Networks for Optical Character Recognition and Face Recognition," Proc. of the International Conference on New Trends in Information Technology (NTIT-2017), pp. 1-6, Amman, Jordan, April 2017.

[8] H. Tomimori, K. T. Chen and T. Baba, "A Convolutional Neural Network with Incremental Learning," Journal of Signal Processing, vol. 21, no. 4, pp. 155-158, 2017.

[9] A. Eladel, R. Ejbali, M. Zaied and C. Ben Amar, "Fast Deep Neural Network Based on Intelligent Dropout and Layer Skipping," Proc. of the International Joint Conference on Neural Networks, [Online], Available: https://doi.org/10.1109/IJCNN.2017.7965947, 2017.

[10] K. S. Younis, "Arabic Handwritten Character Recognition Based on Deep Convolutional Neural Networks," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 3, no. 3, pp. 186-200, 2017.

[11] M. Elleuch, R. Maalej and M. Kherallah, "A New Design Based-SVM of the CNN Classifier Architecture with Dropout for Offline Arabic Handwritten Recognition," Procedia Computer Science, vol. 80, pp. 1712-1723, DOI: 10.1016/j.procs.2016.05.512, 2016.

[12] N. Tagougui and M. Kherallah, "Recognizing Online Arabic Handwritten Characters Using a Deep Architecture," Proc. of the 9th International Conference on Machine Vision (ICMV 2016), vol. 10341, DOI: 10.1117/12.2268419, Nice, France, 2017.

[13] S. Karthik and K. Srikanta Murthy, "Deep Belief Network Based Approach to Recognize Handwritten Kannada Characters Using Histogram of Oriented Gradients and Raw Pixel Features," International Journal of Applied Engineering Research, vol. 11, no. 5, pp. 3553-3557, 2016.

[14] M. E. Mustafa and M. K. Elbashir, "A Deep Learning Approach for Handwritten Arabic Names Recognition," International Journal of Advanced Computer Science and Applications, vol. 11, no. 1, pp. 678-682, DOI: 10.14569/IJACSA.2020.0110183, 2020.

[15] T. M. Ghanim, M. I. Khalil and H. M. Abbas, "Comparative Study on Deep Convolution Neural Networks DCNN-based Offline Arabic Handwriting Recognition," IEEE Access, vol. 8, pp. 95465-95482, 2020.

[16] N. Altwaijry and Isra Al-Turaiki, "Arabic Handwriting Recognition System Using Convolutional Neural Network," Neural Computing and Applications, vol. 33, pp. 2249-2261, 2021.

[17] H. Akouaydi, S. Njah, W. Ouarda, A. Samet, M. Zaied and A. M. Alimi, "Convolutional Neural Networks for Online Arabic Characters Recognition with Beta-elliptic Knowledge Domain," Proc. of the IEEE International Conference on Document Analysis and Recognition Workshops (ICDARW), pp. 41-46, DOI: 10.1109/ICDARW.2019.50114, Sydney, Australia, 2019.

[18] K. He, X. Zhang, S. Ren and J. Sun, "Identity Mappings in Deep Residual Networks," Proc. of the European Conference on Computer Vision (ECCV 2016), Part of the Lecture Notes in Computer Science, vol. 9908, DOI: 10.1007/978-3-319-46493-0_38, Springer, Cham, 2016.

[19] P. Sermanet, S. Chintala and Y. LeCun, "Convolutional Neural Networks Applied to House Numbers Digit Classification," Proc. of the 21st International Conference on Pattern Recognition (ICPR2012), pp. 3288-3291, Tsukuba, Japan, 2012.

[20] M. Längkvist, L. Karlsson and A. Loutfi, "A Review of Unsupervised Feature Learning and Deep Learning for Time-series Modeling," Pattern Recognition Letters, vol. 42, pp. 11-24, 2014.

[21] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778, DOI: 10.1109/ICDARW.2019.50114, Las Vegas, USA, 2016.

[22] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," Proc. of the 32nd International Conference on International Conference on Machine Learning (CML'15), vol. 37, pp. 448–456, 2015.

[23] B. Xu, N. Wang, T. Chen and M. Li, "Empirical Evaluation of Rectified Activations in Convolutional Network," Machine Learning (cs.LG); Computer Vision and Pattern Recognition (cs.CV); Machine Learning (stat.ML), [Online], available: http://arxiv.org/abs/1505.00853, 2015.

[24] A. L. Maas, A. Y. Hannun and A. Y. Ng, "Rectifier Nonlinearities Improve Neural Network Acoustic Models, " Proc. of the 30th International Conference on Machine Learning, JMLR: W&CP, vol. 28, Atlanta, Georgia, USA, 2013.

[25] A. El-Sawy, M. Loey and E. B. Hazem, "Arabic Handwritten Characters Recognition Using Convolutional Neural Network, " WSEAS Transactions on Computer Research, vol. 5, pp.11-19, 2017.

[26] A. El-Sawy, H. EL-Bakry and M. Loey, "CNN for Handwritten Arabic Digits Recognition Based on LeNet-5," Proc. of the International Conference on Advanced Intelligent Systems and Informatics (AISI 2016), Part of the Advances in Intelligent Systems and Computing, vol. 533, DOI: 10.1007/978-3-319-48308-5_54, Springer, Cham, 2017.

[27] N. Lamghari and S. Raghay, "DBAHCL: Database for Arabic Handwritten Characters and Ligatures," International Journal of Multimedia Information Retrieval," vol. 6, pp. 263-269, 2017.

[28] C. Boufenar, M. Batouche and M. Schoenauer, "An Artificial Immune System for Offline Isolated Handwritten Arabic Character Recognition," Evolving Systems, vol. 9, pp. 25-41, 2018.

[29] M. Torki, M. E. Hussein, A. Elsallamy, M. Fayyaz and S. Yaser, "Window-based Descriptors for Arabic Handwritten Alphabet Recognition: A Comparative Study on a Novel Dataset," arXiv:1411.3519v2, [Online], Available: https://arxiv.org/pdf/1411.3519.pdf, 2014.

[30] M. Sokolova and G. Lapalme, "A Systematic Analysis of Performance Measures for Classification Tasks," Information Processing and Management, vol. 45, no. 4, pp. 427-437, 2009.

[31] C. Shorten and T. M. Khoshgoftaar, "A Survey on Image Data Augmentation for Deep Learning," Journal of Big Data, vol. 6, no. 60, DOI: 10.1186/s40537-019-0197-0, 2019.

[32] G. Latif, J. Alghazo, L. Alzubaidi, M. M. Naseer and Y. Alghazo, "Deep Convolutional Neural Network for Recognition of Unified Multi-language Handwritten Numerals," Proc. of the 2nd IEEE International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR), pp. 90-95, DOI: 10.1109/ASAR.2018.8480289, London, UK, 2018.

**ملخص البحث:**

تقترح هذه الدراسة استخدام الشّبكات العصبيّة المتبقية (ResNets) للتعرف على الأحرف العربية المكتوبة بخط اليد، بما في ذلك الأرقام العربية. وتمثل شبكات ResNets نهجاً من التعلّم العميق الذي أظهر فعالية في العديد من التطبيقات أكثر من مناهج التعلم الآلي التقليدية. يتكون النّهج المقترح من ثلاث مراحل رئيسية هي: مرحلة ما قبل المعالجة، وتدريب الشبكة على مجموعة التدريب، واختبار الشبكة المدربة على مجموعات البيانات. تم إجراء تقييم للنّهج المقترح على ثلاث مجموعات بيانات متاحة هي: MADBase و AIA9k وAHCD. وقد حقق النّهج المقترح دقة بلغت 99.8٪ و 99.05٪ و 99.55٪ على مجموعات البيانات المذكورة على التوالي. كما حققت دقّة تحقق بلغت 98.9٪ على مجموعة البيانات المُنشأة بناءً على مجموعات البيانات الثلاث.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 07, No. 02, June 2021.

# FAULT TOLERANCE USING SELF-HEALING SLA AND LOAD BALANCED DYNAMIC RESOURCE PROVISIONING IN CLOUD COMPUTING

Mayank Sohani and S. C. Jain

*(Received: 13-May-2021, Revised: 22-May-2021, Accepted: 24-May-2021)*


## *ABSTRACT*

*Over the internet, application efficiency management has recently emerged as an essential service cloud computing. The Cloud Service Provider (CSP) gives various cloud services based on pay per use, which requires efficient monitoring and measuring of services delivered for management of Quality of Service (QoS) through the Internet of Things (IoT) and therefore needs to fulfil the Service Level Agreements (SLAs). However, avoiding SLA violations and ensuring a user's dynamic demands as per QoS fulfilment are challenging in cloud computing while delivering dedicated cloud services. Cloud environment intricacy, heterogeneity and dynamism are expanding quickly, making cloud frameworks unmanageable and unreliable. Cloud systems need self-management of services to overcome these issues. Therefore, there is a need to develop a resource-provisioning scheme that automatically fulfils cloud user's QoS requirements, thus helping the CSP accomplish the SLAs and avoid SLA violations. This paper presents a prediction-based resource management technique called Predictive Cloud Computing Systems (PCCSs). Focus is on the self-healing-based prediction that handles unexpected failures and self-configuration-based prediction of resources for applications. The Predictive Cloud Computing System (PCCS) performance is evaluated in the cloud simulator. The simulation results revealed that Predictive Cloud Computing Systems (PCCSs) achieve better results than existing techniques, in terms of execution time, cost-effectiveness, resource conflict and SLA breach while delivering reliable services.*



## *KEYWORDS*

*Quality of service, Cloud-service provider, Service-level agreement, Service-level objective, Predictive cloud computing system.*


## 1. INTRODUCTION

Cloud computing focuses on QoS parameters, such as throughput, response time, availability, capability, service cost and reliability, among others. The QoS parameters play a vital role in the ranking of service providers. QoS parameters are continuously monitored and controlled by service providers to avoid SLA breaches. According to the study and analysis, it is reported that the Virtual Machine (VM) requires different durations of boot time before it is ready to operate [1] [2] [3]. The VM needs 5 to 15 minutes to get started; therefore, during this time, system resources would not be available and the requests could not be served due to lack of resources. This leads to an infringement of SLA and due to this, penalties on cloud providers are imposed. Our objective is to design a solution for provisioning and predicting the need for a VM in advance. Making it available only on time could maintain the level of availability and prevent violations of the SLA [4]. This research will discuss cloud computing advantages, such as up-front costs, lower infrastructure maintenance and ease of resource scaling for the users. Cloud computing has various benefits and many issues of energy consumption, resource utilization, VM migration and service-level agreement (SLA) violations [5]-[6]. In this paper, we are using a threshold-based Virtual Machine Consolidation (VMC) strategy. Many issues of the resources need to be addressed. Therefore, VM consolidation (VMC) is the best way to solve them.

In cloud computing, unpredictable situations are handled by an intelligent autonomic system that keeps the system stable based on human guidance and easily adapted to new environmental conditions, such as hardware, software failures, …etc. This system can quickly handle the heterogeneity, availability, reliability and dynamism problems. The system works through monitoring, analyzing, planning and execution phases in a controlled way in order to achieve the application execution goal within the deadline by fulfilling the user's defined QoS parameters with minimum complexity. Virtual machine

M. Sohani and S. C. Jain are with the Department of Computer Science and Engineering, Rajasthan Technical University, Kota-324010, India. Emails: sohanimayank@gmail.com and scjain1@gmail.com

optimum utilization is highly desirable to maintain the required SLA and it is achieved by virtual machine dynamic consolidation. The live VM migration is used for VM reallocation as per current resource workload demands of users and reduces energy consumption [7] [8] [9] [10] [11]. However, virtual machine migration tends to increase application execution latency and infrastructure energy overheads. Many factors are considered in migration cost, such as the number of virtual machines considered for migration, network bandwidth viewed for migration, memory content update rate of the virtual machine, source and destination servers' workload at the time of migration [12]-[13]. During migration cost investigation quantitatively, the power consumption and time of migration linearly increase as network bandwidth and size of VM increase, respectively, whereas migration time decreases if the increase in bandwidth and increases if the VM memory size increases [14]-[15].

Various research literature focuses only on the current resource requirements of the destination host. The future utilization has not been discussed more at the time of the VM allocation stage. That will generate needless VM migrations that would result in more energy consumption and increase SLA violations in the data center [16]. This paper proposes a new prediction-based method for different resource utilizations; i.e., CPU, memory and network. The work focused on the memory utilization of these resources on the hosts at the time of VM placement. Our proposed method is a prediction model based on feedforward neural networks with backpropagation for linear regression-based prediction models. Our detection technique is responsible for current and future resource utilization on the hosts before placing VMs.

As per QoS requirements, a predictive cloud computing system provides self-management of resources that fulfils the following properties of self-management:

- This paper presents a detailed analysis of selected resource provisioning techniques that work for QoS requirements, VM migration strategies, load balancing techniques and SLA violation monitoring schemes.
- It proposes and implements an algorithm for predicting the workload in cloud computing systems.
- The proposed algorithm improves self-healing in a predictive cloud computing system as a capability of the system to identify, analyze and recover from unfortunate faults automatically.
- It proposes and implements self-configuration in a predictive cloud-computing system, which is an indicator of the capability of the system to adapt to the changes in the cloud environment.
- It proposes and implements a new VM migration and load balancing scheme for the cloud-computing system.

In our earlier work, QoS-based Predictive Priority-based Dynamic Resource Provisioning Scheme [17] is proposed. The Predictive Priority-based Dynamic Resource Provisioning Scheme is a novel approach for predicting priority-based scheduling schemes. This explores a new approach that is an efficient emergency priority-aware algorithm. In this scheme, we consider the emergency cloud request and priority is given to load that emergency cloud requests for execution. This will ensure the load request availability and longevity of more sophisticated requests in heterogeneous cloud computing environments without SLA violation monitoring [18]. To realize this, QoS-aware autonomic resource management of cloud services needs to be considered as a crucial aspect that reflects the cloud management complexities. To design a resource management approach which can work as a QoS-based autonomic approach, Predictive Priority-based Dynamic Resource Provisioning Scheme has been further extended by proposing Predictive Cloud Computing System (PCCS). In this research work, a resource management approach which can work as a QoS-based autonomic approach has been proposed which offers fault tolerance using self-healing SLA and load balanced dynamic resource provisioning in cloud computing, to handle sudden failures and provide cloud resource maximum utilization by self-optimization.

The motivation of this paper is to design an intelligent cloud-based and QoS-aware autonomic resource management approach called Fault Tolerance Using Self-healing and Load Balanced Dynamic Resource Provisioning in Cloud Computing. This offers handling of sudden failures of resources through self-healing, resource self-configuration for applications and maximum resource utilization through self-optimization features. The proposed scheme works to minimize SLA violation rate, execution cost, execution time and resource contention and maximize energy efficiency and resource utilization. The PCCS performance is tested with a CloudSim simulation environment using PlanetLab workload traces.

PCCS increases service availability and reliability and improves satisfaction of cloud users. The rest of the paper is organized as follows: Section 2 describes the related work, while the proposed model is presented in Section 3. Section 4 presents the simulation setup, results and discussion. Section 5 presents conclusions and future scope.

## 2. RELATED WORK

Many studies have investigated the SLA management systems in cloud computing, but SLA enforcement is covered only by a few of them. Without considering enough cloud requirements, other environments, such as grid computing and service-oriented architecture, applied the SLA models into cloud computing as per most related works. In the self-healing system, the central part consists of system monitoring and reacting procedures [19]. The Federated Cloud Trust Management Framework (FCTMF) model resolves trust issues. It evaluates trust on the basis of SLA parameters and by customer and CSP feedback [20]. SH-SLA models enforce the SLA monitoring and reacting procedures based on SLA violations in cloud computing. Each SLA is connected with its related SLAs in different layers of the SH-SLA model of cloud computing, so that all corresponding SLAs can notify their status to each SLA. So, cloud service providers can prevent SLA violations before sensing by the end-users without addressing cost and energy consumption QoS parameters [21]. RADAR technique performs autonomic-management properties for self-healing and self-configuration handled during unexpected failures of service and resource configurations, respectively, with minimum human intervention and gives better results for QoS parameters along with managing hardware, software or network faults, but the study unable to address the self-protecting property [22]. Existing approaches consider a host overloaded detection based on threshold-based host CPU utilization and consider available bandwidth equal to base bandwidth, thus leading to performance degradation. The overloaded host VM migration or reallocation towards another under loaded host machine is not addressed in this study [23]. Previous proposed work assumptions are not based on energy consumption and violations of SLA considering network traffic. Energy consumption can be minimized by existing methods considering the size and current utilization of VM, but network traffic can also affect SLA violations [24].

In the cloud environment, this will provide capable monitoring that would be able to share resources in Clouds. In [25], the authors offer the solution cloud federalism, where the different cloud vendors the cloud services in an integrated manner. The Cloud Burst is the best example of cloud federalism. In [26], the authors' discussion is about resource management's performance with the help of live migration. This feature is added in the cloud system that has to provide excellent services into the cloud environment of active fault tolerance by flawless Virtual Machine movement. The consumer is not being aware of any change in a virtualized environment from wavering hardware to unwavering hardware. In these models, virtual technologies have provided resource consolidation with minimum energy consumption and are unable to address the issue of self-management [27]-[28]. Resource over-provisioning can be solved by VM placement as per the VM resource requirements independently based on their requests. Placing more VMs on the same PM by sharing hardware resources exceeds its physical capacity [29]. Unfortunately, over-commitment affects the application performance with QoS violations and SLA penalties by congesting limited PM resources [30]. In UP-VMC, resource requirements for current and future utilization consolidate the VMs with the minimum quantity of active PMs. It uses regression-based prediction for future and current resource utilization, enhancing the QoS and minimizing the number of VM migrations, but application scalability and network resource utilization factors are not addressed in this study [31].

In cloud computing, the overall response time of the system is reduced by load balancing and this policy of workload distribution fulfills the QoS requirements along with efficient cloud resource utilization. Several techniques were proposed; however, VM migration and fault tolerance issues are not still fully addressed [32]. An ideal framework PRMF can identify current workload and future workload prediction for provisioning/deprovisioning cloud resources as per the demand of application users. This framework identifies given workload patterns with key evaluation metrics using statistical techniques. It applies best-fit algorithms from algorithms using predictive methods to provision/de-provision VM instances, but is unable to address issues, such as cost, makespan time and energy consumption [33]. Resource provisioning techniques are working based on predetermined considerations, are reactive and are provided with leading CSPs. Under-or over-provisioning of resources is done in reactive approaches that have time-lag in resource demand and provisioning. The study proposed a predictive technique for cloud resource management to overcome these limitations [34]. In cloud computing, search optimization methods are introduced by many studies, but there is still some scope to get enhanced search for optimal solutions. To achieve this solution, many functions need to be involved; i.e., execution time, power consumption, performance, QoS and SLA violation rate [35]. However, in some earlier works, the maximum three objective functions are taken into consideration to get the optimal solution in cloud computing. The ESCORT framework addresses these issues to optimize execution cost, energy consumption and SLA violation rate [36].

A secure resource provisioning model with SLA integration is proposed to achieve many benefits for cloud users' and cloud service providers' points of view. This secure provisioning model is used by cloud service providers for the security parameters' fulfilment purpose without considering other major QoS parameters, such as execution time, cost, throughput, energy consumption [37], …etc. In cloud computing applications, workload changes as per time and to fulfil such workload resource requirements, cloud service providers dynamically allocate the resources. Dynamic resource provisioning aims to improve resource utilization and reduce resource usage costs for cloud users [38]. To achieve profit-aware resource provisioning, the cloud service provider must provide less renting cost with proper resource utilization to meet the QoS requirements. The dynamic resource provisioning technique works as an effective technique for utilization of resources without considering energy consumption and SLA violations. The goal is to minimize the resource rental cost and maximize resource utilization for profit earning [39]. The CHOPPER framework works based on self-protection, self-healing, self-optimization and self-configuration using three phases of self-management; i.e., Monitor, Analyze and Plan & Execute to address different QoS parameters, but it is unable to calculate the workload resource demand in advance [40]. The increase of cloud users with peak time demands makes the risk of resource faults during interactions with the cloud infrastructures match the execution deadline. That can lead to resource contentions and damage the reputation of cloud service providers due to non-consideration of cost and energy consumption in the study [41]. The authors propose an MASA framework that works based on a healing agent and a consistency manager agent to handle the runtime issues of resource provisioning and SLA violations, but it is unable to manage adaptive fault tolerance scheme for cloud security solution [42].

## 3. PROPOSED MODEL

SLA is the most important part between the cloud service provider and the customer. SLA is a mutual agreement between the cloud service provider and the customer. This Service Level Agreement (SLA) is the official negotiation document at the service level and shall contain performance parameters along with the minimum level of service quality. Our proposed SLA is including an automated cloud healing process based on the above description and prediction. In the proposed method, each service has its function of automatic healing and reaction. This SLA-based prediction will work on the threshold value and related SLAs on the cloud user service. This threshold value helps prevent breaches of the SLA and the specific QoS threshold. If the QoS value is higher than the threshold value, the state of violation prevention shall be shown as active and autonomous healing gets activated.

The proposed prediction-based model optimizes cloud computing energy-efficient resources automatically and considers essential aspects, such as configuration, prediction-based recovery, optimization and protection and automatic QoS-aware resource management. Our most essential contributions offer prediction-based intuitive design of cloud applications and resources by installing missed or old H_Components. Prediction-based automatic healing is provided by handling sudden failures, automatic protection against security attacks and automatic optimization as the resources are being used optimally.

### 3.1 System Architecture

The system behavior and its entire structure are represented ultimately with the help of system architecture only. That can define the system's architectural overview of the whole system. The main aim of the proposed Predictive Cloud Computing System is to predict the future workload and ensure resource provisioning in advance with the best suitable pair of resources to fulfil QoS requirements and avoid any SLA violations occurring due to resource provisioning. The proposed model ensures resource provisioning with less power consumption under low execution cost with the best reliable resource pairs for allocation. Figure 1 represents the predictive cloud computing system's architecture concept map.

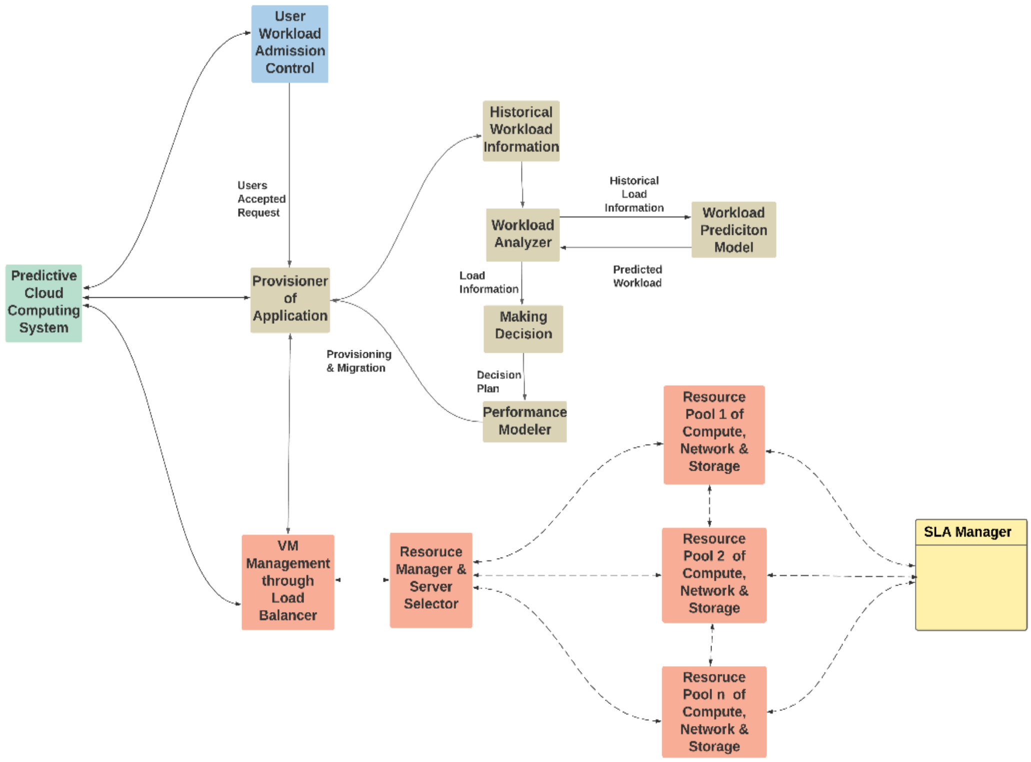



Figure 1. Predictive cloud computing system's architecture concept map.

The workload traces of PlanetLab are taken as an input dataset. The load analyzer performs analysis on workload data based on time series and converts it from unstructured data into structured data as per real-time workload traces received. The workload predictor predicts the future workload and based on this prediction, cloud providers perform arrangement and provisioning of the resources as per QoS requirements. The load analyzer takes care of the current resource utilization of all resource cloud nodes available in the system.

The future workload is predicted based on current workload traces of Planet Lab and prepared structured data based on time series. This predicted workload is used to maintain the SLA commitment towards the cloud user service quality and availability. After predicting the future resource requirements, we can ensure the availability of optimal VM resources and provision them under effective load balancing techniques. The workflow diagram for a Predictive Cloud Computing System is represented in Figure 2. In PCCS, the user submits the request for services based on service types and their properties and negotiation occurs between the user and the CSP. SLA is signed between users and CSP as per QoS requirements and SLA terms. Now, CSP arranges the specific type of resources and sub-resources as per user QoS requirements and provisions these resources for the services used by cloud users. Suppose that required resources are not available in the resource pool. In that case, either renegotiation occurs based on available resources in the resource pool or CSP finds new resources. If resources are available, then the resource configuration is performed using a workflow template. The monitoring unit monitors the entire execution process for user-submitted workload and the workload analyzer prepares a historical workload database. The proposed PCCS applies a predictive cloud computing model on a historical workload database and prepares predicted resources in advance to provide them shortly without violation of the SLA. This PCCS prediction scheme ensures that required resources are ready to be used in advance to save the extra time of provisioning users' workload requests. Application workload is executed using PCCS-provisioned resources. If any demand of current workload is remaining for execution and the same notified by the monitoring unit, then the same is repeated to execute the application workload. This entire process follows four phases; i.e., monitoring, analysis, planning and execution concerning time *t* and updating estimates and actual resource consumption and workload status. Resource configuration upgradation or reconfiguration is performed based on monitoring and analyzing phase inputs for QoS, fault tolerance and SLA fulfilment. VM migration and load balancing are automatically performed based on VM threshold values as per input given by monitoring, analysis, planning and execution phases to the VM load balancer. The entire process gets stoped after the execution of the user-submitted application workload.

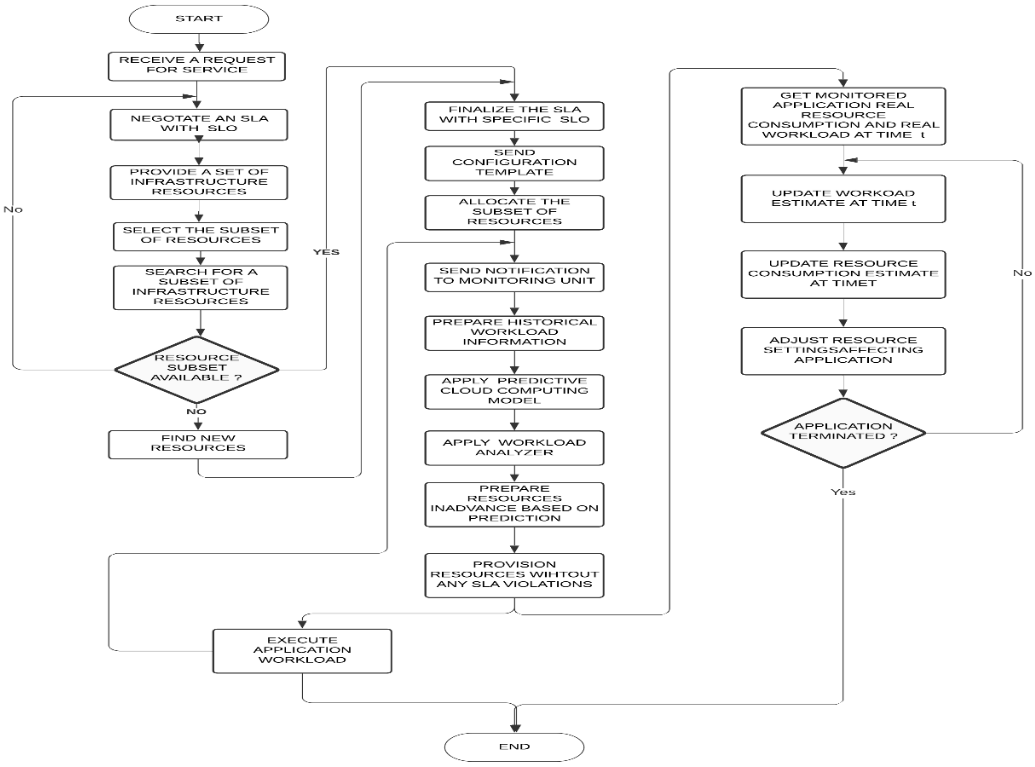



Figure 2. Predictive cloud computing system's workflow.

## 3.2 Workload Analyzer and Predictor

The workload analyzer plays an essential role in the framework to analyze the task load that can only be done after the establishment of cloud infrastructure. The analyzer arranges the unstructured data into structured data based on time series. As unstructured data is no more helpful to get the desired future workload forecasting, the information that can help future workload forecasting is sequential structured data.

Workload predictor is used to take this sequential structured data provided by the workload analyzer as an input to forecast the future workload. Here, backpropagation-based prediction methodology is used, where several input nodes are used to calculate the output using the process of training the workload dataset by the supervised learning methodology. When the task load data formatting is done, it can be used as structured data for quick forecasting purposes, which gives an accurate-manner prediction. This process provides the advantages of automatic learning and reduces the time to fit the analyzer data for every prediction purpose.

### 3.2.1 Host Server Selector and Manager

All the host machine details are listed and managed by the server. As per the workload request, the best possible resources are ranked based on their configuration and performance metrics. These cloud resources are then provisioned based on their performance ranks. The best VM is provisioned for new user workload requests from the load balancer.

### 3.2.2 Forecasting of Predicted Workload

Workload prediction is made on the historical workload data history using the backpropagation algorithm. This algorithm works internally and is used to predict the workload of the near future as a tool.

### 3.2.3 Host State Management

As per the predefined SLA requirements, the host state management performs the cloud resource management as per the resultant prediction without compromising the SLA violations. The host resources are continuously monitored by the host-state manager for their health parameters; i.e., time to start, time to stop, total uptime and total downtime for every physical server.

### 3.2.4 Resource Requirement Forecasting Based on Prediction

In this scheme for forecasting, we are using the backpropagation algorithm, where for the forecast, the future demand of resources is based on the past downtime history. The resource forecasting is carried out using the following equation:

$$\mathrm{RE(t)} = \alpha * \mathrm{RE(t-1)} + (1-\alpha) * \mathrm{RO(t)} \qquad 0 \leq \alpha \leq 1, \tag{1}$$

Here, *RE(t)* expresses the estimated resource and *RO(t)* represents the observed resource load during the time *t*. $\alpha$ is portrayed as a constant that emulates the trade-off between constancy and communion. The proposed model used it to forecast CPU utilization load, predicting the future load calculated in every minute and forecasting immediately.

If the observed load R*O(t)* prediction is in sequential order; i.e., 30, 40, 50 and 60, then prediction for the next term would be more accurate as 70 and then, the algorithm works correctly. Intermediate load values are not forecast by the algorithm. To decrease and increase both order negative value depiction purposes, the formula below for -10 is used in place of the above formula.

$$\mathrm{RE(t)} = -|\alpha| * \mathrm{RE(t-1)} + (1+|\alpha|) * \mathrm{RO(t)} \tag{2}$$

The Depiction for decreasing and increasing order creates confusion to select the exact value between these two. To predict more accurate future workload, the formula is modified as shown below:

$$\mathrm{RE(t)} = \mathrm{m} * \mathrm{RA(t-1)} \tag{3}$$

where m is the multiplier and its value is calculated as:

$$\mathrm{m} = \frac{\mathrm{RA(t-1)}}{\mathrm{RA(t-2)}} \tag{4}$$

Here, the resource-estimated load is calculated using actual resource load concerning time and a multiplier value; i.e., m. This forecasting of workload based on prediction is very near to the relative value of required resources by the cloud users in the near future. However, this prediction model needs resources to predict the forecasting of required resources. To overcome the resource provisioning wastage issues, this model gives perfect VM allocation requirements using backpropagation learning.

The above-defined model provides self-healing for sudden failures, self-protection against security attacks and self-optimization as the resources are being used optimally.

(i) Our proposed model will have no human intervention requirements and will enhance the users' satisfaction level. In this prediction-based model, SLA would efficiently control cloud users' QoS needs and improve the load balancing of the provisioned cloud resource utilization (CPU and memory). Our proposed model optimizes execution cost, time and energy efficiency.

(ii) This model proposed the phases based on prediction-based properties and prediction is based on the regression model. In the execution time of loads, prediction-based performance (QoS value) continuously analyzes the plans and action to handle that message and executes the procedure to maintain efficiency.

(iii) We have classified the load into different categories based on deadline emergency. That will help investigate the impact of various workloads on different QoS parameters. The execution of workloads would enhance the availability of cloud-based services and secure energy-efficiency reliability.

## 3.3 Metrics Based on QoS

The QoS parameters; i.e., waiting time, execution time, energy consumption and execution cost are calculated for user-submitted workload as per the cloud environment consideration.

$$WETi = \sum_{i=1}^{n} \left( \frac{WFTi - WESTi}{n} \right) \tag{5}$$

where $WETi$= workload execution time, $WFTi$= workload finish time, $WESTi$= workload execution start time and $n$= number of workload.

$$WWTi = \sum_{i=1}^{n}\left(\frac{WESTi - WSTi}{n}\right) \tag{6}$$

where $WWTi$= workload waiting time, $WESTi$= workload execution start time, $WSTi$= workload submission time and n= number of workload.

$$WCTi = WETi + WWTi \tag{7}$$

where $WCTi$= workload completion time.

$$EC = ECdc + ECm + ECse + ECe \tag{8}$$

where $EC$= energy consumption, $ECdc$= energy consumption of data center, $ECm$= storage-device energy consumption, $ECse$= switching-equipment energy consumption and $ECe$= extra energy consumption.

$$AC = RC + PC \tag{9}$$

$$RC = WETi \times Price \tag{10}$$

$$PC = \sum_{i=1}^{c}(PCi) \tag{11}$$

where $AC$= average cost, $RC$= resource cost, $PC$=penalty cost and $c \in PC$ = penalty cost set.

$$RUi = \sum_{i=1}^{n}\left(\frac{Resource\ Actual\ Time\ Spent\ to\ Execute\ Workload}{Resource\ Total\ Uptime}\right) \tag{12}$$

where $RU$= resource utilization.

$$FDR = \frac{Number\ of\ Faults\ Detected}{Total\ Number\ of\ Faults} \tag{13}$$

where $FDR$= fault detection rate.

### 3.4 Fault Tolerance Using Self-healing SLA and Load Balancing

In this paper, we are working on a prediction-based self-healing that is a part of proactive fault tolerance (FT) in high-performance computing that prevents computing node failures from affecting running parallel applications, so that nodes would be in the failure process [43]. In this research, our main objective is to work on a prediction-based fault tolerance scheme in cloud computing. Our cloud model approach will work based on the above-described cloud computing and prediction base technique and improve the quality of services.

SLA is an essential document of mutual agreement between the cloud service provider and the customer. In this SLA (Service Level Agreement), the official negotiation document at the service level, the QoS and its service costs shall be agreed upon and shall contain performance parameters and a minimum level of service quality. Our SLA is a proposed SLA, including an automated cloud healing process based on the above description and prediction. In the proposed method, each service has its function of automatic healing and reaction. This SLA-based prediction will work on the threshold value and related SLAs on the cloud service for users. This threshold value helps prevent breaches of the SLA and the specific QoS threshold. The QoS threshold value is compared to the SLO value recorded in the SLA contents. If the QoS value is higher than the threshold value, the state of violation prevention shall be shown as active and autonomous healing. In this paper, prediction-based self-optimization of cloud computing energy-efficient resources, proposing QoS-aware autonomic resource management, considers other essential aspects, such as self-configuration, prediction-based self-healing, self-optimization and self-protection as proposed in algorithms 1, 2 and 3.

The significant contributions of this paper offer prediction-based self-configuration. Algorithm 1 describes the makespan as per MIPS of VM using machHigh and machLow. The task finish time is calculated and is considered as makespan time. This calculated makespan time is then added to predicted task execution time of a VM. If [P_comTimejj< makespan & P_comTimej< makespan] both are true for makespan time, then swap respective parameters; i.e., makespan, task and machine, get executed. After complete execution, makespan time is returned as a value. Our proposed model works for cloud applications and resources on a prediction base that would be installed if the component or cloud application is missing.

**Algorithm 1: Prediction of Workload**

***For all*** $M_i \in machHigh$ ***do***
    ***For all*** $M_j \in machLow$ ***do***
        ***For*** $T_k \in M_i$ ***do***
            $P\_comTime^{ij} \leftarrow computeFinishTime\ (M_j)$
            $P\_comTime^{ij} \leftarrow P\_comTime^{ij} + PredictiveExecutionTime\ of\ T_k\ on\ M_j$
            $P\_comTime^{j} \leftarrow makespan\_PredictiveExecutionTime\ of\ T_k\ on\ M_i$
                ***If*** $P\_comTime^{ij} < makespan$ ***then***
                    ***If*** $P\_comTime^{j} < makespan$ ***then***
                        $Makespan \leftarrow P\_comTime^{ij}$
                        $task \leftarrow T_k$
                        $Machine \leftarrow M_j$
                    ***End if***
                ***End if***
        ***End for***
    ***End for***
***End for***
*return makespan*

In this prediction-based SLA, we aim to predict based healing; therefore, our scheme is a prediction-based algorithm. SLA is a document that is a mutual agreement between client and provider. Our proposed SLA relies on prediction based on load balancing of resources between client and cloud provider. In this framework for load balancing, we create a priority queue of loads and, based on the prediction priority queue, add the load into the priority queue according to a criterion. This criterion is based on threshold values. This load priority queue manages the resources according to predicted execution time and energy consumption and maintains the priority queue; therefore, deadlock is solved according to prediction-based priority queue. In this scheme, we propose prediction-based self-healing as we have all prediction-based priority queues for loads and the resources are allocated according to priority. As that proposed scheme provides the resources based on prediction-based requirements, our proposed system provides the prediction-based resources such as hardware, CPU and memory that our prediction-based SLA can quickly add.

Prediction-based self-optimization, self-healing and auto-configuration are monitored by the monitoring unit and resource performance management is executed using self-management properties as per Algorithm 2. All processing nodes' performance is monitored through QoS agent. The load priority queue considered for workload set (Wpq= {Wp1, Wp2…., Wpm}) is submitted to the load priority queue. The workloads are executed as per QoS and resource availability needs. After provisioning, QoS parameters (execution time, cost and energy consumption) were calculated for every workload using QoS metric equations 5 to 13. Alert is generated if any condition fails [([PET ≤ Dt && PC ≤ BE ] = = 'TRUE') or ([PEC≤PTH] = = 'TRUE')]. Further, in self-healing, the system checks the status of all the components and if any faults are found, it will replace the required components. This entire process maintains log information for current device status and updates resource utilization information. If usage of resources is more than the threshold value [(CurrentStatus ['CPU' || 'MEMORY'] > Value of THRESHOLD)], then alert is generated. All the software versions' status is checked for hardware components in the system. If [(Component version status = OLD || Not-VALID)] is true for OLD or Not-VALID, then generate alters and install the new replacing the old version. The [H_Component_Name and H_Compoenent_Id] is updated based on log information.

Our resources represent nodes that have a state as activate and deactivate. Within this prediction-based SLA, healing uses a hybrid tool in a diagnostic approach. This hybrid tool is used for diagonal purposes and combines analytical methods that cooperate with a common goal. We apply VMC (VM consolidation) based on current and future VM migration in this proposed model. We are using a prediction regression-based model. Algorithm 3 performs the load balancing and VM migration operation. Workload is assigned to a VM as per VM allocation policy and CPU utilization is calculated

***Algorithm 2***

\# **Phase One**: ***Prediction based Self-Optimization***<br>
*Begin*<br>
*Load Priority Queue:* $Wp_q = \{Wp_1, Wp_2, \ldots\ldots, Wp_m\}$<br>
*Add Loads into Priority Queue:* $Wp_a = \{Wp_1, Wp_2, \ldots\ldots, Wp_o\}$ *where* $p_o \le p_m$<br>
*Allocate resources to task loads based on Quality of Services parameters*<br>
*Loop until all Predict base queue loads* ($Wp_a$), *where Predict average cost* ($P_C$) , *Predict energy consumption* ( $P_{EC}$) *and Predict execution time* ($P_{ET}$) *for execution*<br>
*If* ([$P_{ET} \le D_t$ && $P_C \le B_E$] = = 'TRUE') *then*<br>
*If* ([$P_{EC} \le P_{TH}$] = = 'TRUE') *then*<br>
*Schedule execution according to prediction-based priority queue resources*<br>
*Else*<br>
*Alert Message*<br>
*End if*<br>
*Else*<br>
*Alert Message*<br>
*End if*<br>
*End loop*<br>
***# Phase Two: Perdition based Self-Healing***<br>
*Begin*<br>
*Set of Prediction based Priority Queue Nodes:* $PNode_{set} = \{PNode_1, PNode_2, \ldots\ldots, PNode_n\}$, *where* $PNode_c$ *represents current node of queue.*<br>
***If*** *(Predictive Priority Queue == Empty)* ***then***<br>
*Scan drives and check replica of original driver*<br>
*Add node into node set from the current node number*<br>
***Else***<br>
*Generate alert for Priority queue node is already exist*<br>
***End if***<br>
*Repeat loop until all hardware priority queue node (Status of Node)*<br>
*Get detail of current status [EVENTTYPE, TIMESTAMP, EVENTID]*<br>
***If*** *(EVENTTYPE = = 'EMERGENCY' OR 'ERROR')* ***then***<br>
*Database is updated by using log information [NodeName and address of MAC]*<br>
***End if***<br>
***End loop***<br>
*Loop until repeat Software Monitoring [Resource utilization (MEMORY and CPU)]*<br>
***If*** *(CurrentStatus ['CPU' || 'MEMORY'] > Value of THRESHOLD)* ***then***<br>
*Generate alert message*<br>
*Update Resource utilization (Memory and CPU) information*<br>
***End if***<br>
***End loop***<br>
***# Prediction Based Auto Configuration for Self-Healing Process***<br>
*Begin*<br>
*Prediction base Priority Queue of H_Components:* $= \{Hc_1, Hc_2, \ldots\ldots, Hc_p\}$<br>
*Priority Queue of Active H_Components:=* $\{Hc_1, Hc_2, \ldots\ldots, Hc_q\}$, *where* $q \le p$<br>
***While*** *true* ***do***<br>
*Repeat loop for all software S_Components*<br>
*Repeat loop to get all Priority Queue of Active H_Components version status*<br>
***If*** *(Component version status = OLD || Not-VALID)* ***then***<br>
*INSTALL the new version for replacing the old version using the process of uninstall*<br>
***End if***<br>
***End if***<br>
***End loop***<br>
*Repeat loop all hardware H_Components then Track Log Register*<br>
*Repeat loop to get all detail of Priority Queue of Active H_Components status [EVENTTYPE, TIMESTAMP, EVENTID]*<br>
***If*** *(EVENTTYPE == 'EMERGENCY' || 'ERROR')* ***then***<br>
*Database is updated by using log information [H_Component_Name and H_Compoenent_Id]*<br>
***End if***<br>
***End loop***<br>
***End loop***

and monitored. If host CPU utilization < 0.21, then CPU is added to the underutilized host list. If host CPU utilization > 0.79, then the CPU is added to the over-utilized host list; otherwise, the host resides in a safe host list. Now, check which VM is maximally utilized and then upgrade VM configuration if possible; otherwise, migrate VM towards a safe host. Consolidate underutilized host VMs and either shut down the VM or add them to the migration list. Prepare safe host list, increasing order of CPU utilization and performing VM migration.

**Algorithm 3: VM Migration and Load Balancing**

1. *Based on the VM allocation policy schedule the load on the VM.*
2. *Repeat loop for every host to calculate the CPU utilization*
3. *Repeat until*
   - *3.1 Get the first host in the list*
   - *3.2 if host CPU utilization < 0.21, then host add into underutilized host list*
   - *3.3 if host CPU utilization > 0.79, then host add into over utilized host list*
   - *3.4 otherwise host add into safe host list close the condition of the loop*
4. *Repeat until over utilized host list get the VM with maximum utilization*
   - *4.1 get the available MIPS from the host of maximum utilized VM*
   - *4.2 if MIPS is available, then add available MIPS to over utilized VM*
   - *4.3 otherwise, migrate the VM to a safe host based on a safer policy*
5. *Repeat until for each underutilized host*
   - 5.1 *Consolidate every VM on the underutilized host and move those*
   - 5.2 *VMs to migration list close the condition of the loop*
6. *Organize the safe host in increasing order based on CPU utilization and migrate all the VMs based policy*

## 4. Simulation Setup, Results and Discussion

Our proposed Predictive Cloud Computing System is modeled and simulated using CloudSim. This research is carried out using the CloudSim toolkit [44]. The system modeling and behavior of cloud system components; i.e., VMs, Datacentre and RP rules are fully supported by the CloudSim toolkit [7]. The standard resource scheduling methods' implementation can be done with little effort and method extension is possible. The inter-networked and distinct clouds are contained in the cloud environment simulation using the toolkit.

Furthermore, the toolkit supports VM provisioning under an inter-networked cloud environment to implement resource scheduling techniques through custom interfaces. Toolkit benefits provide the performance with time effectiveness, flexibility and applicability for test results. The heterogeneous workload of clouds is considered for experimental results. Each available resource contains one or more processing elements with different Million Instructions Per Second (MIPS). In this outcome, we assume that every workload admitted to the Predictive Cloud Computing System (PCCS) contains a workload of fluctuating sizes of inputs and execution times. These workloads are considered in the form of Cloudlets [44].

The resource configurations for testbed are: 2.4 GHz, Intel Core 2 Duo, 160 GB HDD, 1 GB RAM, with Windows operating system, 2.9 GHz, Intel Core i5-2310, 160 GB HDD, 1 GB RAM, with Linux operating system, 2.0 GHz, Intel Core i7-8550, 256 GB HDD, 4 GB RAM, with Linux operating system. This paper simulates our results with four SLA self-healing MASA, SH-SLA, CHOPPER and RADAR with our new proposed works. According to our simulation results, the proposed prediction-based PCCS SLA is the best. Table 4 gives details of workload types along with missing deadline compensation provided. We have shown the different testbed results in the Table 1, Table 2 and Table 3 given below.

In this simulation, two different cloud infrastructures through different processor configurations (4-core processor and 8-core processor) have been considered to measure the variation of different QoS parameters; i.e., energy efficiency, execution cost, resource utilization, throughput, SLA violation rate, resource contention, waiting time, fault detection rate, reliability, availability, intrusion detection rate and turnaround time. The different QoS parameters Improvement Rate (IR) percentage and simulation statistics summary are described in the tables. The CloudSim simulation environment has been considered with 3000 same-type workload traces of PlanetLab for performance testing. The PCCS is validated for different QoS parameters through autonomic resource management existing techniques, such as CHOPPER [21], SH-SLA [8], RADAR [9] and MASA [22]. The PCCS performance is more stable and efficient in resource management for changing cloud workloads using the coefficient of variation with a small value. The simulation results are presented in Table 1, Table 2 and Table 3 for all the different cloud infrastructures.

Table 1. Simulation results and improvement rate (IR) of PCCS and CHOPPER.

| QoS Parameters | 4-Core Processor | | | 8-Core Processor | | |
|---|---|---|---|---|---|---|
| | PCCS | CHOPPER | IR (%) | PCCS | CHOPPER | IR (%) |
| Energy consumption (kWh) | 88.91 | 117.61 | 24.4 | 124.46 | 162.13 | 23.23 |
| Execution cost (C$) | 94.2 | 128.71 | 26.81 | 162.7 | 219.56 | 25.90 |
| Resource utilization (%) | 79.11 | 71.01 | 10.24 | 83.66 | 77.761 | 7.05 |
| Energy efficiency (%) | 89.81 | 82.85 | 7.75 | 81.45 | 73.89 | 9.28 |
| Throughput (workload/sec) | 549.8 | 559.19 | 1.68 | 669.43 | 619.55 | 7.45 |
| SLA violation rate (%) | 28.15 | 36.56 | 23.00 | 41.46 | 47.91 | 13.46 |
| No. of missed deadlines | 28.11 | 34 | 17.32 | 44 | 49 | 10.20 |
| Resource contention (sec) | 3416.56 | 4180.48 | 18.27 | 4830.78 | 5461.45 | 11.55 |
| Waiting time (sec) | 299.32 | 306.69 | 2.40 | 268.69 | 266.15 | 0.95 |
| Fault detection rate (%) | 67.48 | 64.78 | 4.00 | 74.98 | 71.12 | 5.15 |
| Reliability (%) | 7.91 | 6.23 | 21.24 | 8.42 | 8.18 | 2.85 |
| Availability (%) | 86.39 | 82.71 | 4.26 | 89.77 | 89.22 | 0.61 |
| Intrusion detection rate (%) | 27.98 | 26.48 | 5.36 | 48.78 | 44.69 | 8.38 |
| Turnaround time (sec) | 622.15 | 651.45 | 4.50 | 561.89 | 593.28 | 5.29 |

Table 2. Simulation results and improvement rate (IR) of PCCS and RADAR.

| QoS Parameters | 4-Core Processor | | |
|---|---|---|---|
| | PCCS | RADAR | IR (%) |
| Energy consumption (kWh) | 88.91 | 110.61 | 19.62 |
| Execution cost (C$) | 94.2 | 118.71 | 20.65 |
| Resource utilization (%) | 79.11 | 69.01 | 12.77 |
| SLA violation rate (%) | 28.15 | 35.56 | 20.84 |
| Fault detection rate (%) | 67.48 | 64.78 | 4.00 |
| Turnaround time (sec) | 622.15 | 651.45 | 4.50 |

Table 3. Simulation results and improvement rate (IR) of PCCS, MASA and SH-SLA.

| QoS Parameters | 4-Core Processor | | | | |
|---|---|---|---|---|---|
| | PCCS | MASA | IR (%) | SH-SLA | IR (%) |
| Energy consumption (kWh) | 88.91 | 121.61 | 26.89 | 122 | 27.12 |
| Execution cost (C$) | 94.2 | 125.71 | 25.07 | 129 | 26.98 |
| Resource utilization (%) | 79.11 | 66.01 | 16.56 | 63 | 20.36 |
| SLA violation rate (%) | 28.15 | 37.56 | 25.05 | 38 | 25.92 |
| Fault detection rate (%) | 67.48 | 61.78 | 8.45 | 60.78 | 9.93 |
| Turnaround time (sec) | 622.15 | 658.45 | 5.51 | 666.3 | 6.63 |

Table 4. Workload urgency details with their types.

| Load type | Emergency deadline (P_*Du*) | Slack time (seconds) | Delay time (seconds) | Deviation status | Minimum penalty | Penalty rate |
|---|---|---|---|---|---|---|
| Emergency Deadline | P_Du < 0.25 | 10 | 0–50 | 5 % | 200 s | 5 % |
| | | | 51–100 | 10 % | 400 s | 6 % |
| | | | 101–150 | 15 % | 600 s | 7 % |
| Medium Deadline | 0.25≤P_Du ≤0.75 | 30 | 0–50 | 5 % | 100 s | 4 % |
| | | | 51–100 | 10 % | 200 s | 5 % |
| | | | 101–150 | 15 % | 300 s | 6 % |
| Deadline | P_Du > 0.75 | 60 | 0–50 | 5 % | 50 s | 2 % |
| | | | 51–100 | 10 % | 100 s | 3 % |
| | | | 101–150 | 15 % | 150 s | 4 % |

The results demonstrate that PCCS improves average resource utilization by 13.40%, average energy efficiency by 8.52%, average fault detection rate by 6.31%, average intrusion detection rate by 6.87%, average throughput by 6.39%, average reliability by 12.04%, average availability by 2.44% and minimizes average SLA violation rate by 20.44%, average energy consumption by 24.25%, average execution cost by 23.82%, average number of missed deadlines by 11.34%, average resource contention by 10.36%, average waiting time by 2.59% and average turnaround time by 5.59% as likened to existing resource management techniques. As per the simulation results, it is clearly shown that PCCS outperforms existing techniques in terms of QoS parameters, as PCCS achieves every situation automatically.

Figures 3, 4, 5 and 6 represent the proposed PCCS scheme results of different QoS parameter comparison with the existing techniques of SH-SLA, MASA, RADAR and CHOPPER. The proposed Predictive Cloud Computing System (PCCS) performs better in terms of energy consumption, execution cost, resource utilization, fault detection rate, turnaround time and SLA violation rate for SLA-aware autonomic resource management and gives better results for SLA violation rate along with different QoS parameters.

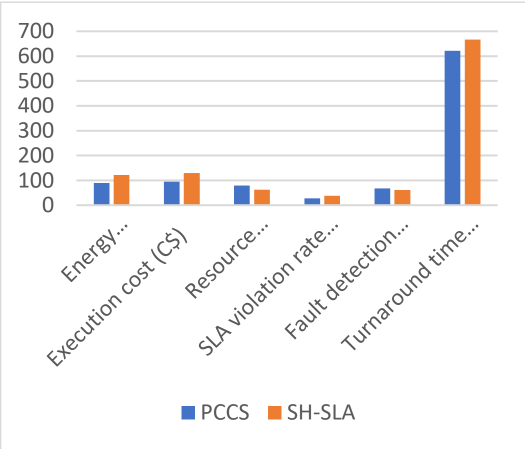



Figure 3. SH-SLA and proposed PCCS comparison on different QoS parameters.

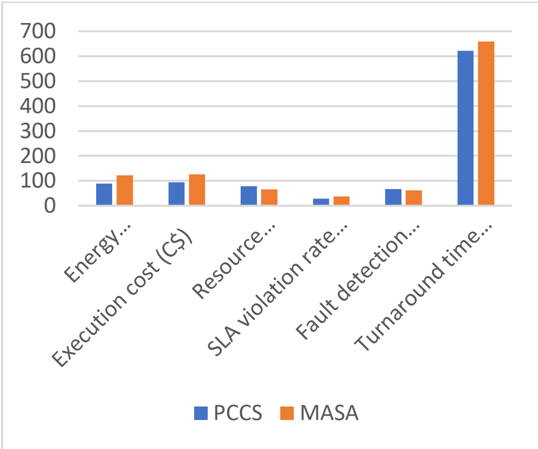



Figure 4. MASA and proposed PCCS comparison on different QoS parameters.

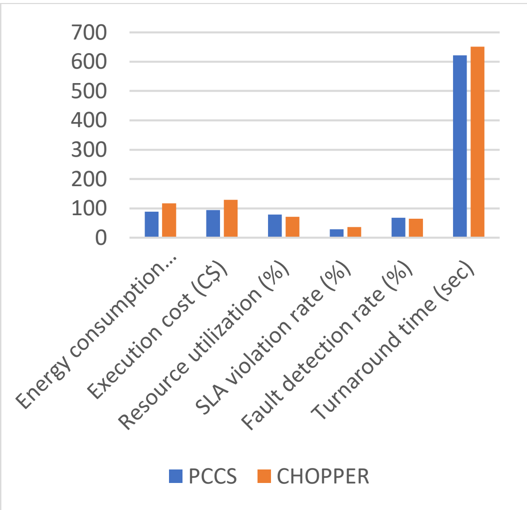



Figure 5. CHOPPER and proposed PCCS comparison on different QoS parameters.

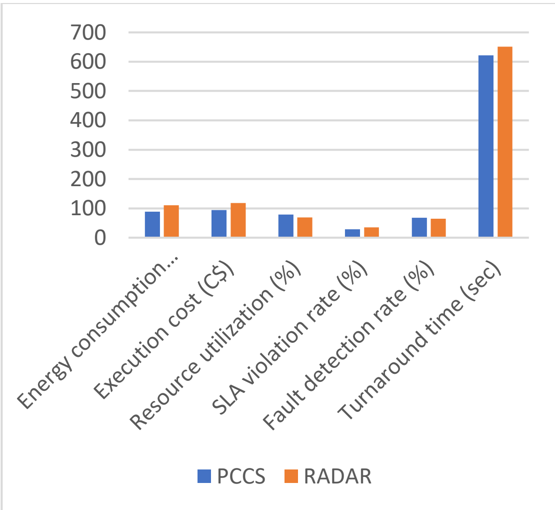



Figure 6. RADAR and proposed PCCS comparison on different QoS parameters.

Figure 3 clearly shows that the proposed PCCS improves resource utilization by 20.36%, fault detection rate by 9.93% and decrease energy consumption by 27.12%, execution cost by 26.98%, turnaround time by 6.63%, an SLA violation rate by 25.92% in comparison to SH-SLA. In Figure 4, the proposed PCCS technique comparative analysis simulated with MASA scheme and simulation results show that the proposed scheme gives better results for fault detection rate by 8.45%, resource utilization by 16.56%, turnaround time by 5.51%, cost of execution by 25.07%, consumption of energy by 26.89% and rate of SLA violation by 25.05%. Figure 5 represents the simulation results of CHOPPER with PCCS proposed technique which justify that comparatively the proposed PCCS takes less energy consumption by 23.81%, execution cost by 26.35%, turnaround time by 4.89%, SLA violation rate by 18.23%, resource utilization by 8.64% and the fault detection rate by 4.57%. Figure 6 clearly shows that the PCCS improves resource utilization by 12.77% and fault detection rate by 4% and decreases energy consumption by 19.62%, execution cost by 20.65%, turnaround time by 4.50% and SLA violation rate by 20.84% in comparison to RADAR.

Our paper has discussed how the cloud provider provides better quality in a Cloud Environment during the user request for resources and management. As the SLA between user and provider is the most crucial document, the proposed prediction base management and PCCS have presented a predictive approach to resource management using a VM migration policy. That will effectively address the overloading problem and provide cloud resource prediction as per SLA for user QoS requirements, where no human intervention will improve user satisfaction.

The proposed model works based on prediction; therefore, configuration, healing, protection and optimization have been automatically done. Our simulation is done on CloudSim in terms of various parameters, such as throughput, reliability, fault detection rate, turnaround time, waiting time and SLA violation rate. According to the results, the proposed prediction-based approach is better than the existing SLA frameworks. Our proposed framework leads to improve the scalability of cloud-based services. Our proposed algorithm simulation reduces the cost and execution time; therefore, this will lead to saving energy. The simulation is done based on the number of VM migrations and SLA violations. Comparison of the results with those of the existing frameworks shows a reduction in VM migrations in energy consumption in the data center to measure energy consumption in terms of idle hosts.

## 5. Conclusion

In this paper, fault-tolerance using a self-healing SLA-based Predictive Cloud Computing System (PCCS) has been proposed with self-management property for heterogeneous workload execution. The main goal of our PCCS is to minimize the SLA violation rate and increase user satisfaction levels by fulfilling their QoS requirements. We propose a new model that uses VMs as a resource allocation unit that provisions threshold-based dynamic allocation of cloud computing resources that performs prediction on the future need of resources by the PCCS scheme. This scheme will prepare resources required as per the future need of the users' applications. The proposed method predicts the future demand of user applications based on historical databases of workload demands. The scheme makes resources ready for provision after predicting the required resource demands and fulfils the actual needs of the application without SLA violation. The proposed method can dynamically configure the necessary resources based on the threshold-based load balancing technique and maximize available cloud resource utilization with reduced user-usage cost. The PCCS improves average resource utilization, energy efficiency, fault detection rate and throughput. It minimizes average energy consumption, execution cost, missed deadlines, resource contention, waiting time and turnaround time. The simulation results show that the proposed PCCS performs better than existing resource provisioning techniques in terms of SLA violation rate.

Our work presented resource requirements prediction, but we have not included hard disk, traffic, network utilization and bandwidth for the forecast. Therefore, future research could focus on having more resources, such as hard disk and bandwidth, for the prediction model. The prospective study consists of work on network utilization and network traffic to maintain scalability of the proposed model.

## References

[1] A. Quiroz, H. Kim, M. Parashar, N. Gnanasambandam and N. Sharma, "Towards Autonomic Workload Provisioning for Enterprise Grids and Clouds," Proc. of the 10th IEEE/ACM International Conference on Grid Computing, pp. 50-57, Banff, AB, Canada, Oct. 2009.

[2] T. Lorido-Botran, J. Miguel-Alonso and J. A. Lozano, "A Review of Auto-scaling Techniques for Elastic

Applications in Cloud Environments," Journal of Grid Computing, vol. 12, no. 4, pp. 559-592, Oct. 2014.

[3] Md. Toukir Imam, S. F. Miskhat, R. M. Rahman and M. A. Amin, "Neural Network and Regression-based Processor Load Prediction for Efficient Scaling of Grid and Cloud Resources," Proc. of the 14th IEEE International Conference on Computer and Information Technology (ICCIT 2011), pp. 333-338, Dhaka, Bangladesh, Dec. 2011.

[4] Z. Zhou, J. Abawajy, M. Chowdhury, Z. Hu, K. Li, H. Cheng, A. A. Alelaiwi and F.-M. Li, "Minimizing SLA Violation and Power Consumption in Cloud Data Centers Using Adaptive Energy-aware Algorithms," Future Generation Computer Systems, vol. 86, pp. 836-850, 2018.

[5] J. Zhu, P. He, Z. Zheng and M. R. Lyu, "Online QoS Prediction for Runtime Service Adaptation *via* Adaptive Matrix Factorization," IEEE Transactions on Parallel and Distributed Systems, vol. 28, no. 10, pp. 2911-2924, Oct. 2017.

[6] Shalu and D. Singh, "Swarm Intelligence Based Virtual Machine Migration Techniques in Cloud Computing," Proc. of the International Conference on Computation, Automation and Knowledge Management (ICCAKM), pp. 120-124, Dubai, United Arab Emirates, 2020.

[7] A. M. R. AlSobeh, S. AlShattnawi, A. Jarrah and M. M. Hammad, "WEAVESIM: A Scalable and Reusable Cloud Simulation Framework Leveraging Aspect-oriented Programming," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 06, no. 02, pp. 182-201, June 2020.

[8] R. Yadav, W. Zhang, K. Li et al., "Managing Overloaded Hosts for Energy-efficiency in Cloud Data Centers," Cluster Computing, vol. 2021, DOI: 10.1007/s10586-020-03182-3, Feb. 2021.

[9] D. A. Shafiq, N. Z. Jhanjhi, A. Abdullah and M. A. Alzain, "A Load Balancing Algorithm for the Data Centres to Optimize Cloud Computing Applications," IEEE Access, vol. 9, pp. 41731-41744, 2021.

[10] S. K. Pande, S. K. Panda, S. Das, K. S. Sahoo, A. K. Luhach et al., "A Resource Management Algorithm for Virtual Machine Migration in Vehicular Cloud Computing," Computers, Materials & Continua, vol. 67, no.2, pp. 2647–2663, 2021.

[11] M. A. Shahid, N. Islam, M. M. Alam, M. M. Su’ud and S. Musa, "A Comprehensive Study of Load Balancing Approaches in the Cloud Computing Environment and a Novel Fault Tolerance Approach," IEEE Access, vol. 8, pp. 130500-130526, 2020.

[12] Z. Chen, K. Lin, B. Lin, X. Chen, X. Zheng and C. Rong, "Adaptive Resource Allocation and Consolidation for Scientific Workflow Scheduling in Multi-cloud Environments," IEEE Access, vol. 8, pp. 190173-190183, 2020.

[13] B. Gul et al., "CPU and RAM Energy-based SLA-aware Workload Consolidation Techniques for Clouds," IEEE Access, vol. 8, pp. 62990-63003, 2020.

[14] R. Yadav, W. Zhang, K. Li et al., "An Adaptive Heuristic for Managing Energy Consumption and Overloaded Hosts in a Cloud Data Center," Wireless Networks, vol. 26, pp. 1905–1919, April 2020.

[15] W. Dargie, "Estimation of the Cost of VM Migration," Proc. of the 23rd IEEE International Conference on Computer Communication and Networks (ICCCN), pp. 1-8, Shanghai, China, 2014.

[16] S. Singh, I. Chana, M. Singh et al., "SOCCER: Self-optimization of Energy-efficient Cloud Resources," Cluster Computing, vol. 19, no. 4, pp. 1787–1800, Sep. 2016.

[17] M. Sohani and S. C. Jain, "A Predictive Priority-based Dynamic Resource Provisioning Scheme with Load Balancing in Heterogeneous Cloud Computing," IEEE Access, vol. 9, pp. 62653-62664, April 2021.

[18] F. Yao, C. Pu and Z. Zhang, "Task Duplication-based Scheduling Algorithm for Budget-constrained Workflows in Cloud Computing," IEEE Access, vol. 9, pp. 37262-37272, 2021.

[19] H. M. Khan, G. Chan and F. Chua, "An Adaptive Monitoring Framework for Ensuring Accountability and Quality of Services in Cloud Computing," Proc. of the International Conference on Information Networking (ICOIN), pp. 249-253, Kota Kinabalu, Malaysia, 2016.

[20] R. Latif, S. H. Afzaal and S. Latif, "A Novel Cloud Management Framework for Trust Establishment and Evaluation in a Federated Cloud Environment," The Journal of Supercomputing, vol. 2021, DOI: 10.1007/s11227-021-03775-8, April 2021.

[21] A. Mosallanejad, R. Atan, M. Azmi Murad and R. Abdullah, "A Hierarchical Self-healing SLA for Cloud Computing," International Journal of Digital Information and Wireless Communications (IJDIWC), vol. 4, no. 1, pp. 43-52, 2014.

[22] S. S. Gill, I. Chana, M. Singh and R. Buyya, "RADAR: Self-configuring and Self-healing in Resource

Management for Enhancing Quality of Cloud Services," Concurrency and Computation: Practice and Experience, vol. 31, no. 1, DOI: 10.1002/cpe.4834, Aug. 2018.

[23] S. Banerjee, S. Roy and S. Khatua, "Efficient Resource Utilization Using Multi-step-ahead Workload Prediction Technique in Cloud," The Journal of Supercomputing, vol. 2021, DOI: 10.1007/s11227-021-03701-y, March 2021.

[24] R. Yadav, W. Zhang, O. Kaiwartya, P. R. Singh, I. A. Elgendy and Y. Tian, "Adaptive Energy-aware Algorithms for Minimizing Energy Consumption and SLA Violation in Cloud Computing," IEEE Access, vol. 6, pp. 55923-55936, 2018.

[25] S. Sotiriadis, N. Bessis and R. Buyya, "Self-managed Virtual Machine Scheduling in Cloud Systems," Information Sciences, vol. 433-434, pp. 381–400, 2018.

[26] A. Paya and D. C. Marinescu, "Energy-aware Load Balancing and Application Scaling for the Cloud Ecosystem," IEEE Transactions on Cloud Computing, vol. 5, no. 1, pp. 15-27, 2017.

[27] I. Odun-Ayo, B. Udemezue and A. Kilanko, "Cloud Service Level Agreements and Resource Management", Advances in Science, Technology and Engineering Systems Journal, vol. 4, no. 2, pp. 228-236, 2019.

[28] R. Yadav, W. Zhang, H. Chen and T. Guo, "MuMs: Energy-aware VM Selection Scheme for Cloud Data Center," Proc. of the 28th IEEE International Workshop on Database and Expert Systems Applications (DEXA), pp. 132-136, Lyon, France, 2017.

[29] M. Dabbagh, B. Hamdaoui, M. Guizani and A. Rayes, "Toward Energy-efficient Cloud Computing: Prediction, Consolidation and Over-commitment," IEEE Network, vol. 29, no. 2, pp. 56-61, 2015.

[30] E. Torre, J. J. Durillo, V. de Maio, P. Agrawal, S. Benedict, N. Saurabh and R. Prodan, "A Dynamic Evolutionary Multi-objective Virtual Machine Placement Heuristic for Cloud Data Centers," Information and Software Technology, vol. 128, DOI: 10.1016/j.infsof.2020.106390, 2020.

[31] F. Farahnakian, T. Pahikkala, P. Liljeberg, J. Plosila, N. T. Hieu and H. Tenhunen, "Energy-aware VM Consolidation in Cloud Data Centers Using Utilization Prediction Model," IEEE Transactions on Cloud Computing, vol. 7, no. 2, pp. 524-536, 2019.

[32] D. Abdulkareem Shafiq, N. Z. Jhanjhi and A. Abdullah, "Load Balancing Techniques in Cloud Computing Environment: A Review," Journal of King Saud University - Computer and Information Sciences, DOI: 10.1016/j.jksuci.2021.02.007, 2021.

[33] M. Balaji, Ch. Aswani Kumar and G. Subrahmanya V. R. K. Rao, "Predictive Cloud Resource Management Framework for Enterprise Workloads," Journal of King Saud University - Computer and Information Sciences, vol. 30, no. 3, pp. 404-415, 2018.

[34] F. Ebadifard and S. M. Babamir, "Autonomic Task Scheduling Algorithm for Dynamic Workloads through a Load Balancing Technique for the Cloud-computing Environment," Cluster Computing, vol. 24, pp. 1075-1101, June 2021.

[35] N. Chaurasia, M. Kumar, R. Chaudhry et al., "Comprehensive Survey on Energy-aware Server Consolidation Techniques in Cloud Computing," The Journal of Supercomputing, vol. 2021, DOI: 10.1007/s11227-021-03760-1, March 2021.

[36] B. K. Dewangan, A., M., V. Agarwal and A. Pasricha, "Energy-aware Autonomic Resource Scheduling Framework for Cloud," International Journal of Mathematical, Engineering and Management Sciences, vol. 4, no. 1, pp. 41-55, 2019.

[37] A. A. Hassan, B. M. Bai and T. J. Gandomani, "An Integrated Model for Secure-on-Demand Resource Provisioning Based on Service Level Agreement (SLA) in Cloud Computing," Journal of Theoretical and Applied Information Technology, vol. 65, no. 2, July 2014.

[38] M. Sohani and S. C. Jain, "State-of-the-art Survey on Cloud Computing Resource Scheduling Approaches," Proc. of Ambient Communications and Computer Systems, Part of the Advances in Intelligent Systems and Computing Book Series, vol. 696, pp. 629-639, March 2018.

[39] W. Lin, J. Z. Wang, C. Liang and D. Qi, "A Threshold-based Dynamic Resource Allocation Scheme for Cloud Computing," Procedia Engineering, vol. 23, pp. 695-703, 2011.

[40] S. S. Gill, I. Chana, M. Singh et al., "CHOPPER: An Intelligent QoS-aware Autonomic Resource Management Approach for Cloud Computing," Cluster Computing, vol. 21, pp. 1203–1241, 2018.

[41] H. Alhussian et al., "Investigating the Schedulability of Periodic Real-time Tasks in Virtualized Cloud

Environment," IEEE Access, vol. 7, pp. 29533-29542, 2019.

[42] M. Azaiez and W. Chainbi, "A Multi-agent System Architecture for Self-healing Cloud Infrastructure," Proceedings of the International Conference on Internet of Things and Cloud Computing (ICC'16), pp. 1-6, DOI: 10.1145/2896387.2896392, March 2016.

[43] S. Talwani and I. Chana, "Fault Tolerance Techniques for Scientific Applications in Cloud," Proc. of the 2nd International Conference on Telecommunication and Networks (TEL-NET), pp. 1-5, Noida, India, 2017.

[44] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose and R. Buyya, "CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms," Software – Practice and Experience, vol. 41, no. 1, pp. 23–50, August 2010.

**ملخص البحث:**

ظهرت حديثاً إدارة فاعلية التطبيقات كجزء أساسي من خدمات الحوسبة السّحابية على الشبكة العنكبوتية. ويقدّم مزود الخدمة السّحابية خدماتٍ متنوعة على أساس الدّفع مقابل الاستخدام؛ الأمر الذي يستلزم الرّصد والقياس للخدمات المقدمة من أجل إدارة جودة الخدمة من خلال إنترنت الأشياء ومن ثم تحقيق الشّروط الواردة في اتفاقية مستوى الخدمة. ومع ذلك، فإنّ تجنُّب خرق الاتفاقية وضمان تحقيق متطلبات المستخدمين الديناميـة يشكلان تحدياً في الحوسبة السّحابية. ويتعاظم التّعقيد وعدم التجانس والديناميـة بسرعةٍ مما يجعل أُطر العمل المتعلقة بالحوسبة السّحابية عصيّة على الإدارة الفاعلة وغير موثوقة. وإنّ الأنظمة السّحابية تحتاج الى إدارة ذاتية للخدمات للتغلُّب على تلك المشكلات. لذا، فإنّ ثمّة حاجة الى تطوير خطة لاتّخاذ التّدابير اللازمة التي من شأنها أن تلبّي متطلبات المستخدمين لأنظمة الحوسبة السّحابية فيما يرتبط بجودة الخدمة وتحول دون خرق اتفاقية مستوى الخدمة. هذه الورقة تقدم تقنيةً مقترحة لإدارة المصادر قائمة على التّوقُّع تسمى (النّظام التّوقُّعي للحوسبة السّحابية). وفي هذا النظام، يتم التركيز على التّوقُّع المستند على الالتئام الذّاتي الذي يُعالج الأخطاء غير المتوقعة، وعلى التّوقُّع المستند على التّرتيب الذاتي للمصادر المتعلقة بالتّطبيقات. وقد جرى تقييم النّظام المقترح في المحاكي السّحابي. وكشفت نتائج المحاكاة أنّ النّظام المقترح تفوّق من حيث الأداء على تقنياتٍ قائمة من حيث زمن التّنفيذ، والجدوى المتعلقة بالتكلفة، وتعارُض المصادر، وخرق اتفاقيات مستوى الخدمة؛ في الوقت الذي قدّم فيه خدماتٍ موثوقة.

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:

ELSEVIER Scopus

DOAJ DIRECTORY OF OPEN ACCESS JOURNALS

crossref Text and Data Mining

OCLC WorldCat

BASE

ACADEMIA

Google scholar






## 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: +962-6-5359949

فاكس: +962-6-7295534

Princess Sumaya University for Technology
PSUT
Since 1991 تأسست
جامعة الأميرة سمية للتكنولوجيا

صندوق دعم البحث العلمي
Scientific Research Support Fund

# المجلة الأردنية للحاسوب و تكنولوجيا المعلومات

حزيران ٢٠٢١ المجلد ٧ العدد ٢
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